www.wisa.or.kr



The 25th World Conference on Information Security Applications

WISA 2024

August 21-23, 2024

MAISON GLAD, Jeju Island, Korea

Co-Hosted by

Korea Institute of Information Security & Cryptology

ETRI Electronics and Telecommunications Research Institute

Sponsored by

NIS

Ministry of Science and ICT

KISA KOREA INTERNET & SECURITY AGENCY

NSR 국가보안기술연구소 National Security Research Institute

KiSTi Korea Institute of Science and Technology Information www.kisti.re.kr

JEJU NATIONAL UNIVERSITY

LG U+

ESTSECURITY



The 25th World Conference on Information Security Applications

WISA 2024

CONTENTS

Papers 3

Posters 347



The 25th World Conference on Information Security Applications

WISA 2024

Papers

CONTENTS

The 2024 WISA Paper Session

DAY 1 August 21 (Wednesday), 2024

Session 1: Cryptography
Session Chair: Keita Emura (Kanazawa University)

ECPM Cryptanalysis Resource Estimation
Dedy Septono Catur Putranto, Rini Wisnu Wardhani, Jaehan Cho and Howon Kim (Pusan National University)

Integral Attack with Bit-based Division Property on the Lightweight Block Cipher LBC
Naoki Shibayama and Yasutaka Igarashi (Tokyo University of Science)

Session 2: Network Security
Session Chair: Sokjoon Lee (Gachon University)

ARP Spoofing Mitigation for the E2 Interface in Open RAN: An xApp Approach
Jihye Kim, Jaehyoung Park and Jong-Hyouk Lee (Sejong University)

Investigating the Transferability of Evasion Attacks in Network Intrusion Detection Systems Considering Domain-Specific Constraints
Mariama Mbow (Kyushu University), Rodrigo Román (University of Malaga), Ayan Seal (PDPM Indian Institute of Information Technology, Design and Manufacturing), Kevin I-Kai Wang (The University of Auckland), Sraban Kumar Mohanty (PDPM Indian Institute of Information Technology, Design and Manufacturing) and Kouichi Sakurai (Kyushu University)

Enhancing Robustness in NIDS via Coverage Guided Fuzzing and Adversarial Training
Jung Yup Rhee, Leo Hyun Park and Taekyoung Kwon (Yonsei University)

DAY 2 August 22 (Thursday), 2024

Session 3: AI Security
Session Chair: Taekyoung Kwon (Yonsei University)

VoteGAN: Generalized Membership Inference Attack Against Generative Models by Multiple Discriminators

Gyeongsup Lim, Wonjun Oh and Junbeom Hur (Korea University)

Unsupervised Contextual Anomalous Communication Detection Using VQ Tokenization with Flow Data

Norihiro Okui (KDDI Research Inc.), Shotaro Fukushima (ARISE analytics inc.), Ayumu Kubota (KDDI Research Inc.) and Takuya Yoshida (Toyota Motor Corporation)

Boosting Black-Box Transferability of Weak Audio Adversarial Attacks with Random Masking

Mai Bui, Thien-Phuc Doan, Kihun Hong and Souhwan Jung (Soongsil University)

Session 4: Network & Application Security
Session Chair: Hwajeong Seo (Hansung University)

Security Function for 5G-Advanced and 6G: Malicious UE Detection in Network Slicing

Jaehyoung Park, Jihye Kim and Jong-Hyouk Lee (Sejong University)

Simple Perturbations Subvert Ethereum Phishing Transactions Detection: An Emperical Analysis

Ahod Alghuried and David Mohaisen (University of Central Florida)

LLM Guardrail Framework: A Novel Approach for Implementing Zero Trust Architecture

Bogeum Kim, Hyejin Sim, Jiwon Yun, Jaehan Cho and Howon Kim (Pusan National University)

Session 5: AI Security
Session Chair: DaeHun Nyang (Ewha Womans University)

ChatDEOB: An Effective Deobfuscation Method based on Large Language Model

Byunggeon Choi, Hongjoo Jin, Dong Hoon Lee and Wonsuk Choi (Korea University)

An effective ensemble algorithm for short-term load forecasting

En-Wei Zhang, Luo-Fan Wu and Chun-Wei Tsai (National Sun Yat-sen University)

Session 6: CPS Security
Session Chair: Heejun Roh (Inha University)

PUF-Based Authentication and Authorization Protocol for IoT

Yu-Tse Shih (National Sun Yat-sen University), Shih-Ming Huang (Industrial Technology Research Institute) and Chun-I Fan (National Sun Yat-sen University)

Field Testing and Detection of Camera Interference for Autonomous Driving

Ki Beom Park and Huy Kang Kim (Korea University)

One-class Classification-based Position Falsification Detection System in C-ITS Communication Network

Ilhwan Ji, ChangJin Choi (Gachon University), Cheolho Lee (ENKI WhiteHat), Seungho Jeon and Jung Taek Seo (Gachon University)

DAY 3 August 23 (Friday), 2024

Session 7: Fuzzing
Session Chair: Chae-Tae Im (KISA)

Reverse Engineering-Guided Fuzzing for CAN Bus Vulnerability Detection

Manu Jo Varghese, Frank Jiang (Deakin University), Abdur Rakib (Coventry University), Robin Doss and Adnan Anwar (Deakin University)

Air-Fuzz: Feasibility analysis of fuzzing-based side-channel information leakage attack in air-gapped networks

Yeon-Jin Kim, Na-Eun Park and Il-Gu Lee (Sungshin Women's University)

Fuzzing JavaScript Engines with Diversified Mutation Strategies

Cheng-Han Shie, Pin-Huang Fang and Chun-I Fan (National Sun Yat-sen University)

Session 8: Malware
Session Chair: Deok Gyu Lee (Seowon University)

Detecting Phishing-targeted Web Push Notifications through Image Similarity Analysis

Rion Matsuzaki and Masaya Sato (Okayama Prefectural University)

Dissecting the Infrastructure Used in Web-based Cryptojacking: A Measurement Perspective

Ayodeji Adeniran, Kieran Human and David Mohaisen (University of Central Florida)

A Survey on Attack Cases with VBS Malware in Windows

GyuHyun Jeon, Seungho Jeon and Jung Taek Seo (Gachon University)

Session 9: Software Security

Session Chair: Jungtae Kim (ETRI)

Adversarial Manhole: Challenging Monocular Depth Estimation and Semantic Segmentation Models with Patch Attack

Naufal Suryanto, Andro Aprila Adiputra, Ahmada Yusril Kadiptya (Pusan National University), Yongsu Kim (SmartM2M) and Howon Kim (Pusan National University)

Plotting OSS-based Supply Chain attack strategies and the defense failure

Arpita Dinesh Sarang, Sang-Hoon Choi and Ki-Woong Park (Sejong University)

A Proposal of a Supply Chain Security Model for Generative AI

Keun Young Lee (Financial Security Institute) and Jiyeon YOO (Sangmyung University)

Session 10: Emerging Topic

Session Chair: Youngsoo Kim (ETRI)

Privacy Enhanced P2P Power Trading Using DID-based Identity Verification in a Microgrid

Taeyang Lee and Jong-Hyouk Lee (Sejong University)

Quantum Implementation of LSH

Yujin Oh, Kyungbae Jang and Hwajeong Seo (Hansung University)

Who ruins the game?: unveiling cheating users in the “Battlefield" game

Dong Young Kim and Huy Kang Kim (Korea University)

ECPM Cryptanalysis Resource Estimation

Dedy Septono Catur Putranto[1,2][0000−0002−1246−7877] *, Rini Wisnu Wardhani[3][0000−0003−0565−6458], Jaehan Cho[3][0009−0008−4878−9123], and Howon Kim[3][0000−0001−8475−7294]

[1] IoT Research Center, Pusan National University, Busan 609735, South Korea
[2] Blockchain Platform Research Center, Pusan National University, Busan 609735,
[3] South Korea School of Computer Science and Engineering, Pusan National University, Busan 609735, South Korea
howonkim@pusan.ac.kr

Abstract. Elliptic Curve Point Multiplication (ECPM) is a key component of the Elliptic Curve Cryptography (ECC) hierarchy protocol. However, the specific estimation of resources required for this process remains underexplored despite its significance in the cryptanalysis of ECC algorithms, particularly binary ECC in GF (2^m). Given the extensive use of ECC algorithms in various security protocols and devices, it is essential to conduct this examination to gain valuable insights into its cryptanalysis, specifically in terms of providing precise resource estimations, which serve as a solid basis for further investigation in solving the Elliptic Curve Discrete Logarithm Problem. Expanding on several significant prior research, in this work, we refer to as ECPM cryptanalysis, we estimate quantum resources, including qubits, gates, and circuit depth, by integrating point addition (PA) and point-doubling (PD) into the ECPM scheme, culminating in a Shor's algorithm-based binary ECC cryptanalysis circuit. Focusing on optimizing depth, we elaborate on and implement the most efficient PD circuit and incorporate optimized Karatsuba multiplication and FLT-based inversion algorithms for PA and PD operations. Compared to the latest PA-only circuits, our preliminary results showcase significant resource optimization for various ECPM implementations, including single-step ECPM, ECPM with combined or selective PA/PD utilization, and total−step ECPM ($2n$ PD $+2$ PA).

Keywords: ECC · ECPM · Point Addition · Point Doubling · Quantum Cryptanalysis.

1 Introduction

Elliptic-curve cryptography (ECC), introduced by Koblitz and Miller in 1985, has become a cornerstone of modern cryptography due to its ability to achieve equivalent security levels with significantly smaller key sizes than traditional cryptosystems. This efficiency has led to widespread adoption in various applications, including Transport Layer Security (TLS) 1.3 [13] and cryptographic standards set by the National Institute of Standards and Technology (NIST) [2]. However, the security of ECC hinges on the difficulty of the elliptic curve discrete logarithm problem (ECDLP). While ECC offers advantages in key size, Shor's algorithm [15], proposed in 1994, poses a significant threat to its long-term security. Extensive research has focused on the practical implementation of Shor's

* This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

algorithm for ECDLP on quantum computers, particularly targeting superconducting qubit architectures. These studies have explored both prime and binary elliptic curves, leading to significant advancements in optimizing quantum circuits for ECDLP solving (e.g., [[14], [4]]).

Building upon this prior work, this research delves deeper into the resource estimation of elliptic curve point multiplication (ECPM), a critical operation within ECC protocols (as depicted in Level 3 of Figure 1). While classical implementations of ECPM have explored optimized architectures utilizing Montgomery modular multiplication (e.g., [[10]]), our focus lies on building optimum depth and estimating resource requirements for ECPM in a quantum context. Notably, this research aims to estimate resources for ECPM as a building block for binary ECC cryptanalysis circuits based on Shor's algorithm, drawing inspiration from established works such as [[14], [1], [12]].

Specifically, we elaborate on depth optimization within the first and second levels of the ECC hierarchy by analyzing existing constructions in finite field arithmetic operations in GF(2^m) for binary ECC circuits. These optimized operations are then employed in both point addition (PA) and point doubling (PD) to construct a resource-efficient ECPM scheme (as the third level) that can be integrated into a binary ECC cryptanalysis circuit. We propose and compare optimized quantum circuit designs for these operations, focusing on a detailed comparative analysis of circuit depth. The key contributions of this research include:

- We leverage recent advancements in finite field arithmetic for GF(2^m). Specifically, we incorporate the improved Karatsuba multiplication algorithm proposed by Putaranto et al. [12] and the FLT-based inversion algorithm from Larasati et al. [9] (or incorporation research in [12]) into our quantum circuits for PA and PD. Additionally, we identify improved PA algorithm from existing research [12] and integrate it into the ECPM circuit construction.
- We analyze three PD circuit versions from Larasati et al. [8] and select the most depth-efficient model for inclusion within our ECPM scheme for binary ECC cryptanalysis based on Shor's algorithm. This choice deviates from the PD circuits employed in previous works by Roetteler et al. [14], Banegas et al. [1], and Putranto et al. [12], prioritizing circuit depth optimization.
- We construct an ECPM circuit within a quantum environment and perform a comprehensive resource estimation, including the number of qubits, gates, and circuit depth. Notably, we compare our findings with concrete binary ECC cryptanalysis previous studies that solely utilize PA circuits (e.g., [[1] [12]]). This comparison highlights the depth optimization achieved through our work.
- We report our preliminary results, based on comparisons with the most recent PA-only circuit from [12], demonstrating the resource efficiency achieved through our approach. These comparisons encompass single-step ECPM, ECPM with combined or selective PA/PD utilization, and total-step ECPM ($2n$ PD $+2$ PA).

2 Related Works

Shor's groundbreaking in 1994 addressed two fundamental problems: integer factorization and computing discrete logarithms in finite fields [15]. Building on this foundation, Proos and Zalka made significant strides by translating Shor's high-level ECDLP algorithm into a format suitable for quantum circuit implementation for specific elliptic curve groups [11]. This paved the way for a surge of research on the practical application of Shor's algorithm for ECDLP in quantum cryptanalysis.

Pioneering work by Roetteler et al. [14] and Haner et al. [4] refined quantum cryptanalysis techniques for ECC over prime fields. More recently, Banegas et al. [1] directed their efforts towards

optimizing Shor's algorithm for binary elliptic curves in a concrete quantum cryptanalysis circuit. Their work centered on optimizing the algorithm's critical multiplication and division circuits, proposing space-efficient techniques like Van Hoof's Karatsuba multiplication to reduce the required qubits and Toffoli gates [5]. The performance of two inversion circuits (GCD-based and FLT-based) was compared to calculate the resource cost of PA. Their primary objective was to minimize the number of qubits and Toffoli gates needed [1]. Complementing these binary field studies, Putranto et al. [12] and Larasati et al. [9] investigated depth-reduction techniques for implementing Shor's algorithm on quantum cryptanalysis binary elliptic curves. This focus on efficiency reflects the ongoing quest to improve the practicality of quantum cryptanalysis for ECC.

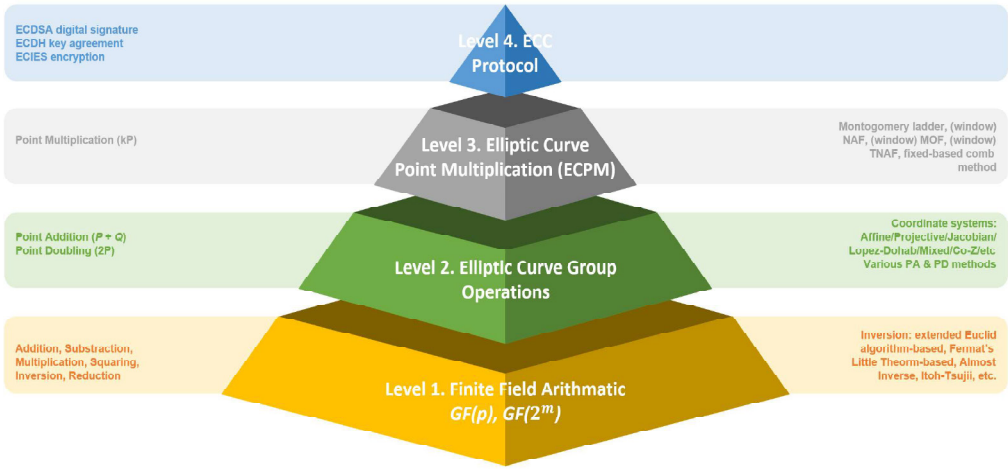


Fig. 1: General Hierarchy of ECC Protocol Implementation. We illustrate levels 1 through 4 as detailed in prior works [9], [12], and [8]. Specifically, this research focuses on levels 1 through 3 to construct the Elliptic Curve Point Multiplication (ECPM) based on several notable prior works.

Figure 1 illustrates a widely adopted hierarchy for ECC implementations. The base layer (Level 1) comprises finite field arithmetic operations in either prime or binary fields (GF(p) or GF(2^m)), including addition, subtraction, multiplication, and division. These operations serve as building blocks for higher-level ECC functionalities. Level 2 utilizes these operations to construct PA and PD circuits. Finally, Level 3 achieves elliptic curve point multiplication, refer as ECPM in this study. Optimization of the underlying quantum arithmetic circuits is crucial for the efficient execution of quantum algorithms like Shor's [12]. This optimization primarily focuses on minimizing circuit width or depth, especially for computationally expensive operations like multiplication and inversion. This becomes particularly critical for intricate circuits like PA, which forms the foundation of scalar multiplication as ECPM– the most resource-intensive step in circuits designed.

This study leverages efficient finite field arithmetic operations in GF(2^m) for binary ECC. Specifically, we examine the improved Karatsuba multiplication algorithm from Putranto et al. [12], and the FLT-based inversion technique from Larasati et al. [9] [12] for constructing second-level quantum circuits for PA and PD scheme. These choices aim to minimize the resource requirements

of the circuits. PA and PD's fundamental operations in ECC are crucial, as is scalar multiplication ECPM construction. Scalar multiplication, in turn, is essential for various cryptographic tasks such as generating public and private keys, encrypting and decrypting data, and verifying digital signatures.

3 Finite Field Arithmetic in GF (2^m) Binary ECC Operations

Elliptic curves defined over a binary field, denoted as $\mathbb{F}_{2^n}$, are a specific type of elliptic curve used in cryptography. Figure 1 depicts a wide hierarchy for implementing finite field arithmetic in ECC. This hierarchy highlights the importance of underlying field operations like modular addition, subtraction, multiplication, squaring, and inversion for optimal ECC implementations, regardless of whether the field is prime ($\mathbb{F}_p$) or binary ($\mathbb{F}_{2^n}$) [1]. This subsection provides a brief introduction to binary elliptic curve cryptography.

This experimental study adopts a polynomial representation for $\mathbb{F}_{2^n}$ similar to Banegas et al. [1] and Putranto et al. [12]. Elements are represented as polynomials of degree less than n with coefficients in $\mathbb{F}_2$. Computations leverage the isomorphism $\mathbb{F}_{2^n} \cong \mathbb{F}_2[z]/m(z)$, where $m(z)$ is an irreducible polynomial of degree n in $\mathbb{F}_2[z]$ in all computations are done modulo $m(z)$ [1]. For defining polynomial $m(z)$ used, Table 1 shows a list of irreducible polynomials in a curve over binary fields that are standardized in [6].

Table 1: List of standardized irreducible polynomials suitable for defining the binary field $\mathbb{F}_{2^n}$ used in elliptic curve cryptography (ECC) implementations. These polynomials are referenced in prior works by Banegas et al. [1], Putranto et al. [12], and are also employed in this study.

Degree (n)	Irreducible polynomial	Source
8	$x^8 + x^4 + x^3 + x + 1$	[3]
16	$x^{16} + x^5 + x^3 + x + 1$	[3]
127	$x^{127} + x + 1$	[3]
163	$z^{163} + z^7 + z^6 + x^3 + 1$	[6]
233	$z^{233} + z^{74} + 1$	[6]
283	$z^{283} + z^{12} + z^7 + z^5 + 1$	[6]
409	$z^{409} + z^{87} + 1$	[12]
571	$z^{571} + z^{10} + z^5 + z^2 + 1$	[6]

A binary elliptic curve is defined by the equation $y^2 + xy = x^3 + ax^2 + b$, where $a \in \mathbb{F}_2$ and $b \in \mathbb{F}_{2^n}^*$. Points on this curve are represented as tuples $P = (x, y) \in \mathbb{F}_{2^n}^2$ that satisfy the curve equation. Additionally, a special point denoted by O serves as the "point at infinity" and acts as the neutral element for PA. The negative of a point $P_1 = (x_1, y_1)$ is defined as $-P_1 = (x_1, y_1 + x_1)$, such that $P_1 + (-P_1) = O$.

PA on the curve is defined as follows. For two distinct points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, their sum, $P_1 + P_2 = P_3 = (x_3, y_3)$, is calculated using specific formulas: $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$ and $y_3 = (x_2 + x_3)\lambda + x_3 + y_2$ where $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$. Similarly, PD, denoted as $[2]P_1$, where $P_1 \neq -P_1$, is achieved using: $x_3 = \lambda^2 + \lambda + a$ and $y_3 = x_1^2 + (\lambda + 1)x_3$ with $\lambda = x_1 + \frac{y_1}{x_1}$.

Elliptic Curve Diffie-Hellman (ECDH) is a key exchange protocol based on elliptic curves. The sender and receiver agree on a public curve with a secret point P (having a large prime order) in

this protocol. Each party then chooses a secret integer, α for the sender and β for the receiver. The shared secret key is established by calculating shared points $P_{\alpha\beta}$. These shared points can be computed in two equivalent ways: $P_{\alpha\beta} = [\alpha\beta]P = [\alpha]P_\beta = [\beta]P_\alpha$ where $P_\alpha = [\alpha]P$ and $P_\beta = [\beta]P$ each shared chosen by the sender and the recipient. The security of ECDH relies on the difficulty of the ECDLP. Given a point P_α and the public base point P, the ECDLP problem asks for the secret integer α used to compute P_α. The complexity of finding or computing α from P_α and P is the basic concept of the ECDLP, which Peter Shor addressed to compute α in time polynomial in $ord(P)$ in his major work in [15].

3.1 Improved Karatsuba Multiplication

Putranto et al. [12] proposes modifying the Karatsuba multiplication algorithm used in prior work by Banegas et al. [1]. Their key improvement lies in eliminating the CONSTMODMULT^{-1} function, which relies heavily on Linear Universal Problem (LUP) decompositions and a high number of CNOT gates. This modification also involves restructuring the function within the algorithm, leading to a reduction in circuit depth, CNOT gates, and Toffoli gates compared to Banegas et al.'s approach. The core idea of Putranto et al.'s [12] improvement involves achieving the same Karatsuba multiplication result with fewer resources by utilizing more efficient operations like MODSHIFT^{-1}, a binary shift function, and free swap/relabeling functions. These functions rely on the irreducible polynomial defining the field rather than the field extension degree (n). The details of enhanced Karatsuba base algorithm are provided below [12]:

- Three calls to KMULT$_n$: two for multiplying k−by−k, and one for multiplying $(n-k)$−by−$(n-k)$.
- $2k$ calls to MODSHIFT$_{m(x)}$ for multiplication by x^k, once in reverse.
- One call to CONSTMODMULT$_{f(x),m(x)}$, with the same polynomial $1 + x^k$ being multiplied each time.
- 4 $(n-k)$ CNOT gates, with the ability to execute half of them simultaneously.

3.2 Improved FLT-based Inversion with Improved Karatsuba Multiplication

Larasati et al. [9] present a depth-reduction technique for the existing quantum circuit implementing Fermat's Little Theorem (FLT)−based inversion in the binary finite field. Their approach revolves around a "complete waterfall" strategy for translating Itoh−Tsujii's variant of FLT into the corresponding quantum circuit. This eliminates the need for inverse squaring operations in Banegas et al.'s work [1]. As detailed in [9], this modification significantly reduces the number of CNOT gates (CNOT count) and slightly decreases the circuit's T-depth. Consequently, the overall circuit depth and gate count are reduced, improving efficiency. This optimization complements adopting the enhanced Karatsuba multiplication algorithm proposed by Putranto et al. [12]. Their work, explained in detail within [12], demonstrates how these combined optimizations significantly reduce the overall resource requirements for quantum cryptanalysis circuits used in binary ECC. This study utilizes the work of Putranto et al. [12] on reconstructing improved FLT-based inversion with improved Karatsuba multiplication to develop an efficient circuit for binary elliptic curve cryptanalysis.

4 ECPM

This research builds upon established methodologies for quantum cryptanalysis of binary elliptic curves within the framework of Shor's algorithm. These methodologies draw inspiration from foundational works by Proos and Zalka [11], Roetteler et al. [14], Banegas et al. [1], and Putranto et

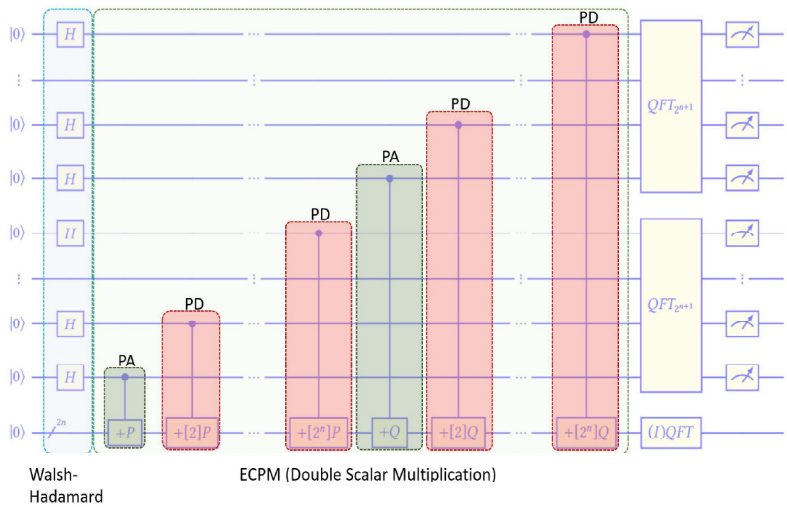


Fig. 2: The Quantum Circuit for Binary Elliptic Curve Cryptanalysis utilizes the Shor Approach and integrates Point Doubling (PD) for Elliptic Curve Point Multiplication (ECPM). We developed a quantum circuit for binary ECC cryptanalysis using a Shor-based approach, modified to include PD. This work builds on the original research by Roetteler et al. [14] and Banegas et al. [1] and incorporates additional PD optimizations as utilized in the study by Putranto et al. [12].

al. [12]. Besides elaborating on improved level 1-2 operations, the main contribution of this research is to integrate the most efficient variant of the PD design from previous work by [8] to quantum cryptanalysis of binary elliptic curves, which are currently only based on PA in [14], [1], and [12]. Figure 2 illustrates PD incorporation within a SHOR-based binary ECC cryptanalysis circuit in this study, which was previously explored without the application of PD. This modification is hypothesized to reduce the required quantum resource depth, thereby enhancing computational capabilities. As noted in [8]'s research, quantum computers are susceptible to various sources of error, including gate errors, decoherence, and crosstalk. Reducing circuit depth correlates with fewer gates and operations, minimizing the potential for error accumulation. Minimizing error rates is crucial in the context of error-corrected quantum computing.

4.1 Point Addition

For PA operation, our circuit construction employed an optimized Karatsuba multiplication algorithm (level 1) from Putranto et al. [12] and an improved FLT-based inversion technique (level 1) for efficiency. Putranto et al. evaluated these optimizations in Qiskit for $n = 8$ due to the lengthy output generated by Qiskit simulations for larger values of n [12]. In contrast, resource estimations for larger circuits (up to $n = 512$) were performed using simulation capabilities. This study elaborates on the integration of PA within binary ECC cryptanalysis circuit; we reconstruct prior work [12] to optimum PA in the Qiskit environment, as depicted in Figure 3 with specifications, utilizing an x64-based desktop PC (Intel i7-8700, six-core CPU) running Windows 10 Pro with



Fig. 3: Point Addition (PA) Circuit. The point addition circuit utilized in this study is derived from the research on [12], which incorporates an enhanced Karatsuba multiplication technique for level 1 operations and employs FLT-based inversion sourced from [12] [9].

64GB of RAM. Our software environment includes Python 3.9.7 and Qiskit version 0.30.0. The resulting gate resource estimation data is presented in a benchmarking table due to the significant size of the simulated circuits for single-step and total-step PA in Table 4.

4.2 Point Doubling

Recent research by Larasati et al. [8] emphasizes the significance of PD, particularly when the scalar k leads to $P = R$ (point coincidence). Unlike PA, which combines two distinct points on the elliptic curve, PD adds a point to itself $(P + P)$ to yield a new point $(2P)$. This operation plays a crucial role in scalar multiplication (kP), which necessitates a series of PA, as exemplified in the binary elliptic curve cryptanalysis circuit (Figure 2) [[14], [1], [12]]. This work delves into the analysis of PD operations presented by Larasati et al. [8] to establish the foundation for the ECPM circuit model that utilizes PD along with PA. We utilize Qiskit to simulate three model PD circuits on an x64-based desktop PC with an Intel i7-8700 six-core CPU running Windows 10 Pro, 64GB of RAM, Python 3.9.7, and Qiskit version 0.30.0.

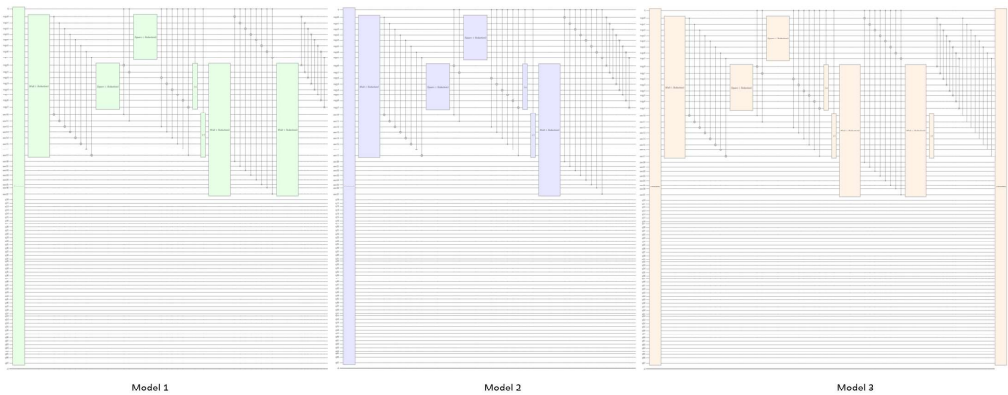


Fig. 4: Three-Model Point-Double (PD) Circuit. The circuits utilized in this study are based on the research presented in [7], which proposed three versions of PD circuit designs. This paper refers to these designs as Models 1 through 3. Model 1 features a design with one cleared ancilla register; Model 2 omits the uncomputation process; and Model 3 incorporates a complete uncomputation process.

We compared the design circuit proposed by Larasati et al. [8] regarding resource requirements and performance relative to existing PA-only-based circuits. Figure 4 depicts the Larasati et al. model implemented in our Qiskit environment. We comprehensively compared resource requirements for optimal PA and the three PD versions (number of qubits, depth, width, size, CCX (Toffoli) count, and CX (CNOT count) in Table 2.

Regarding formula complexity comparison, depicted in Table 3 Model 3 requires computations up to formula line 15, while Models 1 and 2 conclude at lines 12 and 10, respectively. In this work, we decompose the Point Addition (PA) step based on the foundational work of Banegas et al. [1]

Table 2: Resource Estimation for Point Addition (PA) and Three Models of Point Doubling (PD): Quantification of quantum resources required for a single operation of PA and each PD version, including Depth (number of sequential operations), Width (number of qubits involved), Size (number of quantum states), CCX (Toffoli) gates, and CX (CNOT) gates.

Group Operation		n	Qubit Counts	Overall Depth	Toffoli Depth	Overall Gate Count	Toffoli Count	CNOT Count
Point Addition		8	81	2,853	33	3,326	348	2968
		16	209	6,372	57	18,514	1,344	17,160
		127	2,668	9,672	498	1,624,081	57,143	1,566,928
		163	3,261	218,114	498	2,391,927	96,363	2,295,554
		233	4,894	1,259,306	708	4,961,503	152,099	4,809,394
		283	6,510	734,129	858	8,123,381	267,179	7,856,092
		409	9,408	293,0002	1,236	16,011,803	445,085	15,566,708
		571	14,847	1,716,035	1,722	35,775,039	935,947	34.839.082
Point Doubling	Model 1	8	89	1,235	24	2,224	232	1992
		16	225	4,262	40	11,536	842	10,694
		127	2,795	59,769	262	947,250	33,029	914,221
		163	3,424	134,043	334	1,419,228	57,357	1,361,871
		233	5,127	784,917	474	2,907,971	88,988	2,818,983
		283	6,793	427,698	574	4,691,980	154,661	4,537,319
		409	9,817	1,791,019	826	9,261,328	257,333	9,003,995
		571	15,418	9,988,487	1,150	20,287,244	531,049	19,756,195
	Model 2	8	89	1,089	23	1,969	205	1,764
		16	225	3,845	39	10,449	761	9,688
		127	2,795	146,313	261	2,465,023	85,469	2,379,554
		163	3,424	124,226	333	1,313,879	52,970	1,260,909
		233	5.127	724,650	473	2,703,752	82,665	2,621,087
		283	6,793	402,941	573	4,388,894	144,388	4,244,506
		409	9,817	1,663,168	825	8,651,040	240,232	8,410,808
		571	15,418	939,686	1,149	19,114,233	499,878	18,614,355
	Model 3	8	89	2,561	26	4,988	502	4,486
		16	225	9,296	42	27,772	1,976	25,796
		127	2,795	55,772	264	884,740	33,029	853,896
		163	3,424	340,122	336	3,622,619	145,097	3,477,522
		233	5,127	1,905,876	476	7,495,234	228,094	7,267,140
		283	6,793	1,101,299	576	12,252,297	401,213	11,851,084
		409	9,817	4,422,595	828	24,116,751	667,757	23,448,994
		571	15,418	2,512,026	1,149	53,815,007	1,403,837	52,411,170

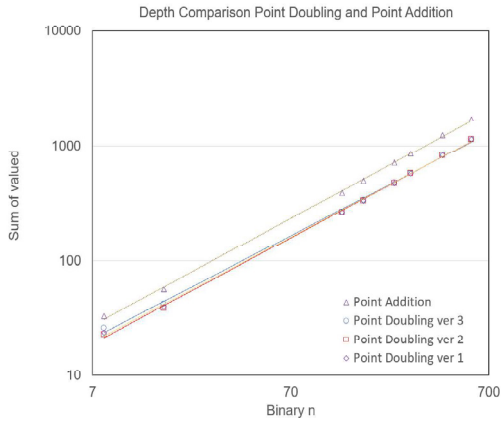


Fig. 5: Comparative Analysis of the Depth of Operations for Point Addition and three variations of Point Doubling. We calculate the interpolation of the value n and the sum value depth required for estimating the resource.

Table 3: Comparative analysis of formula complexity for point addition and three variations of point doubling.

Step	Point Addition		Point Doubling model 1		Point Doubling model 2		Point Doubling model 3	
	$q=1$	$q=0$	$q=1$	$q=0$	$q=1$	$q=0$	$q=1$	$q=0$
1	$x=x_1+x_2$	$x=x_1+x_2$	$anc_1=\frac{y_1}{x_1}$	$anc_1=\frac{y_1}{x_1}$	$anc_1=\frac{y_1}{x_1}$	$anc_1=\frac{y_1}{x_1}$	$anc_1=\frac{y_1}{x_1}$	$anc_1=\frac{y_1}{x_1}$
2	$y=y_1+y_2$	$y=y_1$	$y=0$	$y=y_1$	$y=0$	$y=y_1$	$y=0$	$y=y_1$
3	$\lambda=\frac{y_1+y_2}{x_1+x_2}$	$\lambda=\frac{y_1}{x_1+x_2}$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$	$anc_1=\frac{y_1}{x_1}+x_1=\lambda$
4	$y=0$	$y=0$	$y=\lambda^2$	$y=y_1$	$y=\lambda^2$	$y=y_1$	$y=\lambda^2$	$y=y_1$
5	$y=\lambda^2$	$y=\lambda^2$	$y=\lambda^2+\lambda$	$y=y_1$	$y=\lambda^2+\lambda$	$y=y_1$	$y=\lambda^2+\lambda$	$y=y_1$
6	$x=x_2+x_3$	$x=x_1+x_2$	$y=\lambda^2+\lambda+a=x_3$	$y=y_1$	$y=\lambda^2+\lambda+a=x_3$	$y=y_1$	$y=\lambda^2+\lambda+a=x_3$	$y=y_1$
7	$x=x_2+x_3$	$x=x_1+x_2$	$anc_1=\lambda+1$	$anc_1=\lambda+1$	$anc_1=\lambda+1$	$anc_1=\lambda+1$	$anc_1=\lambda+1$	$anc_1=\lambda+1$
8	$x=x_2+x_3$	$x=x_1+x_2$	$anc_2=(\lambda+1)x_3$	$anc_2=(\lambda+1)y_1$	$anc_2=(\lambda+1)x_3$	$anc_2=(\lambda+1)y_1$	$anc_2=(\lambda+1)x_3$	$anc_2=(\lambda+1)y_1$
9	$y=0$	$y=0$	$x=x_1^2$	$x=x_1$	$x=x_1^2$	$x=x_1$	$x=x_1^2$	$x=x_1$
10	$y=(x_1+x_2)+\lambda$	$y=y_1$	$x=x_1^2+(\lambda+1)x_3=y_3$	$x=x_1$	$x=x_1^2+(\lambda+1)x_3=y_3$	$x=x_1$	$x=x_1^2+(\lambda+1)x_3=y_3$	$x=x_1$
11	$\lambda=0$	$\lambda=0$	$anc_2=0$	$anc_2=0$	–	–	$anc_2=0$	$anc_2=0$
12	$x=x_3$	$x=x_1$	$swap: x=x_3.y_3$	$none: x=x_1.y_1$	–	–	$swap: x=x_3.y_3$	$none: x=x_1.y_1$
13	$y=y_2$	$y=y_1$	–	–	–	–	$anc_1=(\lambda+1)-1=\lambda$	$anc_1=\lambda$
14	–	–	–	–	–	–	$anc_1=\lambda$	$anc_1=\lambda-x_1=\frac{y_1}{x_1}$
15	–	–	–	–	–	–	$anc_1=\lambda$	$anc_1=0$

on reversible PAs, as the basis in [12]. Conversely, for Point-Doubling (PD), we present a high-level overview of the steps involved in the approach proposed by Larasati et al. [7]. This analysis facilitates a comparison of the complexity of each method for elliptic curve points. An elliptic curve point P_1 is represented as two binary polynomials x_1, y_1 stored in x, y of size n (where q denotes qubit control, λ array calculation with initialization state $|0\rangle$ and anc refer to ancillary qubit).

Based on those analyses, including visualized the depth comparison of PA and PD circuits (in Figure 5), in this study, we chose version 2 (lowest depth) for integration into the ECPM circuit to achieve optimal depth efficiency.

4.3 ECPM Results

By utilizing recent developments in finite field arithmetic for GF(2^m), specifically optimized algorithms for Karatsuba multiplication [12] and FLT-based inversion (Larasati et al. [9] or an alternative approach in [12]), we have incorporated these advancements into our quantum circuits for PA and PD. This integration results in an improved PA and model 2 of PD, with depth optimization. This study performed a comprehensive resource estimation, including qubits, gates, and circuit depth, comparing these findings with existing concrete binary ECC cryptanalysis studies that solely utilize PA circuits [12].

Table 4 presents a comprehensive comparison of the resource required for different ECPM implementations. This table compares our work, based on Qiskit simulations and resource estimation, and the work conducted by Putranto et al. [12] for $n-$bit ECC. The comparison specifically examines the qubits in columns ii and iii. The single-step ECPM comparison is presented in columns v and vi, where ECPM with combined PA or PD is compared against the single-step PA algorithm (from PUT). The result from our recent study on total-step ECPM ($2n$ PD + 2 PA) revealed reduced quantum resource usage in column vii, and we compared it to recent PUT results in column viii in scenarios that exclusively utilize point addition ($2n+2$ PA).

Table 4: Comparison of Resource Analysis of Single Step ECPM (PA or PD) and Point Addition (1 PA) and in Total Step ECPM($2n$ PD +2 PA) Point Addition ($2n + 2$ PA) for n bit ECC with Previous Work. In both Toffoli and CNOT gates, qubit and Depth, This Work Based on Qiskit Simulation Result. Note that, PUT (Point Addition algorithm by Putranto et al. [12]).

n	Ours	PUT	Ours (1 ECPM (PA or PD))			PUT (1 PA)			Ours ($2n$PD + 2PA)	PUT ($2n+2$ PA)
	qubits	qubits	Toffoli	CNOT	depth	Toffoli	CNOT	depth	Toffoli	Toffoli
i	ii	iii	v			vi			vii	viii
8	89	74	205	1,764	23	348	2,734	210	3,976	6,264
16	225	194	761	9,688	39	1,344	15,924	623	27,040	45,696
127	2,795	2,320	85,469	2,379,554	261	52,773	1,352,497	7,010	21,823,412	13,509,888
163	3,424	2,805	52,970	1,260,909	333	96,299	2,137,063	14,632	17,460,946	31,586,072
233	5,127	4,228	82,665	2,621,087	473	152,067	4,480,745	46,702	38,826,088	71,167,356
283	6,793	5,694	144,388	4,244,506	573	267,115	7,376,571	34,792	82,257,966	151,721,320
409	9,817	8,214	240,232	18,410,808	825	445,021	14,565,173	111,858	197,399,946	364,917,220
571	15,418	13,167	499,878	18,614,355	1,149	935,883	32,888,178	93,142	572,732,570	1,070,650,152

5 Conclusion

This work advances resource estimation for Elliptic Curve Point Multiplication (ECPM) in the context of quantum cryptanalysis of binary ECC. We leveraged recent advancements in finite field arithmetic for GF(2^m) by incorporating the improved Karatsuba multiplication algorithm from Putaranto et al. [12] and the FLT-based inversion algorithm from Larasati et al. [9] (or the combination approach presented in [12]) into our quantum circuits for both point addition (PA) and point doubling (PD). Additionally, we integrate an optimized PA algorithm from prior research [12] into the ECPM construction. Furthermore, we analyzed three PD circuit versions by Larasati et al. [8] and selected Model 2 as the most depth-efficient model for our binary ECC cryptanalysis scheme based on Shor's algorithm. Finally, we performed a comprehensive resource estimation, including qubits, gates, and circuit depth. In the final comparisons, we compared ECPM as single-step, ECPM with combination or selection in PA/PD utilization, and total-step ECPM ($2n$ PD +2 PA). We compare these findings with existing concrete binary ECC cryptanalysis studies that solely utilize PA circuits [12], highlighting the depth optimization achieved through the proposed approach.

References

1. Banegas, G., Bernstein, D.J., van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 451–472 (2021)
2. Chen, L., Moody, D., Regenscheid, A., Randall, K.: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters. Tech. rep., National Institute of Standards and Technology (2019)
3. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of elliptic and hyperelliptic curve cryptography. CRC press (2005)
4. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms. In: Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11. pp. 425–444. Springer (2020)
5. van Hoof, I.: Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic toffoli gate count. arXiv preprint arXiv:1910.02849 (2019)

6. Kerry, C.F., Gallagher, P.D.: Digital signature standard (dss). FIPS PUB pp. 186–4 (2013)
7. Larasati, H.T., Kim, H.: Quantum cryptanalysis landscape of shor's algorithm for elliptic curve discrete logarithm problem. In: Information Security Applications: 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11–13, 2021, Revised Selected Papers 22. pp. 91–104. Springer (2021)
8. Larasati, H.T., Kim, H.: Quantum circuit designs of point doubling operation for binary elliptic curves. In: International Conference on Information Security Applications. pp. 297–309. Springer (2023)
9. Larasati, H.T., Putranto, D.S.C., Wardhani, R.W., Park, J., Kim, H.: Depth optimization of flt-based quantum inversion circuit. IEEE Access (2023)
10. Mohammadi, M., Molahosseini, A.S.: Efficient design of elliptic curve point multiplication based on fast montgomery modular multiplication. In: ICCKE 2013. pp. 424–429. IEEE (2013)
11. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. arXiv preprint quant-ph/0301141 (2003)
12. Putranto, D.S.C., Wardhani, R.W., Larasati, H.T., Ji, J., Kim, H.: Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access (2023)
13. Rescorla, E.: The transport layer security (tls) protocol version 1.3. RFC 8446, RFC Editor (August 2018)
14. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Advances in Cryptology–AsiaCrypt 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23. pp. 241–270. Springer (2017)
15. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. Ieee (1994)

Integral Attack with Bit-based Division Property on the Lightweight Block Cipher LBC

Naoki Shibayama and Yasutaka Igarashi

Tokyo University of Science
2641 Yamazaki, Noda, Chiba, 278-8510, Japan
{7323703@ed,yasutaka@rs}.tus.ac.jp

Abstract. LBC is the lightweight block cipher proposed by Kapalova *et al.* in 2023. The block size is 64 bits, the secret key size is 80 bits, and the number of rounds is 20, respectively. The designer analyzed the avalanche effect and evaluated the security against differential and linear cryptanalysis. On the other hand, the security against integral cryptanalysis, one of the most powerful cryptanalyses on block cipher, has yet to be evaluated. In this paper, we evaluated the security of LBC by applying integral cryptanalysis. We investigated the integral characteristics with the bit-based division property using Mixed Integer Linear Programming (MILP). Consequently, we found that LBC has the 18-round integral characteristic using the 60-th order differential. Exploiting the discovered 16-round integral characteristic using the 48-th order differential, the integral attack on the full-round LBC can be efficiently performed with $2^{50.8}$ blocks of chosen plaintext and $2^{51.4}$ times of encryption operation. Then, we improve the round function of LBC and discuss its security against integral attack.

Keywords: Cryptanalysis · Integral attack · Bit-based division property · Block cipher · LBC.

1 Introduction

Information and communication technology has developed remarkably in recent years, including the Internet of Things (IoT) system, in which everything is connected. A lightweight block cipher is an effective solution to provide security in resource-constrained environments, such as low power consumption, small footprint, and low computational complexity.

LBC [1] is a 64-bit block cipher with an 80-bit secret key designed by Kapalova *et al.* in 2023. The recommended number of rounds is 20. In [1], the avalanche effect of LBC was investigated, and its properties were described as similar to random numbers. Then, the security against differential and linear attacks were evaluated, and LBC was reported to resist these attacks. Meanwhile, the security against integral attack has not been evaluated. The integral attack [2] is cryptanalysis proposed by Knudsen *et al.*, which focuses on the XOR sum property of a data set and can be widely applied to symmetric key cryptographic algorithms. Todo *et al.* [3], [4] proposed a method for finding

better integral characteristics: division property (DP), which enables the use of algebraic degrees, and bit-based division property (BDP). Subsequently, Xiang *et al.* [5] proposed an efficient method for searching integral characteristics by BDP using MILP.

In this paper, we evaluate the security of LBC against integral attack. We explored the integral characteristics by BDP using MILP and found that LBC has 18-round integral characteristics. Moreover, exploiting the 16-round integral characteristic using the 48-th order differential, an integral attack can be efficiently performed on the full-round LBC with $2^{50.8}$ blocks of chosen plaintext and $2^{51.4}$ times of encryption operation. In order to overcome this weakness, we attempt to improve the round function of LBC. Specifically, we added a linear layer to the original one. As a result, the length of the integral characteristic was reduced to 8 rounds in the improved LBC, which strengthens the original LBC algorithm.

The remainder of this paper is organized as follows. Section 2 explains the algorithm of LBC. Section 3 gives an overview of the integral attack. Section 4 explains the division property and the method for searching the integral characteristic using MILP. Section 5 shows the results of the integral characteristics of LBC. Then, we perform the integral attack to full-round LBC in section 6. Section 7 presents the improved LBC and describes its security against integral attack. Section 8 finally concludes the paper.

2 The Algorithm of LBC

This section briefly describes the structure of LBC. LBC has a 64-bit block size and an 80-bit secret key size. Encryption iterations are performed for 20 rounds. It consists of the key addition, the nonlinear transformation, and the left cyclic shift. Fig. 1 shows the data processing part of LBC. In the figure, the symbol $\oplus$ represents an Exclusive-OR (XOR) operation, and the symbol S denotes four parallel 4-bit S-boxes, which are bijective and nonlinear. The 4-bit S-box s_4 is shown in Table 1 in hexadecimal format. Also, RL is a linear function given by

$$\mathrm{RL}(X) = X \oplus (X \lll 7) \oplus (X \lll 10), \tag{1}$$

where $X \in \mathrm{GF}(2)^{16}$, $\lll \ell$ is a left cyclic shift by ℓ bits. Let $\mathbf{X}^{(i)}$=$(X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)})$, $X_j^{(i)}$=$(x_{j,1}^{(i)}, x_{j,2}^{(i)}, x_{j,3}^{(i)}, x_{j,4}^{(i)})$, $x_{j,\ell}^{(i)} \in \mathrm{GF}(2)^4$ be the input of the i-th round, and $\mathbf{C}^{(i)}$=$(C_1^{(i)}, C_2^{(i)}, C_3^{(i)}, C_4^{(i)})$, $C_j^{(i)}$=$(c_{j,1}^{(i)}, c_{j,2}^{(i)}, c_{j,3}^{(i)}, c_{j,4}^{(i)})$, $c_{j,\ell}^{(i)} \in \mathrm{GF}(2)^4$ be the output. Note that $1 \le i \le 20$, $1 \le j \le 4$, and $1 \le \ell \le 4$. The data processing part generates 64-bit ciphertext $\mathbf{C}^{(20)}$ from 64-bit plaintext $\mathbf{X}^{(1)}$, 64-bit whitening key $\mathbf{WK}$=(WK_1, WK_2, WK_3, WK_4), $WK_j \in \mathrm{GF}(2)^{16}$, and twenty 64-bit round keys $\mathbf{RK}^{(i)} = (RK_1^{(i)}, RK_2^{(i)}, RK_3^{(i)}, RK_4^{(i)})$, $RK_j^{(i)} = (rk_{j,1}^{(i)}, rk_{j,2}^{(i)}, rk_{j,3}^{(i)}, rk_{j,4}^{(i)})$, $rk_{j,\ell}^{(i)} \in \mathrm{GF}(2)^4$.

Since our attack does not use the relationship between the round keys, we omit a description of the key schedule. Please refer to [1] for detailed specifications.

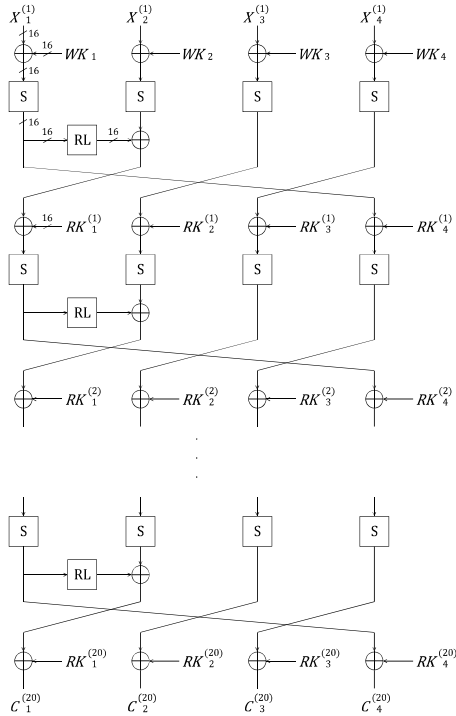


Fig. 1. Data processing part of LBC.

Table 1. S-box s_4.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$s_4(x)$	9	2	A	4	0	6	7	D	5	1	8	3	E	F	B	C

3 Integral Attack

The integral attack [2] is a cryptanalysis proposed by Knudsen and Wagner in 2002. We describe the definition of integral properties related to this paper.

Integral Property We define the following four properties of the set $\{X_i | X_i \in \{0,1\}^\ell, 0 \le i < 2^\ell\}$ of ℓ-bit data X.

Constant (C) : if $\forall i, j \;\; X_i = X_j$,

All (A) : if $\forall i, j \; (i \neq j) \;\; X_i \neq X_j$,

Balance (B) : if $\bigoplus_i X_i = 0$,

Unknown (U) : Others.

In this paper, for example, if the integral property of 1-nibble data $x_{j,\ell}^{(i)}$ is B, we denote this as $\{x_{j,\ell}^{(i)}\}$ = B, where $1 \le i \le 20$, $1 \le j \le 4$, and $1 \le \ell \le 4$.

In addition, as an example, if the integral property of 4-nibble data $(x_{j,1}^{(i)}, x_{j,2}^{(i)}, x_{j,3}^{(i)}, x_{j,4}^{(i)})$ is (B B B U), then we represent this as $\{(x_{j,1}^{(i)}, x_{j,2}^{(i)}, x_{j,3}^{(i)}, x_{j,4}^{(i)})\} = (\mathrm{B}\,\mathrm{B}\,\mathrm{B}\,\mathrm{U})$. Furthermore, if the integral property of each nibble data is the same, for example, $\{(x_{j,1}^{(i)}, x_{j,2}^{(i)}, x_{j,3}^{(i)}, x_{j,4}^{(i)})\} = (\mathrm{B}\,\mathrm{B}\,\mathrm{B}\,\mathrm{B})$, this can be expressed as $\{X_j^{(i)}\} = \mathbf{B}$.

The attacker encrypts a set of chosen plaintexts and investigates the propagation of the integral property. Suppose 2^ℓ chosen plaintexts are input, and the integral property of the output from a reduced r-round cipher is All, Balance, or Constant. Then, we say the cipher has an r-round integral characteristic using the ℓ-th order differential. By exploiting the characteristic that the XOR sum is 0, the attacker derives an attack equation to estimate and determine the key by solving it.

4 MILP-aided Bit-based Division Property

Todo *et al.* [3], [4] generalized the integral property and proposed the division property (DP), which can utilize algebraic degrees. Then, the bit-based division property (BDP) was devised to enable the bitwise operation of DP.

4.1 Bit Product Function and BDP

This subsection presents definitions of the bit product function and BDP.

Bit Product Function For any $a \in \mathrm{GF}(2)^n$, let $a[i]$ be the i-th bit of a, and the Hamming weight of a is calculated as $w(a) = \sum_{i=1}^{n} a[i]$. For any n-bit variable $u \in \mathrm{GF}(2)^n$ and input variable $x \in \mathrm{GF}(2)^n$, the bit product function $\pi_u(x)$ is defined by

$$\pi_u(x) := \prod_{i=1}^{n} x[i]^{u[i]}, \tag{2}$$

where $x[i]^0 = 1$ and $x[i]^1 = x[i]$.

Next, for an arbitrary m-dimensional vector $\boldsymbol{u} = (u_1, u_2, \cdots, u_m)$, $u_i \in \mathrm{GF}(2)$, an input vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$, a bit product function $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ for $x_i \in \mathrm{GF}(2)$ is defined as follows.

$$\pi_{\boldsymbol{u}}(\boldsymbol{x}) := \prod_{i=1}^{m} \pi_{u_i}(x_i) \tag{3}$$

BDP Let $\mathbb{X}$ be a multiset whose elements are m-dimensional vectors $\boldsymbol{x}$, and we call $\mathbb{X}$ has BDP $D_{\mathbb{K}}^{1^m}$ if the set $\mathbb{X}$ satisfies the following equation.

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \begin{cases} \text{Unknown, if there is } \boldsymbol{k} \in \mathbb{K} \text{ s.t. } W(\boldsymbol{u}) \succeq \boldsymbol{k}, \\ 0 \qquad \text{, otherwise,} \end{cases}$$

where $W(\boldsymbol{u}) = (w(u_1), w(u_2), \cdots, w(u_m))$, and we denote $\boldsymbol{k'} \succeq \boldsymbol{k}$ if $k'_i \geq k_i$ for all i. Note that $n_1, n_2, \cdots, n_m$ are restricted to 1 when we consider BDP $D_{\mathbb{K}}^{1^m}$.

4.2 Division Trail

Consider an n-bit BDP passing through the round function until the r-th round, which changes as shown in the following equation.

$$D_{\mathbb{K}_0}^{1^n} \to D_{\mathbb{K}_1}^{1^n} \to \cdots \to D_{\mathbb{K}_r}^{1^n} \tag{4}$$

At this point, for any $\overrightarrow{k_i^*} \in \mathbb{K}_i$ in $\mathbb{K}_i\,(1 \le i \le r)$, a propagation vector $\overrightarrow{k_{i-1}^*}$ in $\mathbb{K}_{i-1}$ exists. Therefore, the propagation of $\overrightarrow{k_i^*}$ can be expressed as in Eq. (5), called the division trail.

$$\overrightarrow{k_0^*} \to \overrightarrow{k_1^*} \to \cdots \to \overrightarrow{k_r^*} \tag{5}$$

4.3 MILP-aided Bit-based Division Property

In 2016, Xiang *et al.* [5] proposed a technique that combines the BDP and searches for the division trails by employing the MILP. We use it to search for the integral characteristics of LBC.

Constraints for Copy, XOR and AND We denote a bit-wise linear inequalities model of Copy, XOR and AND, and show how to model BDP propagations of S-box by linear inequalities. For more details, please refer to [5].

Model 1 (Copy) Let $(a) \xrightarrow{\text{Copy}} (b_1, b_2)$ be a division trail of Copy. The following inequalities are sufficient to describe the propagation of the division property for Copy.

$$\begin{cases} a - b_1 - b_2 = 0, \\ a, b_1, b_2 \text{ are binaries.} \end{cases}$$

Model 2 (XOR) Let $(a_1, a_2) \xrightarrow{\text{XOR}} (b)$ be a division trail or XOR. The following inequalities are sufficient to describe the propagation of the division property for XOR.

$$\begin{cases} a_1 + a_2 - b = 0, \\ a_1, a_2, b \text{ are binaries.} \end{cases}$$

Model 3 (AND) Let $(a_1, a_2) \xrightarrow{\text{AND}} (b)$ be a division trail of AND. The following inequalities are sufficient to describe the propagation of the division property for AND.

$$\begin{cases} b - a_1 \ge 0, \\ b - a_2 \ge 0, \\ b - a_1 - a_2 \le 0, \\ a_1, a_2, b \text{ are binaries.} \end{cases}$$

Model 4 (S-box) Each division trail of an n-bit S-box can be viewed as a $2n$-dimensional vector in $\{0,1\}^{2n} \subset \mathbb{R}^{2n}$ where $\mathbb{R}$ is the real number field. Thus, all division trails form a subset P of $\{0,1\}^{2n}$. Next, we compute the H-representation of $\text{Conv}(P)$ using the $inequality_generator()$ function Sagemath [6], which returns a set of linear inequalities $\mathcal{L}$.

Table 2. r-round integral characteristics of LBC using d-th order differential.

d	r	Integral property of $\mathbf{C}^{(r)}$
1−3	6	**U** (B B B U) **U U**
4−11	7	**U B U U**
12−15	8	
16−19	9	
20−23	10	
24−27	11	
28−35	12	
36−39	13	
40−43	14	
44−47	15	
48−55	16	
56−59	17	
60−63	18	

Objective Function The vector $\overrightarrow{k_r^*}$ in the r-th round of the division trail can be expressed as Eq. (6).

$$\overrightarrow{k_r^*} = (a_0^r, a_1^r, \cdots, a_{n-1}^r) \tag{6}$$

The objective function is then given by Eq. (7).

$$obj = Min\{a_0^r + a_1^r + \cdots + a_{n-1}^r\} \tag{7}$$

Here, when the value of the objective function is 1, a unit vector is in the set $\mathbb{K}_r$ of the r-th round. It means that the bitwise integral property ' U ' exists.

5 Integral Characteristics of LBC

The results of the BDP investigation are shown in Table 2 using Gurobi [7], one of the MILP solvers. In the table, note that d represents the order of differential, and r denotes the number of rounds of the integral characteristic. However, for the $d\,(1 \leq d \leq 63)$-th order differential, the bits from the right to the d-th of the plaintext are set to All. A comprehensive examination using the 63-rd order differential revealed that the 19-round integral characteristic was not found, so the 18-round integral characteristic using the 60-th order differential, the longest with the lowest order of differential, was the best.

6 Integral Attack on the full-round LBC

By exploiting the 16-round integral characteristic of LBC using the 48-th order differential, the integral attack on the full-round LBC is possible efficiently. This section estimates the number of chosen plaintexts and the computational complexity required for the attack.

6.1 Attack Equation

Fig. 2 shows the equivalent circuit from the 20-th round output ($C_1^{(20)}, C_2^{(20)}$, $C_4^{(20)}$) to the 16-th round output $C_2^{(16)}$. The S^{-1} in the figure shows four inverse functions of S-box s_4 in parallel. RL is a linear operation, and the 16-bit round key $RK_4^{(i)}$ of the i-th round can be moved to directly after the S-box, which is a nonlinear operation. In addition, the round key $RK_1^{(17)}$, $RK_4^{(17)}$, $RK_1^{(20)}$, and $RK_4^{(20)}$ are moved equivalently, as shown in Fig. 2, to reduce the attack cost. Note that $RK'^{(i)}_1 = RK_1^{(i)} \oplus \mathrm{RL}(RK_4^{(i)})$. Let $H^{(i)} = (h_1^{(i)}, h_2^{(i)}, h_3^{(i)}, h_4^{(i)})$, $h_\ell^{(i)} \in \mathrm{GF}(2)^4$ represents the data after adding the key $RK'^{(i)}_1$ of the i-th round as the intermediate variable, and $Z^{(i)} = (z_1^{(i)}, z_2^{(i)}, z_3^{(i)}, z_4^{(i)})$, $z_\ell^{(i)} \in \mathrm{GF}(2)^4$ denotes the data after the output of RL. Furthermore, RL is expressed as follows.

$$\mathrm{RL} = \begin{bmatrix} m_1\ m_2\ m_3\ m_4 \\ m_4\ m_1\ m_2\ m_3 \\ m_3\ m_4\ m_1\ m_2 \\ m_2\ m_3\ m_4\ m_1 \end{bmatrix}, \tag{8}$$

$$m_1 = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix},\ m_2 = \begin{bmatrix} 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix},\ m_3 = \begin{bmatrix} 0\,0\,1\,0 \\ 1\,0\,0\,1 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix},\ m_4 = \begin{bmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix}.$$

Examining the subsequent change in the integral property ‘ B ,’ which appears in $c_{2,\ell}^{(16)}$ $(1 \le \ell \le 4)$ in the 16-round integral characteristic using the 48-th order differential, we find that this passes through the S-box s_4 in the 17-th round. Then, the key $rk'^{(17)}_{1,\ell}$ is added, and the following integral property holds for the intermediate variable $h_\ell^{(17)}$.

$$\{h_\ell^{(17)}\} = \mathrm{B}$$

Therefore, from its XOR sum property, by applying the meet-in-the-middle technique [8], the attack equation is given by

$$\bigoplus z_\ell^{(17)} = \bigoplus c_{1,\ell}^{(17)}, \tag{9}$$

where

$$^{\mathrm{T}}(z_1^{(17)}, z_2^{(17)}, z_3^{(17)}, z_4^{(17)}) = \mathrm{RL}\,^{\mathrm{T}}(c_{4,1}^{(17)}, c_{4,2}^{(17)}, c_{4,3}^{(17)}, c_{4,4}^{(17)}), c_{4,\ell}^{(17)} = s_4^{-1}(c_{3,\ell}^{(18)} \oplus rk_{3,\ell}^{(18)}),$$
$$c_{1,\ell}^{(17)} = s_4^{-1}(c_{4,\ell}^{(18)} \oplus rk_{4,\ell}^{(18)}), c_{3,\ell}^{(18)} = s_4^{-1}(c_{2,\ell}^{(19)} \oplus rk_{2,\ell}^{(19)}), c_{4,\ell}^{(18)} = s_4^{-1}(c_{3,\ell}^{(19)} \oplus rk_{3,\ell}^{(19)}),$$
$$c_{2,\ell}^{(19)} = s_4^{-1}(h_\ell^{(20)} \oplus rk'^{(20)}_{1,\ell}), c_{3,\ell}^{(19)} = s_4^{-1}(c_{2,\ell}^{(20)} \oplus rk_{2,\ell}^{(20)}), h_\ell^{(20)} = z_\ell^{(20)} \oplus c_{1,\ell}^{(20)},$$
$$^{\mathrm{T}}(z_1^{(20)}, z_2^{(20)}, z_3^{(20)}, z_4^{(20)}) = \mathrm{RL}\,^{\mathrm{T}}(c_{4,1}^{(20)}, c_{4,2}^{(20)}, c_{4,3}^{(20)}, c_{4,4}^{(20)}).$$

Note that s_4^{-1} is the inverse function of S-box s_4, and T is the transposition of a vector or matrix. To compute the attack equation in Eq. (9), we need the output

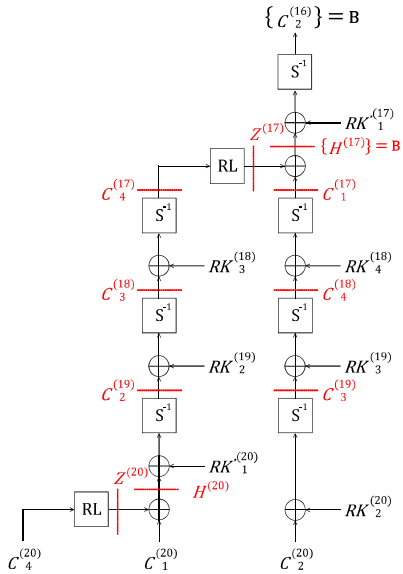


Fig. 2. Equivalent circuit from the 20-th round output $(C_1^{(20)}, C_2^{(20)}, C_4^{(20)})$ to the 16-th round output $C_2^{(16)}$.

of 32-bit $(C_1^{(20)}, C_4^{(20)})$ on the left side and 16-bit $C_2^{(20)}$ on the right side. Here, the keys to be identified are the six 16-bit keys $RK_3^{(18)}$, $RK_4^{(18)}$, $RK_2^{(19)}$, $RK_3^{(19)}$, $RK'^{(20)}_1$, and $RK_2^{(20)}$ located in the equivalent circuit from $(C_1^{(20)}, C_2^{(20)}, C_4^{(20)})$ to $H^{(17)}$ in Fig. 2, i.e., 96-bit keys in total.

In this paper, to improve the efficiency of the attack, we use the Mod2 Frequency Distribution Table (MFDT) of $z_\ell^{(17)}$ and $c_{1,\ell}^{(17)}$, which are derived sequentially by the partial sum technique [9] proposed by Ferguson *et al.*, and $\bigoplus z_\ell^{(17)}$ and $\bigoplus c_{1,\ell}^{(17)}$ are calculated, respectively. Then, the true key is identified by examining Eq. (9).

6.2 Derivation of MFDT of $Z^{(17)}$

In this subsection, we show the procedure for sequentially deriving the MFDT of the intermediate variable $Z^{(17)}$ of the 17-th round from the MFDT of the intermediate variable $H^{(20)}$ of the 20-th round in Fig. 2.

Fig. 3 shows the equivalent circuit from $H^{(20)}$ to $Z^{(17)}$ expressed in nibble values. As shown in the figure, the nibble variables after the XOR operation at the two locations directly before $z_\ell^{(17)}$ $(1 \le \ell \le 4)$ are v_ℓ and w_ℓ from the left. First, the MFDT of $z_\ell^{(17)}$ can be derived from the 8-bit MFDT of $(w_\ell, h_4^{(20)})$ by assuming the 12-bit key $rk'^{(20)}_{1,4}, rk_{2,4}^{(19)}$, and $rk_{3,4}^{(18)}$. The MFDT of $(w_\ell, h_4^{(20)})$ can also be derived from the 12-bit MFDT of $(v_\ell, h_3^{(20)}, h_4^{(20)})$ by assuming the 12-bit

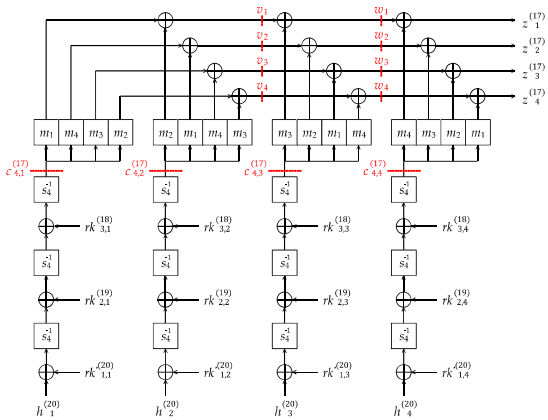


Fig. 3. Equivalent circuit from the 20-th round intermediate variable $h_\ell^{(20)}$ $(1 \le \ell \le 4)$ to the 17-round intermediate variable $z_\ell^{(17)}$.

key $rk'^{(20)}_{1,3}, rk^{(19)}_{2,3}$, and $rk^{(18)}_{3,3}$. Furthermore, the MFDT of $(v_\ell, h_3^{(20)}, h_4^{(20)})$ can be derived from the total 16-bit MFDT of $(h_1^{(20)}, h_2^{(20)}, h_3^{(20)}, h_4^{(20)})$ by assuming the 24-bit key $rk'^{(20)}_{1,1}, rk^{(19)}_{2,1}, rk^{(18)}_{3,1}, rk'^{(20)}_{1,2}, rk^{(19)}_{2,2}$, and $rk^{(18)}_{3,2}$.

6.3 Attack Algorithm

The attacker recovers the 96-bit round keys according to the following procedure.

Step1. Encrypt 2^{48} plaintexts of 48-bit data $(X_2^{(1)}, X_3^{(1)}, X_4^{(1)})$ whose integral property is All.

Step2. From the ciphertext corresponding to the plaintext above, the 16-bit MFDT $(\texttt{MFDT}(H^{(20)}))$ of $H^{(20)} (= C_1^{(20)} \oplus \mathrm{RL}(C_4^{(20)}))$ and the four 4-bit MFDTs $(\texttt{MFDT}(c_{2,\ell}^{(20)}))$ of $c_{2,\ell}^{(20)}$ $(1 \le \ell \le 4)$ are derived.

Step3. By assuming the 24-bit key $rk'^{(20)}_{1,1}, rk^{(19)}_{2,1}, rk^{(18)}_{3,1}, rk'^{(20)}_{1,2}, rk^{(19)}_{2,2}$, and $rk^{(18)}_{3,2}$, the 16-bit MFDT $(\texttt{MFDT}(c_{4,1}^{(17)}, c_{4,2}^{(17)}, h_3^{(20)}, h_4^{(20)}))$ of $(c_{4,1}^{(17)}, c_{4,2}^{(17)}, h_3^{(20)}, h_4^{(20)})$ can calculate from $\texttt{MFDT}(H^{(20)})$.

Step4. Using $\texttt{MFDT}(c_{4,1}^{(17)}, c_{4,2}^{(17)}, h_3^{(20)}, h_4^{(20)})$ and the linear operations m_ℓ, calculate the four 12-bit MFDTs $(\texttt{MFDT}(v_\ell, h_3^{(20)}, h_4^{(20)}))$ of $(v_\ell, h_3^{(20)}, h_4^{(20)})$.

Step5. By assuming the 12-bit key $rk'^{(20)}_{1,3}, rk^{(19)}_{2,3}$, and $rk^{(18)}_{3,3}$, the four 12-bit MFDTs $(\texttt{MFDT}(v_\ell, c_{4,3}^{(17)}, h_4^{(20)}))$ of $(v_\ell, c_{4,3}^{(17)}, h_4^{(20)})$ can calculate from $\texttt{MFDT}(v_\ell, h_3^{(20)}, h_4^{(20)})$.

Step6. Using $\texttt{MFDT}(v_\ell, c_{4,3}^{(17)}, h_4^{(20)})$ and the linear operations m_ℓ, calculate the four 8-bit MFDTs $(\texttt{MFDT}(w_\ell, h_4^{(20)}))$ of $(w_\ell, h_4^{(20)})$.

Step7. By assuming the 12-bit key $rk'^{(20)}_{1,4}, rk^{(19)}_{2,4}$, and $rk^{(18)}_{3,4}$, the four 8-bit MFDTs $(\texttt{MFDT}(w_\ell, c_{4,4}^{(17)}))$ of $(w_\ell, c_{4,4}^{(17)})$ can calculate from $\texttt{MFDT}(w_\ell, h_4^{(20)})$.

Step8. Using $\texttt{MFDT}(w_\ell, c_{4,4}^{(17)})$ and the linear operations m_ℓ, calculate the four 4-bit MFDTs ($\texttt{MFDT}(z_\ell^{(17)})$) of $z_\ell^{(17)}$.
Step9. By assuming the four 12-bit keys $rk_{2,\ell}^{(20)}, rk_{3,\ell}^{(19)}$, and $rk_{4,\ell}^{(18)}$, the four 4-bit MFDTs ($\texttt{MFDT}(c_{1,\ell}^{(17)})$) of $c_{1,\ell}^{(17)}$ can calculate from $\texttt{MFDT}(c_{2,\ell}^{(20)})$.
Step10. From $\texttt{MFDT}(z_\ell^{(17)})$ and $\texttt{MFDT}(c_{1,\ell}^{(17)})$, calculate the value of $\bigoplus z_\ell^{(17)}$ and $\bigoplus c_{1,\ell}^{(17)}$, respectively. Then, an attacker can judge whether the key is correct by examining Eq. (9) holds.
Step11. The 32-bit key $RK_3^{(18)}$ and $RK_4^{(18)}$ in the 18-th round has been determined so far; the remaining 48 $(= 80 - 32)$ bits key in the 80-bit key register of the 18-th round of the key schedule is identified by an exhaustive key search.

6.4 Complexity Estimation

Since Eqs. (9) are four 4-bit attack equations, the probability that it is satisfied for a false key is $(2^{-4})^4 = 2^{-16}$. Thus, to reduce the number of 2^{96} candidate keys to the true key, it is sufficient to have $7 (> \frac{96}{16})$ sets of 48-th order differential. Therefore, the number of chosen plaintexts required for the attack is $7 \cdot 2^{48} \approx 2^{50.8}$.

Next, we consider the computational complexity of recovering all keys. From the attack algorithm, the overall computational complexity $\mathbf{T}$ is given by

$$\begin{aligned}
&\mathbf{T} = T_1 + T_2 + T_3 \approx 2^{50.8} + 2^{49.3} + 2^{48} \approx 2^{51.4},\\
&T_1 = 2^{50.8},\\
&T_2 = 2^{24}(2^{16} \cdot 6 + 2^{12}(2^{12} \cdot 3 + 2^{12}(2^8 \cdot 3))) + (2^{12} \cdot 2^4 \cdot 3) \cdot 4 \approx 2^{57.6}\,(\text{S-box}),\\
&T_3 = 2^{48},
\end{aligned}$$

where T_1 is the computational complexity of 7 sets of 48-th order differential, T_2 and T_3 are the computational complexity required to determine the 96-bit key by solving the attack equation and exhaustive key search for the remaining 48-bit key, respectively. Because the full-round LBC consists of 320 $(= 16 \times 20)$ S-boxes, the computational complexity of T_2 is equivalent to $2^{57.6}/320 \approx 2^{49.3}$ times of encryption operation.

Table 3 shows the results of applying an integral attack on the full-round LBC exploiting different integral characteristics. Moreover, the attack exploiting the 15-round or less characteristic is ineffective because the computational complexity required for the attack exceeds 2^{80}, and the attack exploiting the 16-round integral characteristic is the best from the table.

7 Improvement of LBC

Due to the low diffusion property, LBC has more than 16-round integral characteristics, and these can be exploited for the integral attack on the full-round LBC. This section attempts to strengthen the LBC algorithm and discusses its security against integral attack.

Table 3. Results of integral attacks on the full-round LBC.

d	r	Guess Keys	Data Complexity	Time Complexity
60	18	32	$2^{61.6}$	$2^{61.6}$
56	17	64	$2^{58.3}$	$2^{58.3}$
48	16	96	$2^{50.8}$	$2^{51.4}$

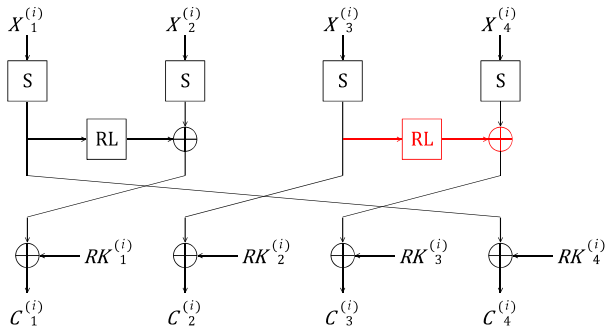


Fig. 4. Round function of the improved LBC.

7.1 The Algorithm

Fig. 4 shows the improved LBC's round function. To introduce the better diffusion property, an XOR operation from the third 16-bit data, which applied the linear function RL, to the fourth 16-bit data (in the red-colored part of Fig. 4) is added after the S-boxes.

7.2 Security against Integral Attack

In this subsection, we evaluate the security of the improved LBC against integral attack.

The integral characteristics of the improved LBC were investigated using the search method shown in Section 5. As a result, we found the 8-round integral characteristics shown in Table 4 to be the longest. In the table, d denotes the order of differential. Since the 9-round integral characteristic was not confirmed in the comprehensive search using the 63-rd order differential, it was revealed that the 8-round integral characteristic using the 52-nd order differential was the best in terms of efficient attack. Thus, its length is reduced to 8 rounds compared to the 18-round integral characteristic discovered in LBC so that the algorithm of the improved LBC is confirmed to be stronger than the original one.

We apply the integral attack on the improved LBC by exploiting the 8-round integral characteristic using the 52-nd order differential. Although the details of the attack algorithm are omitted here due to the space limit, its key recovery is executed similarly, as shown in Section 6. We confirmed that the improved 11-round LBC can be attacked more efficiently than an exhaustive key search.

Table 4. 8-round integral characteristics using d-th order differential of the improved LBC.

d	Integral property of $\mathbf{C}^{(8)}$
52−59	**U B B U**
60−63	**B B B B**

8 Conclusion

In this paper, we evaluated the security against integral attack on the block cipher LBC. First, we searched for integral characteristics with the bit-based division property using MILP and found the 18-round integral characteristic using the 60-th order differential. We also showed the 16-round integral characteristic using the 48-th order differential, which can efficiently be exploited for an integral attack on the full-round LBC with $2^{50.8}$ blocks of chosen plaintext and $2^{51.4}$ times of encryption operation. Furthermore, we tried to improve the round function of LBC. Then, we confirmed that the improved LBC strengthens the original one because the length of the integral characteristic is reduced to 8 rounds.

In future work, we will analyze the key scheduling part and consider the attack that exploits its property.

References

1. N. Kapalova, K. Algazy, and A. Haumen, "Development of A New Lightweight Encryption Algorithm," Eastern-European Journal of Enterprise Technologies ISSN 1729-3774, pp.6–19, 2023.
2. L. Knudsen and D. Wagner, "Integral Cryptanalysis," FSE2002, LNCS2365, pp.112–127, Springer, Heidelberg, 2002.
3. Y. Todo, "Structural Evaluation by Generalized Integral Property," EUROCRYPT2015, Springer, LNCS9056, pp.287–314, 2015.
4. Y. Todo and M. Morii, "Bit-Based Division Property and Application to SIMON Family," FSE2016, LNCS9783, pp.375–377, Springer, 2016.
5. Z. Xiang, W. Zhang, Z. Bao, and D. Lin, "Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers," ASIACRYPT2016, LNCS10031, pp.648–678, Springer, 2016.
6. SageMathCell, `https://sagecell.sagemath.org/`.
7. Gurobi Optimization, LCC, "Gurobi Optimizer Reference Manual," `https://www.gurobi.com`, 2023.
8. Y. Sasaki and L. Wang, "Meet-in-the-Middle Technique for Integral Attacks against Feistel Ciphers," SAC2012, LNCS7707, pp.234–251, Springer, 2013.
9. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wanger, and D. Whiting, "Improved Cryptanalysis of Rijndael," FSE2000, LNCS1978, pp.213–230, Springer, 2001.

ARP Spoofing Mitigation for the E2 Interface in Open RAN: An xApp Approach

Jihye Kim[1], Jaehyoung Park[2], and Jong-Hyouk Lee[2]

[1] Department of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea
jihye@pel.sejong.ac.kr
[2] Department of Computer and Information Security & Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea
jaehyoung@pel.sejong.ac.kr, jonghyouk@sejong.ac.kr

Abstract. In the 5G-Advanced and 6G era, the Open Radio Access Network (Open RAN) will support hardware and software interoperability between different vendors. The O-RAN Alliance, which leads the Open RAN initiative, has released standards addressing potential threats and mitigation requirements to the framework. While the standards specify Man-in-the-Middle (MitM) attacks on the E2 interface and recommend using IPsec, a Layer 3 security protocol, they do not cover defenses against Layer 2 attacks, such as Address Resolution Protocol (ARP) spoofing. This paper proposes an xApp with preventive and reactive mechanisms to protect the E2 interface from ARP spoofing in Open RAN. The xApp proactively prevents attacks through a static ARP table and continuously monitors the E2 interface to detect and react to the ARP spoofing, offering better network resilience than existing methods.

Keywords: Open RAN · E2 interface · xApp · MitM.

1 Introduction

In anticipation of the 5G-Advanced and 6G era, Open Radio Access Network (Open RAN) has emerged as the next-generation RAN technology to rapidly process the massive amounts of data generated by an increasing number of mobile devices and applications in the existing RAN. Open RAN incorporates new components and interfaces into the NG-RAN as defined by the 3rd Generation Partnership Project (3GPP). Open RAN, which aims to enhance the flexibility and interoperability of networks, emphasizes the openness and intelligence of RAN. Open RAN disaggregates functions of the base station to improve data processing in the traditional RAN, providing a distributed deployment. And software-based components are interconnected through open interfaces. Various equipment suppliers enable interoperability between different vendors by flexibly deploying virtualized software. This allows multiple enterprises and individual users to leverage the RAN ecosystem. Furthermore, Artificial Intelligence (AI) models are employed to maintain the RAN's optimal state in Open RAN. AI models are integrated into the RAN Intelligent Controller (RIC) in Open RAN to automate network operations and provide efficient performance [1].

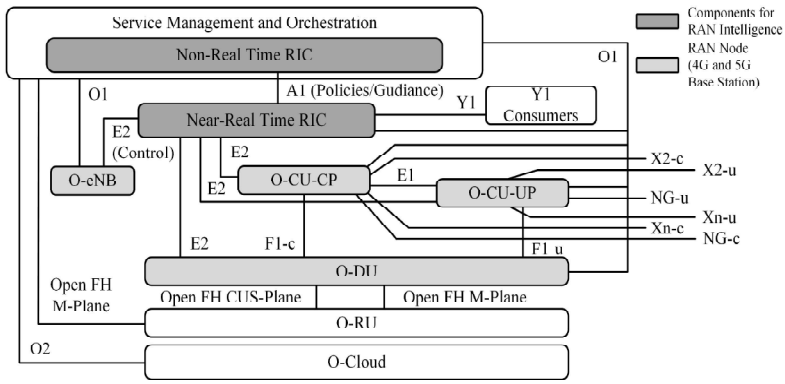


Fig. 1. Open RAN architecture by the O-RAN Alliance

The O-RAN Alliance takes a leading role in addressing security standards for the advancement of Open RAN. Figure 1 illustrates the Open RAN architecture provided by the O-RAN Alliance. Non-Real-Time (Non-RT) RICs and Near-Real-Time (Near-RT) RICs are key to the intelligence of the RAN. The Non-RT RIC utilizes advanced AI models to predict and analyze network traffic and Quality of Service (QoS). The Near-RT RIC leverages the AI models deployed by the Non-RT RIC to monitor and dynamically control the state of the RAN. The A1 interface connecting two RICs and the E2 interface connecting the Near-RT RIC with RAN nodes (i.e., gNBs) contribute to the automation and optimization of RAN, enhancing its intelligence [2].

While commercialization of Open RAN is still in its early stages, recent research and documents have also focused on security aspects to complement Open RAN. The O-RAN Alliance defines potential security threats and requirements for each component and interface from the Open RAN framework [3]. Recently, new releases of Open RAN have been published approximately yearly, aiming to reinforce the components necessary to prepare for Open RAN deployment. We have referenced the standards published by the O-RAN Alliance to identify threats and requirements for the E2 interface, which is utilized for the intelligence of RAN. The O-RAN Alliance identifies Man-in-the-Middle (MitM) attacks as potential threats to the E2 interface. Furthermore, to enhance the security of the E2 interface, the O-RAN Alliance proposes utilizing the IPsec, the Layer 3 security protocol. MitM attacks include not only Layer 3 attacks like IP spoofing and ICMP redirect but also Layer 2 attacks like ARP spoofing. However, since the published standards only address Layer 3 security, this paper proposes functionalities to protect the E2 interface from Layer 2 attacks. The contribution of this paper is threefold:

- By proposing protective functionalities for Layer 2 of the E2 interface, we emphasize the necessity of Layer 2 security for the E2 interface.
- We provide advanced functionalities beyond existing research by proposing both preventive and reactive measures to prevent ARP spoofing.

– Our research establishes a foundation for security-related studies and standardization efforts aimed at advancing the next-generation RANs.

The structure of this paper is as follows. Section 2 analyzes existing research on security for the E2 interface of Open RAN, specifically existing mechanisms aimed at preventing and mitigating Layer 2 attacks. Section 3 proposes an ARP spoofing mitigation xApp to protect the E2 interface. Section 4 evaluates the proposed xApp for ARP spoofing mitigation by comparing it with existing research. Section 5 discusses the result and limitation of the proposed xApp, and finally, Section 6 concludes this paper.

2 Related Work

2.1 Security Threat and Requirements on the E2 Interface

The E2 interface connects the Near-RT RIC with RAN nodes and serves as a crucial interface in Open RAN for RAN intelligence and optimization. The Near-RT RIC collects the status information from RAN nodes via the E2 interface to assess their optimization status, predicts future status, and controls the RAN in Near-RT (e.g., 0.01s to 1s). If attacks occur on the E2 interface, it can lead to system errors such as RAN being paused or performance degradation. This emphasizes the critical importance of securing the E2 interface in maintaining the operational integrity and performance of the RAN. Therefore, the E2 interface security must be considered mandatory.

The O-RAN Alliance WG11 provides security standards for components and interfaces. According to the O-RAN security threat modeling and risk assessment published by WG11, Open RAN systems can be vulnerable to attackers who may exploit the E2 interface to compromise the system. Furthermore, data transmitted through the E2 interface can be maliciously manipulated by malicious xApps. In the E2 interface, if an untrusted Near-RT RIC or E2 node communicates with other components through the E2 interface, it is vulnerable to mutual authentication issues. Furthermore, attackers can access the E2 interface via MitM attacks, allowing them to read, manipulate, and insert messages. As a result, the Near-RT RIC or the E2 nodes can receive malicious messages [3]. Table 1 lists security threats related to the E2 interface referencing the threat IDs defined by the O-RAN Alliance.

Table 1. Security threats on the E2 interface by the O-RAN Alliance

ID	Title
T-O-RAN-05	Attacker penetrates and compromises the O-RAN system through the O-RAN's Fronthaul, O1, O2, A1, and E2
T-xApp-01	Attacker exploits xApps vulnerabilities and misconfiguration
T-E2-01	Untrusted Near-RT-RIC and E2 Nodes
T-E2-02	Malicious actor monitors messaging across the E2 interface

According to the security requirements and controls specification published by WG11, the E2 interface must support confidentiality, integrity, replay protec-

tion, and data origin authentication. Furthermore, for the IP layer protection of the E2 interface, it should support IPsec [3]. Table 2 lists security requirements related to the E2 interface referencing the IDs defined by the O-RAN Alliance.

Table 2. Security requirements and controls on the E2 interface by the O-RAN Alliance

ID	Description
REQ-SEC-E2-1	E2 interface shall support confidentiality, integrity, replay protection and data origin authentication.
SEC-CTL-E2	For the security protection at the IP layer on E2 interface, IPsec shall be supported as specified in O-RAN Security Protocols Specifications.

According to the standards of the O-RAN Alliance, the E2 interface is susceptible to MitM attacks, where attackers can intercept messages transmitted through the E2 interface by performing MitM attacks. This exposes the E2 interface to secondary threats such as message manipulation. To protect the E2 interface from such threats, it should support functionalities that ensure confidentiality and integrity. Additionally, support for IPsec is necessary to safeguard communications at the IP layer. However, the IPsec, the Layer 3 security protocol, provided in the standards by the O-RAN Alliance can only address Layer 3 attacks such as IP spoofing and ICMP redirect. Therefore, there is a need for functionalities to prevent Layer 2 attacks, such as ARP spoofing.

2.2 Literature Review of the E2 Interface Security

This section presents existing research to identify and mitigate security threats to the E2 interface above the Open RAN environment.

Hung et al. [4] provide insights into threats discovered from O-RAN A-Release to H-Release, the open-source of Open RAN. The author emphasizes the necessity of detecting malicious xApps. In the open-source environment discussed, the lack of specific access control privilege for xApps allows malicious attackers to potentially disrupt the RAN by attacking legitimate the xApps and the E2 nodes. The author demonstrates privilege management threats, E2 subscription issues, and Denial of Service (DoS) attacks through experiments.

Djuitcheu et al. [5] analyze threats related to complex services and functionalities on the E2 interface. The author analyzes threats posed by agents such as malicious xApps, compromised xApps, malicious E2 nodes, and terminations of interfaces. The study focuses on threats related to the E2 interface, including DoS attacks, MitM attacks, protocol exploitation, replay attacks, unauthorized access, manipulation and tampering of E2 data, and malware deployment. In response to these threats, mitigation strategies proposed theoretically include applying IPsec, leveraging xApps and Machine Learning (ML), and utilizing a new security design incorporating zero trust architecture.

Radhakrishnan et al. [6] provide use cases involving communication between the Near-RT RIC and the E2 interface, as well as controlling the RAN through multi-xApps. Following this, the author explains the security of the E2 interface

for the Near-RT RIC. The author provides scenarios of resource exhaustion and signaling storm DoS attacks on the E2 interface and presents the results of designing and developing xApp-based latency monitoring to detect and mitigate these attacks.

Table 3 compares security threats or potential attacks on the E2 interface across existing studies and outlines mitigation strategies. We denoted with "-", which papers do not provide mitigation strategies.

Table 3. Comparison of related works for the E2 interface security

Ref.	Security threats and attacks	Mitigation strategies
[4]	Privilege management, E2 subscription issue, DoS attack	-
[5]	DoS attack, MitM attack, Protocol exploitation, Replay attack, Unauthorized access, Manipulation and tampering of E2 data, Malware deployment	IPsec, xApp, Machine Learning, Zero trust architecture
[6]	DoS attack	Latency monitoring xApp

According to existing research, there is a lack of research solely focused on protecting the E2 interface and developing security mechanisms specifically for the E2 interface. Also, the protection of the E2 interface can be achieved by designing xApps effectively. Additionally, applying security protocols like IPsec and leveraging AI can help mitigate attacks effectively.

2.3 Literature Review on Mitigation Strategies of Layer 2 Attacks

Continuous research is essential to safeguard the E2 interface in Open RAN, against Layer 2 attacks like ARP spoofing that are not covered in current Open RAN literature. This paper introduces existing research focused on preventing and mitigating traditional Layer 2 attacks in similar network environments.

Static ARP Cache. ARP spoofing is when an attacker alters a node's MAC address to impersonate it, tricking others into thinking the attacker's device is the real node. This manipulation affects ARP tables of other Local Area Network (LAN) nodes, displaying the attacker's MAC as the legitimate one. Setting static ARP mappings on each LAN host can prevent unauthorized network access [7].

Duplicated MAC Address. ARP spoofing involves an attacker changing a node's MAC address to their own, causing the ARP table to associate the attacker's MAC address with the victim node. Therefore, duplicate MAC addresses in the ARP table can be detected as an ARP spoofing attempt [8,9].

Delete the Attacker's ARP Cache. ARP spoofing involves an attacker attacking the interface between two legitimate nodes on a network. Therefore, if the attacker attempts to attack the interface, their cache in the ARP table can be cleared to remove their presence and block them [10,11].

MAC Address Blocking Rule. ARP spoofing involves an attacker attacking the connection between nodes, which can lead to secondary threats such as packet manipulation. Therefore, when detecting the attacker, setting a rule to block the attacker's MAC address prevents them from attempting further attacks [12].

Restore the Original ARP Table. ARP spoofing is when an attacker manipulates a node's MAC address. After detecting and blocking the attack, it's crucial to restore the affected node's connection by updating the ARP table with its original MAC address using functions [9].

3 Proposed ARP Spoofing Mitigation xApp

3.1 System Architecture

This section describes the ARP spoofing mitigation xApp proposed to protect the E2 interface from ARP spoofing, which is a potential MitM attack specific to Layer 2. A typical xApp is an application utilized by the Near-RT RIC to provide specific services. The Near-RT RIC utilizes multi-xApps to provide various services. Additionally, users can design, develop, and utilize xApps to provide the services they need. The proposed xApp includes both preventive and post-event response functions for ARP spoofing prevention on the E2 interface, shown by the gray part in Fig. 2. The proposed xApp can be utilized alongside various other xApps within Near-RT RIC, serving as an application and service that contributes to the overall functionality and optimization of the network. The notations within this xApp are described in Table 4.

The preventive function utilized in the proposed xApp manages the ARP table, while the post-event response function detects and responds to ARP spoofing. ARP spoofing can be prevented in advance by statically setting the MAC addresses of the nodes connected through the E2 interface. However, despite the

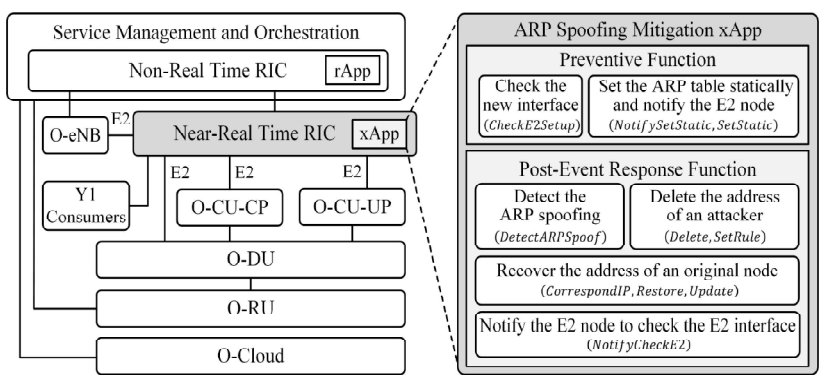


Fig. 2. System architecture containing the proposed xApp for the E2 interface security

implementation of preventive functions, ARP spoofing can still occur when performing tasks such as the RAN status measurement through the E2 interface. To address this attack, the proposed xApp provides the post-event response function that deletes the attacker's traces from the Near-RT RIC's ARP table and sets rules to block the attacker.

Table 4. Notation of the proposed xApp

Notation	Description
$MAC_{NearRTRIC}$	MAC address of a Near-RT RIC
MAC_{E2node}	MAC address of an original E2 node
$MAC_{attacker}$	MAC address suspected by an ARP spoofing attacker
$IP_{NearRTRIC}$	IP address of the Near-RT RIC
IP_{E2node}	IP address of the original E2 node
$IP_{attacker}$	IP address suspected by the ARP spoofing attacker
ARP_{table}	ARP table of the Near-RT RIC
$CheckE2Setup(x)$	Check the setting of the new E2 interface with x
$NotifySetStatic(x, y)$	Notify the E2 node to set the ARP cache statically corresponding to x and y
$DetectARPSpoof(x)$	Detect the ARP spoofing on the E2 interface with x
$Delete(x, y)$	Delete y from x
$SetStatic(x, y)$	Set the ARP cache statically corresponding to x and y
$SetRule(x, y)$	Set a rule to block x and y
$CorrespondIP(x)$	IP address corresponding to x
$Restore(x)$	Restore the MAC address of the original node with x
$Update(x, y, z)$	Update y to z from x
$NotifyCheckE2(x)$	Notify the E2 node to check the E2 interface connectivity and verify that the MAC address of the Near-RT RIC is x

3.2 Prevention Function of the ARP Spoofing Mitigation xApp

The proposed xApp statically manages the ARP table of Near-RT RIC to prevent ARP spoofing on the E2 interface. ARP spoofing targets interfaces, so it is possible only when the E2 interface exists between the Near-RT RIC and the E2 node. Therefore, the principle of this function is for the Near-RT RIC to statically set the MAC address of the newly connected E2 node when setting the E2 interface. Furthermore, ARP spoofing involves an attacker falsifying the MAC address of the node. By statically managing the ARP table, the attacker is prevented from manipulating MAC addresses.

The proposed xApp can verify the setting of E2 interfaces by monitoring packets related to the E2 interface. The setting of E2 interfaces refers to the E2 setup procedure provided by the O-RAN Alliance. As depicted in Fig. 3, the E2 interface is established when the Near-RT RIC receives an E2 SETUP REQUEST from the E2 node and subsequently sends an E2 SETUP RESPONSE to the E2 node [3]. Using this procedure, you can monitor the setting of the E2 interface and statically configure the MAC address of the E2 node as soon as it is established.



Fig. 3. E2 setup procedure by the O-RAN Alliance [3]

Algorithm 1 is the first function of the proposed xApp, which aims to prevent ARP spoofing proactively by statically managing the ARP table of the Near-RT RIC. Lines 2 to 3 monitor packets transmitted over the E2 interface, checking the setting of the E2 interface by observing the process where the Near-RT RIC receives the E2 SETUP REQUEST from an E2 node and sends the E2 SETUP RESPONSE to that node. If confirmed, it saves the MAC address of the E2 node connecting to the new E2 interface as MAC_{E2node} and stores the corresponding IP address as IP_{E2node}. Lines 4 to 7 are executed if IP_{E2node} and MAC_{E2node} is found in ARP_{table}. In this case, we can see that the E2 node connected to the new E2 interface is successfully connected, so we notify the E2 node to set the ARP cache for $IP_{NearRTRIC}$ and $MAC_{NearRTRIC}$ statically in ARP table of it, by transmitting those addresses. Later, statically set the ARP cache corresponding to IP_{E2node} and MAC_{E2node} in ARP_{table}. This prevents ARP spoofing by making it difficult to manipulate the MAC address of the E2 node in the ARP table.

Algorithm 1 Preventive Function: ARP table management

```
1: begin
2: for each packet in packets do
3:     IP_E2node, MAC_E2node ← CheckE2Setup(packet)
4:     if IP_E2node, MAC_E2node in ARP_table then
5:         NotifySetStatic(IP_NearRTRIC, MAC_NearRTRIC)
6:         SetStatic(IP_E2node, MAC_E2node)
7:     end if
8: end for
9: end
```

3.3 Post-Event Function of the ARP Spoofing Mitigation xApp

In the proposed xApp, ARP spoofing prevention for the E2 interface is achieved by continuously detecting and responding to ARP spoofing after the attack event. This function operates after performing the preemptive prevention measures for the E2 interface. ARP spoofing involves an attacker falsifying the MAC address of an E2 node connected to the Near-RT RIC via the E2 interface. Consequently, the MAC addresses of nodes stored in the Near-RT RIC's ARP table are also altered to reflect the attacker's MAC address. Therefore, the principle of the post-event response function is to detect ARP spoofing attempts on the E2 interface, block the attacker, and restore the ARP table with the original

E2 node's MAC address. Furthermore, it prevents potential reattacks during the restoration process by immediately blocking the attacker upon detecting the attack and restoring the original address.

The proposed xApp enables ARP spoofing detection through packets received by the Near-RT RIC. An ARP REPLY follows the format shown in Fig. 4, and an attacker exploits this to attempt ARP spoofing. The attacker continuously sends the ARP REPLY to the Near-RT RIC, where the target protocol address is the IP address of a legitimate E2 node connected to the E2 interface, and the target hardware address is the MAC address of the attacker. Furthermore, the attacker continuously sends the ARP REPLY to the E2 node, where the target protocol address is the IP address of the Near-RT RIC, and the target hardware address is the MAC address of the attacker itself. Therefore, the Near-RT RIC and the E2 node each received the ARP REPLY will think that the attacker is each other's counterpart node. By continuously monitoring the transmission of numerous ARP REPLY, it is possible to detect ARP spoofing. The node sending these packets can be identified as the attacker attempting ARP spoofing.

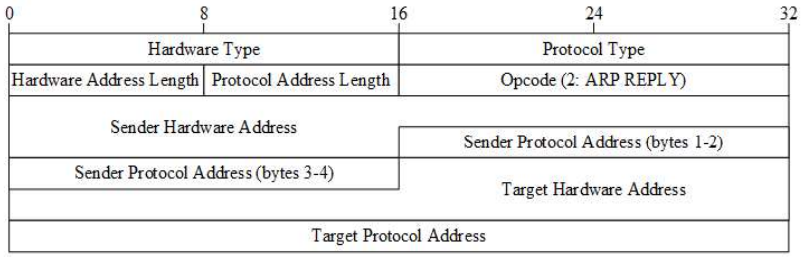


Fig. 4. ARP REPLY packet format

Algorithm 2 is the second function of the proposed xApp, which detects and responds to ARP spoofing on the E2 interface. Lines 2 to 3 monitor packets transmitted over the E2 interface and, upon detecting an attack through the Near-RT RIC continuously receiving an ARP REPLY, consider the node responsible for the attack as the attacker. It then stores the IP address of the node suspected to be the attacker as $IP_{attacker}$ and its MAC address as $MAC_{attacker}$. Lines 4 to 11 are executed if $IP_{attacker}$ and $MAC_{attacker}$ is found in ARP_{table}. In this case, two IP addresses associated with $MAC_{attacker}$ is checked. The IP address is found to be different from the attacker's IP address, $IP_{attacker}$. A different IP address corresponds to the original E2 node previously connected to the E2 interface, and this address is saved as IP_{E2node}. Afterward, the ARP cache corresponding to $IP_{attacker}$ and $MAC_{attacker}$ in the ARP_{table} is deleted. The process involves verifying and removing duplicate MAC addresses. And setting network rules to block the $IP_{attacker}$ and $MAC_{attacker}$ which is a suspected attacker. Since the attacker has been blocked, retrieve the original MAC address corresponding to the IP_{E2node} of the original E2 node and store it in MAC_{E2node}. Finally, update ARP_{table} by replacing $MAC_{attacker}$ with MAC_{E2node} to restore it to its origi-

nal state. This enables detection and response against ARP spoofing on the E2 interface. The Near-RT RIC notifies the E2 node to check the E2 interface ARP spoofing is attempted and set up a new E2 interface connection, by transmitting $IP_{NearRTRIC}$ and $MAC_{NearRTRIC}$.

Algorithm 2 Post-Event Response Function: ARP spoofing detection and response

```
begin
for each packet in packets do
    IP_attacker, MAC_attacker ← DetectARPSpoof(packet)
    if IP_attacker, MAC_attacker in ARP_table then
        IP_E2node ← CorrespondIP(IP_attacker, MAC_attacker)
        Delete(ARP_table, IP_attacker, MAC_attacker)
        SetRule(IP_attakcker, MAC_attacker)
        MAC_E2node ← Restore(IP_E2node)
        Update(ARP_table, MAC_attacker, MAC_E2node)
        NotifyCheckE2(IP_NearRTRIC, MAC_NearRTRIC)
    end if
end for
end
```

4 Comparison

The proposed xApp provides functions to prevent and respond to Layer 2 attacks such as ARP spoofing on the E2 interface. There are no mechanisms or solutions currently available to prevent attacks in Open RAN. Therefore, by comparing the existing ARP spoofing prevention mechanisms introduced in Section 2 with the proposed xApp in Table 5, we demonstrate the superiority of our system.

The existing research involves setting a static ARP table and deleting the attacker's cache. Furthermore, it sets up rules to block the attacker and restores the original ARP table to detect and respond to ARP spoofing. The proposed xApp adheres to all five functionalities in existing research. To prevent intrusions by attackers proactively, the xApp manages the ARP table by statically setting the MAC address of the E2 node. However, despite these proactive measures, considering the possibility of advanced and sophisticated attacks, the xApp also monitors duplicate MAC addresses in the ARP table and removes the cache of attackers with those addresses. Additionally, the xApp sets up rules to block the identified attackers, making it difficult for them to attempt further attacks. It then verifies the original MAC address of the E2 node and updates the ARP table to ensure uninterrupted connectivity.

5 Discussion

Since Open RAN allows implementation using publicly available software, including interfaces such as E2 and A1, there is a possibility of attacks like MitM attacks [13]. If a Layer 2 attack like ARP spoofing occurs on the E2 interface,

Table 5. Comparison between the proposal xApp and the existing defense mechanisms

Defense mechanism	Static ARP table	Duplicated MAC address	Delete the attacker's ARP cache	Attacker blocking rule	Restore the original ARP table
Galal et al. [8]	X	O	X	X	X
Prasad et al. [9]	X	O	X	X	O
Hou et al. [10]	O	X	O	X	X
Hijazi et al. [11]	O	X	O	X	O
Ortega et al. [12]	X	X	X	O	X
Proposal xApp	O	O	O	O	O

the attacker can eavesdrop on the status information of RAN nodes. By manipulating this information, they can potentially induce pause states in the RAN, thereby delaying its operations. As a result, if the Near-RT RIC fails to monitor the RAN due to such attacks, it can lead to errors in RAN management.

Therefore, the proposed xApp provides preventive and post-event response functions within the Near-RT RIC to prevent and mitigate the ARP spoofing on the E2 interface. Due to the nature of ARP spoofing, where attackers falsify MAC addresses of legitimate nodes to impersonate them, utilizing preventive measures ensures an environment where attackers cannot manipulate MAC addresses. However, this xApp sets the MAC address of an E2 node statically as soon as it connects to the E2 interface, without distinguishing whether the node is legitimate or an attacker. Therefore, it is necessary to verify whether the node is trusted or not, and this verification process should be expanded in future research to focus on developing secure security functions and mechanisms.

Existing studies suggest static ARP table configuration to prevent ARP spoofing. However, concerns remain about ARP spoofing attempts due to system failures, mismanagement, or sophisticated attacks. To prepare for such worst-case scenarios, continuous monitoring of the E2 interface is essential even after managing the ARP table statically.

6 Conclusion

Open RAN has emerged as a new paradigm for next-generation RAN, particularly towards 5G-Advanced and 6G. It is being researched to overcome the limitations of traditional RAN, with ongoing studies focused on security to facilitate its future commercialization. This paper investigates the protection of the E2 interface, which is crucial for the intelligence of RAN. According to the O-RAN Alliance, MitM attacks are possible on the E2 interface, necessitating the implementation of IPsec. However, MitM attacks include not only Layer 3 attacks but also Layer 2 attacks such as ARP spoofing. Thus, despite IPsec implementation, bypassing attacks can still occur. To address this, we have presented the proposed xApp, which protects the E2 interface from Layer 2 attacks, specifically ARP spoofing. The xApp prevents attacks proactively through static ARP table management and continuously monitors the E2 interface to detect

attacks, block attackers, and restore the ARP table, providing better network resilience compared to existing research. For future research, we aim to enhance the preventive function of the xApp to develop a superior ARP spoofing mitigation mechanism. Furthermore, through the xApp, we emphasize the necessity of Layer 2 security for the E2 interface in Open RAN.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00796, Research on Foundational Technologies for 6G Autonomous Security-by-Design to Guarantee Constant Quality of Security).

References

1. Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T.: Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys & Tutorials **25**(2), 1376–1411 (2023)
2. Tiberti, W., Fina, E., Marotta, A., and Cassioli, D.: Impact of Man-in-the-Middle Attacks to the O-RAN Inter-Controllers Interface. 2022 IEEE Future Networks World Forum (FNWF), pp. 367–372 (2022)
3. O-RAN ALLIANCE, https://specifications.o-ran.org/specifications, last accessed 2024/06/02
4. Hung, C.-F., Chen. Y.-R., Tseng, C.-H., and Cheng, S.-M.: Security Threats to xApps Access Control and E2 Interface in O-RAN. IEEE Open Journal of the Communications Society **5**(1), 1197–1203 (2024)
5. Djuitcheu, H., et al.: Exploring the Implications and Methodologies of Securing the E2 Interface. Authorea Preprints (2024)
6. Radhakrishnan, V.: Detection of Denial of Service Attacks on the Open Radio Access Network Intelligent Controller through the E2 Interface. PhD Thesis, Virginia Tech (2023)
7. Meghana, J., Subashri, T., and Vimal, K.: A survey on ARP cache poisoning and techniques for detection and mitigation. In: 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN). IEEE, pp. 1–6 (2017)
8. Galal, A., Ghalwash, A., Nasr, M.: A new approach for detecting and mitigating address resolution protocol (arp) poisoning. International Journal of Advanced Computer Science and Applications **13**(6), 377–382 (2022)
9. Prasad, A. and Chandra, S.: Defending arp spoofing-based mitm attack using machine learning and device profiling. In: 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 978-982 (2022)
10. Hou, X., Jiang, Z., and Tian, X.: The detection and prevention for ARP Spoofing based on Snort. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) (2010)
11. Hijazi, S. and Obaidat, M.: Address resolution protocol spoofing attacks and security approaches: A survey. Security and Privacy **2**(1) (2019)
12. Ortega, A., Marcos, X., Chiang, L., and Abad, C.: Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt. In: 2009 Latin American Network Operations and Management Symposium. IEEE, pp. 1–9 (2009)
13. Kim, J., Park, J., and Lee. J.-H.: Simulation of an ARP Spoofing Attack on the E2 Interface in Open RAN. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2024) (2024)

Investigating the Transferability of Evasion Attacks in Network Intrusion Detection Systems Considering Domain-Specific Constraints

Mariama Mbow[1], Rodrigo Roman[2], Ayan Seal[3], Kevin I-Kai Wang[4], Sraban Kumar Mohanty[3], and Kouichi Sakurai[1]

[1] Department of Informatics, Kyushu University, Japan
[2] Network, Information and Computer Security (NICS) Lab, University of Malaga, Spain
[3] PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482005, India
[4] Department of Electrical, Computer and Software Engineering, The University of Auckland, Auckland 1010, New Zealand

Abstract. Recently, adversarial attacks against machine learning-based network intrusion detection systems (NIDS) have gained significant attention in cybersecurity. This study investigates the transferability of these adversarial attacks in the context of NIDS. In addition, most existing studies in adversarial learning against NIDS adopted attack strategies designed originally for image classification without considering network traffic characteristics. However, this is impractical and will fail in the real world as network traffic features are constrained, and ignoring the functional behavior of the network traffic features leads to invalid network traffic flow or produces adversarial samples that do not retain their original functionality (malicious or benign). To address these issues, we propose a constrained momentum iterative fast gradient sign method (C-MIFGSM) to generate adversarial network flows that can successfully evade an ML-based IDS through transfer-based attacks while preserving the functional behavior of the network traffic. Our approach was validated using several target NIDS models built with the NSLKDD benchmark dataset. Experimental results demonstrate that even without knowledge of the target model and under feature constraints, it is possible to generate adversarial network traffic flows that achieve a high evasion attack success rate against NIDS built with deep learning and classical ML models. For example, the attack degraded the detection rate of the DoS traffic drops from 92.35% to 20.27% for the MLP model with an evasion increase rate of 78.04%.

Keywords: Cybersecurity · Network intrusion detection systems · Adversarial machine learning · Evasion attacks.

1 Introduction

Network intrusion detection systems (NIDS) are the standard method used in cybersecurity to detect malicious activities on network traffic [1]. There are two

main methods for detecting intrusions: signature-based and anomaly-based. The signature-based method detects attacks using their signatures. This technique works well for spotting known threats but fails to detect novel, unknown attacks like zero-day exploits. In contrast, anomaly-based NIDS monitors typical network traffic behavior and flags deviations from this norm as potential attacks. Anomaly-based methods have been thoroughly researched due to their ability to detect novel attacks without pre-existing signatures. Additionally, anomaly-based detection enhanced by machine learning models has demonstrated promising outcomes, improving NIDS with a low false alarm rate, high detection rate, and robust capabilities to manage large volumes of network traffic [2]. Machine learning (ML)-based NIDS is the main topic of this article.

Despite their remarkable performance in NIDS, prior research has demonstrated that ML models, and deep learning (DL) models in particular, are vulnerable to adversarial examples [3, 4]. These examples are small, carefully crafted perturbations to the inputs designed to deceive the model and degrade its classification performance. Although this phenomenon was initially identified in image recognition [5], it has also been proven in various other ML applications. Furthermore, previous work in computer vision has demonstrated empirical findings about the transferability of adversarial examples between neural networks [6]. This means that adversarial examples capable of deceiving one model can likely deceive another, even if the models have different architectures or training datasets. Consequently, an attacker can train a surrogate model to generate adversarial examples and transfer these examples to the target model with minimal knowledge of the target. When applied to cybersecurity tools, such techniques enable attackers to bypass detection systems, posing a significant threat to security, particularly for NIDS. This vulnerability raises serious concerns about the reliability and effectiveness of ML in security-critical applications like NIDS [7].

1.1 Challenging issues

The evasion attacks against machine learning are seeing great advancement, and several studies have explored the vulnerability of DL-based NIDS to adversarial attacks. Most of the existing research has predominantly focused on crafting adversarial examples under white-box scenarios where the attacker has full knowledge of the target model [9, 10]. Recent studies are focusing on more realistic threat models investigating the gray-box or black-box scenario [12]. On the other hand, little attention has been paid to the transferability of these attacks in the context of NIDS. In addition, the practicality of the proposed attack methods is overlooked. Many studies have employed attack strategies originally designed for image classification, assuming they can add perturbation to any features, ignoring the unique characteristics of network traffic. Network traffic structures differ significantly from image pixels, and modifying certain features alters the functional behavior of the traffic flow [11]. This however invalidates the network traffic features. Furthermore, a significant number of previous studies assume that the adversary has complete knowledge of the NIDS models (white-box at-

tack). However, this assumption is impractical, as it is unlikely for an adversary to have knowledge of the detection model and its internal configuration.

1.2 Contributions

To address these issues, this work investigates a more practical evasion attack in a gray box setting, leveraging the transferability properties to perform adversarial attacks while preserving the functional behavior of network traffic flow. We propose a constrained momentum iterative fast gradient sign method (C-MIFGSM)-based approach that extends the momentum iterative fast gradient method (MIFGSM) [13] while complying with domain constraints. Our approach is evaluated on various NIDS models using one of the most used network intrusion datasets, the NSLKDD, to assess the robustness of various ML and DL models. Our contributions can be summarized as follows:

- We propose a C-MIFGSM, an algorithm that extends the MIFGSM to generate adversarial network traffic flows based on the L_∞norm. The proposed method generates adversarial examples that preserve the functional behavior of the network traffic flow.
- We investigate the transferability attack in the context of NIDS to perform adversarial attacks that can evade detection without prior knowledge of the target model.
- We examine the transferability of adversarial samples from deep neural networks (DNN) to various classical ML algorithms, such as random forest (RF), decision tree (DT), XGBoost, multilayer perceptron (MLP), logistic regression(LR) and two state-of-the-art DL models, using NSLKDD dataset. We find that samples transfer well across models trained with different ML techniques.

The remainder of this paper is organized as follows: Section 2 provides the background of this work. In section 3, we propose our approach. Section 4 presents the experiments and evaluations of the proposed approach. Finally section 5 presents the conclusion of the work.

2 Background

2.1 Related Work

Previous studies have demonstrated that DNNs are vulnerable to evasion attacks. Szegedy et al. [5] and Biggio et al. [15] showed the vulnerability of image classifiers to adversarial attacks in a white box setting. Since then, there have been many proposed works in crafting adversarial examples by adding perturbations to the image pixel [6–8]. However, it has been demonstrated that these attacks exist beyond images. Carlini et al. [9] showed how an adversary can fool an automatic speech recognition system. Li et al. [11] illustrated how to generate adversarial text attacks that preserve the utility of benign text. Grosses et al.

[10] studied adversarial examples for malware detection while maintaining the malware's functionality. However, in each of the above-stated domains, adversarial examples have been crafted differently. Such a situation illustrates that the feasibility of generating successful adversarial attacks is domain-specific. Therefore, we need to understand its feasibility in the context of network traffic to successfully craft adversarial examples in NIDS.

There have been studies on adversarial attacks against NIDS [9, 10]. However, despite the numerous works addressing these attacks against NIDS, little attention was paid to the challenges of the generated adversarial examples. Consequently, a repeated criticism made by recent surveys [10, 20, 21] is the impracticality of most of the previous works in a real-world NIDS scenario. Most prior efforts at crafting adversarial examples on NIDS ignore the preservation of the functional behavior of the network traffic [11] or assume a white-box attack.

Recent studies have focused on maintaining the functional behavior of generated adversarial network traffic flows in NSLKDD [11, 14]. Despite limiting their attacks to non-functional features, these studies did not impose restrictions on the generated perturbations within the valid range. This oversight can result in unreasonable values, potentially invalidating the adversarial network traffic flows. Additionally, they did not consider the type of features. Furthermore, they did not investigate the transferability of these adversarial samples.

2.2 Challenges in generating adversarial examples in NIDS

In network traffic, successful adversarial attacks must not only introduce small perturbations but also adhere to practical constraints. Unlike unstructured data like image pixels, network traffic data are structured and constrained by several factors that impact the feasibility of attacks on NIDS. These constraints include:

- Feature type: Network data can include various feature types, such as continuous, categorical, and binary. Adversarial examples must respect these distinctions to avoid violating data integrity. For instance, perturbations should not convert binary features into non-binary values or alter the representation of categorical data.
- Out-of-range values: Perturbations must not exceed the maximum or fall below the minimum valid values for each feature. Adversarial examples should stay within the permissible ranges to avoid disrupting the functionality of the network traffic.
- Functional features: Each category of attack has specific functional features that represent the fundamental characteristics or behaviors of that particular type of attack. Therefore, when generating adversarial examples, it is essential to ensure that perturbations are added only to non-functional features. This approach ensures that the malicious adversarial traffic preserves its functionality while being misclassified [11].

These characteristics imply a different method than adversarial attacks developed for images should be investigated to satisfy the domain-specific constraints by preserving the functional behavior of the generated adversarial examples and achieving valid adversarial network traffic.

3 Approach

This section provides a detailed explanation of our proposed method. The objective is to generate adversarial examples that can effectively deceive the NIDS model.

3.1 Problem definition

Given a network traffic flow represented as an n-dimensional feature vector $x \in X$ extracted by feature extraction methods, the NIDS model classifier $f : X \to Y$ takes x as input and predicts the label $y \in Y$. Our goal is to mislead the NIDS model into misclassifying malicious traffic. Specifically, we seek an example x^* in the vicinity of x that is misclassified by the detector.

3.2 Methodology

One strategy to attack the target without direct access to the victim's internal configuration is to train a local model to approximate a decision boundary of the target. Adversarial samples are then crafted using this surrogate model, and the transferability property is exploited to transfer these samples to the target model, to which the adversary has no direct access.

In our scenario, we employ a gray-box attack, assuming that the internal structure and parameters of the detection system are unknown to attackers. However, we assume the adversary has already compromised one or more devices and gained read access to the victim's dataset for replication and preprocessing. Subsequently, the adversary creates a local NIDS model to mimic the victim NIDS model and craft adversarial examples that deceive the local model. Finally, these adversarial samples are transferred to evade the victim NIDS.

Preserving the functional behavior of the attack

- Feature type: To ensure the generated adversarial examples do not modify the feature type we proceed as follows. For binary features, we use a threshold of 0.5 to keep their values into binary values. If the value of the perturbation is less than 0.5, we change it to 0, and 1 if greater or equal to 0.5. In addition, we do not add perturbation to the non-numerical features, such as the one-hot encoded features converted from the categorical features (for instance, the protocol type).
- Valid range: We identify the minimum and maximum values for each feature in the dataset, defining the interval between these values as the valid range. After generating perturbations, we ensure the modified features remain within this range by applying the clipping method. This approach keeps perturbations within the predefined limits, preventing out-of-range values. For instance, if an adversarial sample exceeds a feature's maximum value, we clip it to the maximum value.

– Functional features: To conduct the attack, we begin by analyzing the functional and non-functional features within the dataset. We then generate adversarial examples, by adding perturbation on the non-functional features. These non-functional features are those that can be altered using legitimate transformations. This approach ensures that the intended meaning of the attack traffic remains unchanged and minimizes the risk of invalidating the data format.

As shown in Algorithm 1, we enforce the constraints by adding a mask on the features we do not want to add perturbation (the functional features) at each iteration of the training. This is done by creating a mask vector and using the element-wise multiplication, which is the Hadamard product, to maximize the loss only on the allowed features shown in table 1. Here, δ is the perturbation we want to add to the feature vector, mask $\boldsymbol{v}$ is the constraint, and $\odot$ is the element-wise multiplication.

Table 1: Illustration of the constraint

Vector type	x1	x2	x3	x4	x5	x6	x7	...	xn
perturbation δ	δ_{x1}	δ_{x2}	δ_{x3}	δ_{x4}	δ_{x5}	δ_{x6}	δ_{x7}	...	δ_{xn}
mask vector ($\boldsymbol{v}$)	1	0	1	1	0	1	0	...	1
$\delta \odot \boldsymbol{v}$	δ_{x1}	0	δ_{x3}	δ_{x4}	0	δ_{x6}	0	...	δ_{xn}

$$\boldsymbol{v_k} = (v_1, v_2, ..., vn)^T \qquad \boldsymbol{v_k} = \begin{cases} 1, & \text{if } k \in \text{independent feature} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

3.3 Generating adversarial samples

Although adversarial samples are often transferable, we should also consider the optimal approach to improve the transferability. In this paper, we propose C-MIFGSM to enhance the transferability of adversarial examples within a constrained domain. C-MIFGSM is an extension of the MIFGSM developed by Dong et al. [13]. MIFGSM integrates the momentum method into iterative gradient-based attack methods to mitigate poor transferability and enhance the success rate of black-box attacks.

As the algorithm allows perturbation on any feature, our proposed approach extends MIFGSM by incorporating constraints to limit perturbations using a mask vector and controlling the perturbation on binary features. Algorithm 1 illustrates the extended approach.

4 Experiments and Evaluation

In this section, we first train the adversary's local network, a DNN model to achieve a matching performance with the state-of-art performance on recent

Algorithm 1 Crafting adversarial examples for NIDS

Input:
f: Classifier with loss function J.
x: Real example input.
y: Ground-truth label.
ϵ: Size of perturbation.
T: Number of iterations.
μ: Decay factor.
mask: mask to restrict perturbations.
Output:
x^*: Adversarial example with $\|x^* - x\|_\infty \leq \epsilon$.

$\alpha \leftarrow \epsilon/T$
$g_0 \leftarrow 0$
$x_0^* \leftarrow x$
for $t = 0$ to $T - 1$ **do**
    Input x_t^* to f and obtain the gradient $\nabla_x J(x_t^*, y)$
    Update g_{t+1} by accumulating the velocity vector in the gradient direction:
        $g_{t+1} \leftarrow \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1}$
    Update x_{t+1}^* by applying the sign gradient and mask:
        $x_{t+1}^* \leftarrow x_t^* + \alpha \cdot \text{sign}(g_{t+1}) \odot \boldsymbol{v}$
    Add binary constraints
    Clip the samples within the valid range
end for
return $x^* = x_T^*$

works on NIDS and then generate adversarial examples in the local model with the proposed algorithm. Finally, the generated examples are transferred to mislead various target models. These target models include several classical machine learning and two DL models.

4.1 Dataset and pre-processing

We evaluate our approach using the NSLKDD dataset. NSLKDD is the most used dataset in academia for NIDS. It is a public dataset built by Tavallaee et al. [16] as an improved version of the KDD'99 dataset. In this work, we use the NSLKDDTrain+ as our training data and NSLKDDTest+ as test data. The dataset contains 41 features. These features can be divided into four different categories: (1-9) basic features, (10-22) content features, (23-31) time-based features, and (32-41) host-based features. The records of attacks in the dataset can be grouped into four categories of attacks: denial of service (DoS), probing (probe), root to local (R2L), and user to root (U2R) attacks, and the rest of the records represent normal traffic flow. Table 2 shows the functional features of each attack category (✓). To ensure the crafted adversarial examples will not

invalidate the network traffic, we add perturbation to the non-funtional features only as mentioned in section 3.

Since both DoS and Probe are network-based attacks, we focused our attack on DoS traffic. Additionally, we crafted attacks based on traffic content, such as U2R and R2L. Given the similar characteristics and functional features of U2R and R2L and their relatively small presence in the dataset, we combined U2R and R2L into a single attack group for our study.

The dataset contains categorical features, such as 'flag', 'protocol', and 'service'. In the preprocessing, we encode the categorical feature using one-hot encoding. After preprocessing, we obtain 122 features, which are then normalized between 0 and 1 using the min-max scaler in equation (2).

$$x_{scaled} = \frac{x - Min(x)}{Max(x) - Min(x)} \tag{2}$$

Table 2: Functional features of attack categories in NSLKDD

Attack types	Functional features			
	Intrinsic	Content	Time-based	Host-based
DoS	✓		✓	✓
Probe	✓		✓	
R2L	✓	✓		
U2R	✓	✓		

4.2 Evaluation metrics

To evaluate the work, we use two common metrics in the literature: the detection rate and the evasion increase rate. the detection rate (DR), also called the recall rate evaluates the ability of the model to detect the classes correctly. The evasion increase rate (EIR) measures the effectiveness of the attack. These metrics are shown in equations (3), and (4) respectively.

$$DR = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$EIR = 1 - \frac{Adversarial\ DR}{Original\ detection} \tag{4}$$

4.3 Experimental results

Attacker local NIDS model The DNN model is built with 4 hidden layers, using RELU as an activation function, Adam with a learning rate of 0.001 as an optimizer, plus early stopping during the training to avoid overfitting. The performance of the model is reported in Table 3.

Once we have evaluated the performance of our local NIDS, we generate adversarial samples with the C-MIFGSM method shown in Algorithm 1. As one of our goals is to remain stealthy by using small perturbations that are not easily detected, we transfer the adversarial examples crafted with a maximum magnitude of perturbation $\epsilon = 0.2$ to our victim models (where at each iteration, a perturbation of 0.02 was added to the legitimated input, and the number of iterations T=10).

Table 3 shows the performance of the local model under attack with $\epsilon = 0.2$. The model performance is significantly degraded. We observe the DR drops from 92.23% to 22.44% for DoS traffic with an EIR of 73.38%. With U2R&R2L attacks, the generated adversarial examples decreased the model DR from 81.19% to 4.17%. The results from Table 3 show the attack algorithm misled the classifiers and considerably degraded its detection performance.

Table 3: Performance of the DNN local model under adversarial attack

Metrics	DoS	U2R&R2L
DR before attack	92.23%	81.19%
DR after attack	22.44%	4.17%
EIR	73.38%	94.85%

Target NIDS models After generating the adversarial samples on the local NIDS, in this section, we perform the attack by transferring the generated adversarial samples to the victim models.

We implement the victim NIDS models using various classical machine learning RF, DT, XGBoost, AdaBoost, and MLP. The choice of using RF, XGBoost, and Adaboost is because they are ensemble learning algorithms, and studies [22, 23] have shown ensemble model's performance outperforms other traditional ML models. The traditional machine learning models in our experiments are implemented using the scikit-learn library with their default values, and the neural networks are implemented using the open-source deep learning library TensorFlow 2. The MLP is built with one hidden layer of 100 units in both datasets with RELU as activation, and a batch size of 32. We train the model using early stopping with a patience of 50. In addition, we implement two state-of-the-art deep learning models AlerNet [17] and DeepNet [18] to evaluate the effectiveness of our algorithm. We choose to evaluate the robustness of the works in [17, 18] because their proposed NIDS achieved a high performance compared to existing works [19] and are among the most cited DL-based NIDS published recently.

Transferability attack on the target models Tables 4 and 5 present the performance of ML and DL NIDS models, respectively, under transfer-based at-

tack. Initially, all models achieved performance comparable to state-of-the-art NIDS, with DL models exhibiting the highest performance. However, adversarial examples are observed to be transferable across models implemented with different techniques.

Tables 4 and 5 illustrate that these attacks significantly degrade the detection rates (DR) of all NIDS target models. For example, the detection rate of the DoS traffic drops from 92.35% to 20.27% for the MLP model with an evasion increase rate of 78.04%. Similarly, for the RF model, the DR decreased from 91.59% to 85.53%, indicating that 6.61% of DDoS traffic evaded detection (EIR = 6.61%). Furthermore, 86.13% of the generated adversarial U2R&R2L traffic evaded the DT model (EIR = 82.13%), and 92% evaded the Alernet model.

Despite RF demonstrating the most robust performance against these attacks, evasion rates of 6.61% for DoS and 2.07% for U2R&R2L are intolerable in domains such as NIDS. In general, the ML models demonstrated to be less vulnerable than the DL models.

These results highlight that adversaries can successfully mount attacks against NIDS models with limited knowledge and under constraints. Additionally, adversarial samples are transferable from DL to classical ML models, although their impact on different classifiers varies.

Table 4: Performance of the transferability of CMIFGSM attack on the ML-based NIDS models

ML/DL model	RF		DT		XGBoost		Adaboost		LR	
Metrics	DoS	U2R_R2L	DoS	U2R_R2L	DoS	U2R_R2L	DoS	U2R_R2L	DoS	U2R_R2L
DR before attack	91.59%	78.06%	92.08%	79.90%	92.38%	79.04%	90.59%	79.20%	89.34%	76.85%
DR after attack	85.53%	76.44%	91.45%	11.07%	89.11%	18.66%	70.20%	31.59%	59.50%	63.93%
EIR	6.61%	2.07%	0.68%	86.13%	3.54%	76.39%	22.51%	60.10%	33.40%	16.81%

Table 5: Performance of the transferability of CMIFGSM attack on the DL-based NIDS models

ML/DL model	MLP		DeepNet		Alernet	
Metrics	DoS	U2R_R2L	DoS	U2R_R2L	DoS	U2R_R2L
DR before attack	92.35%	81.79%	91.71%	81.34%	90.60%	80.08%
DR after attack	20.27%	9.54%	32.26%	13.38%	51.16%	5.65%
EIR	78.04%	88.34%	64.81%	83.54%	43.35%	92.93%

5 Conclusion

In this work, we investigated evasion attacks against ML-based NIDS. Our objective was to perform a practical gray box attack that can deceive the model

and reduce its effectiveness. We proposed an attack algorithm, C-MIFGSM, to craft small, specially designed perturbations to mislead the target classifiers. The findings of this work demonstrate that adversaries can successfully mount adversarial attacks to fool ML-based NIDS without knowledge of the target model and under constraints. This raises significant security concerns for developing robust ML-based NIDS.

In future work, we plan to extend our investigation to various datasets and explore defenses against adversarial attacks in NIDS.

Acknowledgment

The 1st author is supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT). The 2nd author is supported by the Spanish Ministry of Universities and the European Union - NextGeneration EU funds through the Grants for the Requalification of the Spanish University System 2021-2023 (University of Malaga, Resolution July 1st 2021). The 4th author is partially supported by the International Exchange, Foreign Researcher Invitation Program of National Institute of Information and Communications Technology (NICT). This work was partially supported by the India-Japan Cooperative Science Programme (IJSCP) through the Department of Science and Technology (DST, India) and the Japan Society for the Promotion of Science (JSPS).

References

1. Axelsson, Stefan. “Intrusion Detection Systems: A Survey and Taxonomy.” (2002).
2. Liu, Hongyu and Bo Lang. “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey.” Applied Sciences (2019): n. pag.
3. Szegedy, Christian et al. “Intriguing properties of neural networks.” CoRR abs/1312.6199 (2013): n. pag.
4. Goodfellow, Ian J. et al. “Explaining and Harnessing Adversarial Examples.” CoRR abs/1412.6572 (2014): n. pag.
5. Akhtar, Naveed and Ajmal S. Mian. “Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey.” IEEE Access 6 (2018): 14410-14430.
6. Papernot, Nicolas et al. “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples.” ArXiv abs/1605.07277 (2016): n. pag.
7. Guan, Zhenyu et al. “When Machine Learning meets Security Issues: A survey.” 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR) (2018): 158-165.
8. Apruzzese, Giovanni et al. “Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems.” Digital Threats: Research and Practice (DTRAP) 3 (2021): 1 - 19.
9. Martins, Nuno et al. “Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review.” IEEE Access 8 (2020): 35403-35419.
10. Mbow, M., Sakurai, K., Koide, H. (2022). Advances in Adversarial Attacks and Defenses in Intrusion Detection System: A Survey. In: Su, C., Sakurai, K. (eds) Science of Cyber Security - SciSec 2022 Workshops. SciSec 2022. Communications in Computer and Information Science, vol 1680. Springer, Singapore.

11. Usama, Muhammad et al. "Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems." 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (2019): 78-83.
12. He, Ke et al. "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey." IEEE Communications Surveys & Tutorials 25 (2023): 538-566.
13. Dong, Yinpeng et al. "Boosting Adversarial Attacks with Momentum." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017): 9185-9193.
14. Duy, Phan The et al. "DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks." Comput. Secur. 109 (2021): 102367.
15. Biggio, Battista et al. "Evasion Attacks against Machine Learning at Test Time." ArXiv abs/1708.06131 (2013): n. pag.
16. Tavallaee, Mahbod et al. "A detailed analysis of the KDD CUP 99 data set." 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (2009): 1-6.
17. Vinayakumar, R. et al. "Deep Learning Approach for Intelligent Intrusion Detection System." IEEE Access 7 (2019): 41525-41550..
18. Gao, Minghui et al. "Malicious Network Traffic Detection Based on Deep Neural Networks and Association Analysis." Sensors (Basel, Switzerland) 20 (2020): n. pag.
19. Zolbayar, Bolor Erdene et al. "Generating Practical Adversarial Network Traffic Flows Using NIDSGAN." ArXiv abs/2203.06694 (2022): n. pag.
20. Apruzzese, Giovanni et al. "Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems." Digital Threats: Research and Practice (DTRAP) 3 (2021): 1 - 19.
21. McCarthy, Andrew et al. "Functionality-Preserving Adversarial Machine Learning for Robust Classification in Cybersecurity and Intrusion Detection Domains: A Survey." J. Cybersecur. Priv. 2 (2022): 154-190.
22. Chen, Tianqi and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016): n. pag.
23. Rokach, Lior. "Ensemble-based classifiers." Artificial Intelligence Review 33 (2010): 1-39.

Enhancing Robustness in NIDS via Coverage Guided Fuzzing and Adversarial Training

Jung Yup Rhee[1], Leo Hyun Park[2], and Taekyoung Kwon[1,2]

[1] Graduate School of Artificial Intelligence, Yonsei University, Seoul, Korea
[2] Graduate School of Information, Yonsei University, Seoul, Korea
{wjdduq234, dofi, taekyoung}@yonsei.ac.kr
http://seclab.yonsei.ac.kr/

Abstract. AI-based Network Intrusion Detection Systems (NIDS) are essential for defending modern network environments against diverse traffic-based attacks. However, these systems, including the advanced Kitsune model which employs an ensemble of small autoencoders for real-time intrusion detection, are susceptible to adversarial attacks. Such attacks manipulate the AI models by introducing adversarial examples, potentially leading to system failure. This paper introduces a novel methodology that applies coverage-guided fuzzing, a technique traditionally used in software vulnerability detection, to uncover adversarial examples in NIDS models. Furthermore, we present an innovative approach of how to enhance robustness of these systems using adversarial training with the identified adversarial examples. We demonstrate the effectiveness of our approach by applying it to Kitsune-based models with two open-source benchmark datasets. The experimental results indicate that the defense rate against adversarial attacks has improved by up to 98%, proving that our methods not only detect critical vulnerabilities but also significantly strengthen the model's defense mechanisms against such attacks.

Keywords: Network Intrusion Detection Systems · Anomaly Detection · Coverage Guided Fuzzing · Adversarial Training.

1 Introduction

Cyber attacks through networks are becoming increasingly frequent. Furthermore, the proliferation of IoT devices and the increasing volume of data transmitted and received by mobile devices have expanded the targets for network attacks [1]. Consequently, sophisticated and difficult-to-detect network attack techniques that target computer or databases, such as CHARGEN and Ubiquiti, have been developed. These attacks pose significant social problems by causing various economic issues, such as privacy leakage, or insertion of malicious code, affecting not only corporations but also individuals. To mitigate such damages, deep learning based NIDS (Network Intrusion Detection System) models that detect network attacks are being researched and developed.

However, existing NIDS research has primarily focused on improving accuracy, neglecting aspects of robustness verification and enhancement. Consequently, NIDS are highly vulnerable to adversarial attacks that can cause malfunctions, such as judging network attacks as benign, thereby rendering NIDS models ineffective and exposing them to various attacks. To develop practical NIDS, it is essential to enhance their security through robustness verification and reinforcement. Despite the long-standing recognition of this issue, the challenge of addressing NIDS robustness remains significant due to the scarcity of prior research and the inherent difficulty of the problem.

Thus, this paper proposes methodologies for verificating and enhancing robustness to aid the development of robust and secure NIDS technologies. Robustness verification is conducted using coverage guided fuzzing, which leverages new mutation and coverage metrics. Further on, for robustness enhancement, we propose a novel methodology that utilizes the vulnerabilities discovered during robustness verification for adversarial training. The effectiveness and validity of the proposed methodologies are demonstrated through experiments.

The validation experiments of our methodology are conducted on the Kitsune model [2]. Kitsune, with its lightweight architecture, is suitable for online processing and real-time inference, and can be applied to datasets in CSV format without preprocessing, making it a practical and scalable. Its technical excellence is further evidenced by its active use in subsequent research [3–5]. However, since the datasets used in the Kitsune paper for reproduction are unavailable, we reproduced the experiments using other open-source datasets to find out that the model did not generalize well, resulting in poor accuracy. Consequently, we undertake the process of improving the open-source Kitsune model to contribute to future NIDS research and performance development.

Our study has three main contributions:

1. We introduce suitable mutation techniques for robustness verification and analyze optimal coverage and fuzzing conditions. We then conduct Coverage Guided Fuzzing on the Kitsune model to demonstrate the effectiveness of this methodology in robustness verification through vulnerability detection.
2. We propose a novel dual-structure training method, divided into alpha and beta models. Vulnerabilities discovered during the robustness verification phase are utilized as input for the beta model in adversarial training, aiming to simultaneously improve robustness and accuracy. This study aims to identify the limitations of currently utilized network traffic detection models, providing directions for not only performance improvement but also the development of robust networks in the future.
3. To enhance the performance and generalization of the open-source Kitsune model, we adjust model architecture, modify training methods, and optimize parameters, and prove the improvements using open-source datasets.

2 Background

2.1 Network Intrusion Detection System

NIDS is an anomaly detection technique, detecting if the traffic sender is an adversary. The detection methodologies are based on classification, Nearest Neighbor, clustering, and etc [5, 6]. Unsupervised deep learning-based anomaly detection uses an autoencoder and is trained with only normal traffic data. Inside the autoencoder, the input data is compressed and reconstructed to minimize reconstruction error, called root mean squared error (RMSE). As a result, if RMSE is below the threshold, the traffic is classified as normal; otherwise, it is classified as malicious. A representative example of NIDS is Kitsune [2]. Kitsune is divided into the tail network and head network: the tail consists of an ensemble of autoencoders, each handling one of the k clusters of feature information, and the head is a single autoencoder that receives the output values of the k autoencoders from the tail.

2.2 Adversarial Robustness

Adversarial attacks refer to attacks that induce malfunctions in a deep learning model by exploiting features such as gradients to create adversarial perturbations. Starting with FGSM [7], various methods such as PGD [8] and BIM [9] have been developed to strengthen these attacks. These attacks cause critical defects in safety, finance, and security, such as misrecognition of pedestrians by autonomous vehicles or data theft through network intrusion attack. Consequently, adversarial defense techniques have been developed to block these attacks. One example of such a defense technique is adversarial training [10], which involves generating adversarial data from the model to train with it. While this is a simple and effective defense technique, it has an accuracy-robustness trade-off issue, where an increase in robustness leads to a decrease in performance, necessitating improvements.

NIDS models are used to detect malicious network traffic, but they have not been prepared for attacks that target NIDS models themselves. While it is very easy to induce misclassification by conducting adversarial attacks on these models, there has been no research that verifies the robustness of models against such attacks. For instance, when we conduct an adversarial attack with an epsilon of 0.05 on 1000 data targeting the Kitsune model (trained with CICDDoS2019 dataset, AUC reaching 0.9609), an average of 995 attacks succeeded. Therefore, in addition to accuracy, NIDS must be robust against adversarial attacks to be a realistic and practical detection model, emphasizing the importance of robustness verification and enhancement.

3 System Design and Implementation

3.1 System Outline

The threat model is that the adversary aims to deceive the NIDS with their malicious traffic, causing it to be misclassified as normal for further cyber attacks.

This scenario assumes a white-box situation where the attacker has information about the target NIDS, which is the Kitsune model. Therefore, from the defender's perspective, we need to ensure the performance of NIDS for proper judgment and robustness against adversarial attacks.

In the traditional anomaly detection environment, only normal data is accessible to train an unsupervised model. However, it is practically challenging to verify or enhance robustness without malicious data. Additionally, as of 2024, various fields have collected malicious data over several years. While using tens of thousands of malicious data points may be unrealistic, utilizing only a small amount is reasonable. Thus, we assume that the defender has access to such small-scale malicious data.

In this study, we first improve the performance of an open-source model and verify its robustness by conducting coverage-guided fuzzing. Adversarial examples discovered during the fuzzing process will be used for adversarial training to improve the robustness of the model.

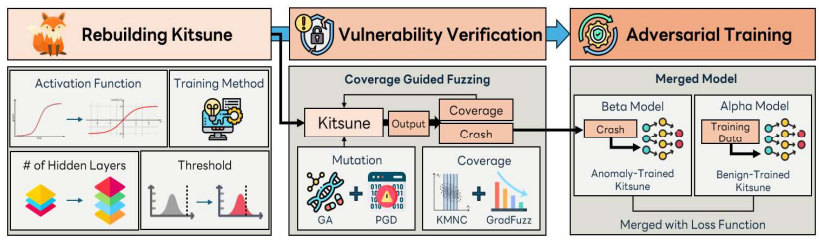


Fig. 1. Outline of the Research Conducted in this Paper

3.2 Improving Open-Source Kitsune

The performance of Kitsune in the official repository [11] and the unofficial PyTorch version [12] differed significantly from the original paper, making it unsuitable for direct use. Therefore, we applied some modifications to the Kitsune model. The entire code is reconstructed from Numpy to PyTorch. We also added a normalization layer before the tail and head layer to not only increase the computational speed but also stabilize the model output, preventing a local optima. Further on, the activation function has changed from Sigmoid to Hyperbolic Tangent ($tanh$) and the number of hidden layers has increased from 10 times to provide more space for the features to get trained.

The original learning rate was fixed at $1e^{-3}$, but a learning rate scheduler is applied that gradually decreases from $1e^{-3}$ to $1e^{-4}$. Also, the default threshold for determining anomalies was set to the maximum value of the training loss, but is modified to the sum of the mean and standard deviation of the training loss, significantly increasing true positives with a slight decrease in true negatives. The training method of the original model trains both the tail and the head simultaneously on a single batch of data. However, due to the low accuracy, we train the tail first, followed by the head.

3.3 Dataset and Dataloader

The dataset used for training Kitsune was collected directly from the author and is not publicly available, making it inaccessible. Therefore, we utilize the CICDoS2019 [13, 14] and NSL-KDD [15] datasets, which are benchmark datasets that have been used or analyzed in over 15,000 and 2,000 studies on network intrusion, respectively, and are still actively used today.

We designed a new dataloader for continuous learning. The original dataloader fetches data one by one when forming batches, which decreases the training speed. In addition, it leads to performance change due to not utilizing the entire dataset during the generalization and subsequent shifts in data distribution. Our dataloader improves speed and learning stability by initially loading all samples from the dataset and then using a fixed number of samples for batches.

3.4 Robustness Verification

Robustness verification in anomaly detection is the process of checking whether the model can be induced to misclassify anomalies as normal data. Therefore, if the transformed anomalies are successfully inferenced as benign data, they are defined as a vulnerability of the target NIDS.

Coverage Guided Fuzzing Coverage-guided fuzzing is a technique that generates diverse test data using coverage to search for vulnerabilities. While traditional software testing utilizes simple coverage metrics like code coverage, which measures the percentage of codes executed, deep learning models have to use special coverage metrics because they have neuron-based structures, in contrast with code-based software. Similarly, the effectiveness of mutation techniques (i.e., simple transformation algorithms such as replace and modify) may vary depending on the target model and dataset. Thus, the dedicated coverage metric [16] and mutation techniques [17–19] should be selected for coverage-guided fuzzing. The comprehensive process of coverage-guided fuzzing for NIDS models proceeds as Algorithm 1. The details are explained below.

Mutation for Coverage Guided Fuzzing The mutation process transforms the data of a parent seed from a seed corpus to generate a new child. The mutation techniques used in our study are based on genetic algorithms (GA) [20] and adversarial perturbation (PGD). GA mutation is inspired by the actual mutation process of genes. The main procedure of GA include selection, crossover, and mutation. Selection means fixing parts that should not undergo transformation, while crossover mixes data with target datum in unfixed parts. Mutation is adding noise to the data, overcoming the limitation of GA in potentially falling into local optima due to transformations within a limited data set. In the case of network traffic features composed of integers, we apply only the selection and crossover processes. Conversely, for traffic features composed of floats, all three processes are executed. Adversarial perturbation-based mutation uses PGD that

Algorithm 1 Coverage Guided Fuzzing

```
Iteration ← 0
Seed_Corpus ← [] # Initialize seed corpus
adversarial_examples ← [] # Initialize detected adversarial examples
while Iteration < 1000 do
    Iteration ← Iteration + 1
    parent ← RandomSelect(Seed_Corpus)
    mutation_func ← RandomSelect([GA, PGD])
    child ← mutation_func(parent)
    output ← target_model(child)
    if is_new_coverage(child.coverage) then
        Seed_Corpus.append(child)
    end if
    if is_adversarial(output) then
        adversarial_examples.append(child)
    end if
end while
```

induces malfunction by adjusting inputs in a way that maximizes the loss value based on the model information. Scaling and clamping are applied to keep feature values within the valid range to prevent excessive modification.

In the unsupervised model Kitsune, the mutation is applied to anomaly data. Anomaly data outputs a loss value above the threshold according to the NIDS model. The main objective is to modify the feature values so that the resulting loss is below the threshold, causing it to be misclassified as benign. Therefore, a crash is defined as an anomaly data that outputs a loss value that is lower than the threshold, deceiving the NIDS model to interpret it as a normal data. Crashes are used as adversarial examples for adversarial training.

Coverage Measurements Coverage metrics are divided into vector-based and distance-based. Vector-based metrics update the coverage map when some vector elements are newly activated, while distance-based ones update the coverage when the vector values are sufficiently far from the existing inputs. In our study, we use KMNC [21], as a vector-based metric, that divides the output range of the activation function of each neuron into k segments and measures which region has been activated. For a distance-based metric, we use NLC [22] which calculates the covariance matrix (distance) between two neurons in the same layer.

Additionally, we use GradFuzz [23], a gradient-based coverage metric that more accurately identifies guidance for inducing misclassification. GradFuzz uses the gradient of neurons according to the loss function as coverage, instead of neuron activation that only explains the output results, enabling a more logical explanation for both the output results and their causes due to the input. Therefore, experiments are conducted using four coverage methods: KMNC, NLC, GradFuzz+KMNC, and GradFuzz+NLC.

3.5 Adversarial Training

Adversarial training trains the model with adversarial examples to enhance robustness, but unsupervised models, unlike typical models, only learns from

normal-class data, making it impossible for us to utilize the adversarial examples collected during the robustness verification. Therefore, a model trained using only the adversarial data (hereafter referred to as the *beta model*), is merged with the existing model which was trained with only the normal samples (referred to as the *alpha model*). The beta model is also based on Kitsune, having an identical structure to the alpha model. When the merged model receives an input, the alpha and beta models process the input simultaneously, resulting in two respective loss values. The final loss value is then computed as below.

$$Loss^M = Loss^\alpha + \frac{1}{Loss^\beta}, \quad Threshold^M = mean(loss^M) + std(loss^M)^1$$

The threshold for judging benign is applied in the same way as before, using the sum of the mean and standard deviation of the loss values from the benign data.

4 Results

4.1 Research Environment

The string-based feature information in both datasets we use are removed, and the model is trained with a batch size of 200. The GPU is used only for speed measurement while CPU is used for all the other experiments.

4.2 Improving Performance of Open-Source Kitsune

The original open-source Kitsune model achieved an AUC of 0.8800, 0.4990 each for CICDDoS2019 and NSL-KDD dataset. After modifying the model structure and parameters, as mentioned in Section 3.2, the Kitsune model achieved an AUC of 0.9609, 0.9392 each for the two datasets, showing significant improvement compared to the open-source. Additionally, we examine the impact of the number of ensembles on performance and inference speed. The results showed no correlation between the number of ensembles and performance, which led us to use no ensembles for future experiments. In terms of inference speed, PyTorch CPU was the fastest, followed by Numpy CPU and PyTorch GPU, indicating that online processing can be maintained similarly like the original model.

4.3 Robustness Verification

Mutation Techniques for Fuzzing In this study, we compared the performance of the mutation algorithms within two different environments. The first environment, shallow mutation, discards all mutated seeds regardless of coverage improvement, maintaining only the initial seeds in the seed corpus. Consequently, all child seeds have undergone only one mutation. The second environment, deep mutation, adds child samples that improve coverage to the seed corpus, following

[1] M: Merged Model, α : Alpha Model, β : Beta Model

Table 1. Experiment Result of the Mutation Methods

Dataset		CICDDoS2019		NSL-KDD	
Mutation Method		# Crashes	% Successful Testing	# Crashes	% Successful Testing
Deep Mutation	Gaussian Noise	5	5%	0	0%
	PGD	**569**	**100%**	**38**	**24%**
	GA	282	91%	5	5%
	PGD+GA	516	99%	26	21%
Shallow Mutation	Gaussian Noise	27	23%	0	0%
	PGD	**998**	**100%**	**47**	**19%**
	GA	403	89%	0	0%
	PGD+GA	702	**100%**	23	15%

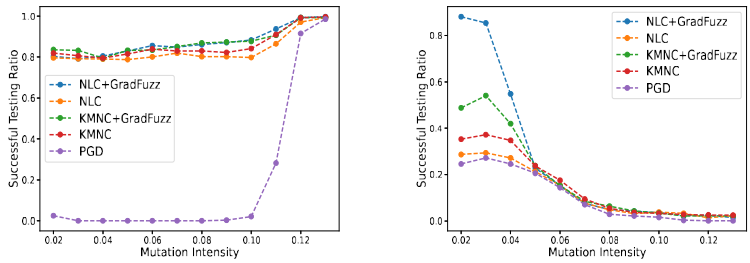
the traditional fuzzing process. We verified PGD-based and GA-based mutation techniques and their combination, against a control group using Gaussian-based mutation. The Gaussian-based mutation adds values derived from a Gaussian distribution with a mean of 0 and a standard deviation of 1, scaled by multiplying 0.1. We perform 30 mutations for each of the 100 initial seeds. We used two evaluation metrics: the total number of adversarial examples (crashes) found and the percentage of initial seeds that reached at least one adversarial example through mutation (successful testing).

As shown in Table 1, NSL-KDD is found to be more challenging to detect adversarial examples compared to CICDDoS2019. Random mutations using Gaussian noise rarely detect adversarial examples, while PGD and GA achieve significant performance in vulnerability detection. Although PGD alone detects the most adversarial examples, it results in limited diversity due to its reliance solely on gradients. Thus, using both PGD and GA simultaneously is recommended to ensure diversity and realism.

Coverage Guided Fuzzing Results Coverage-guided fuzzing in this experiment performs 500 input mutations for the 100 initial seeds. The input chooser selects seeds based on the probability of Depth/Count (Depth: the number of mutations applied on the seed; Count: the number of times the seed has been selected as a parent seed). We utilized a combination of GA and PGD for the mutation of fuzzing. All the other environments remain the same as in the previous experiments.

Table 2. Experiment Result of Diverse Testing Environments

Dataset		CICDDoS2019		NSL-KDD	
Testing Environment		# Crashes	% Successful Testing	# Crashes	% Successful Testing
Gaussian Noise		10.7	9.3%	0	0%
Mutational Testing	Deep Mutation	6376.7	**100%**	4193.3	**100%**
	Shallow Mutation	3659.3	94.7%	1924	36%
Adversarial Attack (PGD)		31.3	31.3%	3.3	3.3%
Coverage Guided Fuzzing	KMNC	6103.7	**100%**	4437.7	**100%**
	KMNC+GradFuzz	5309.7	98.7%	5102.7	93.7%
	NLC	4229.2	99.7%	**7712**	**100%**
	NLC+GradFuzz	**7534.7**	**100%**	4647.3	99.7%



(a) CICDDoS2019 (b) NSL-KDD

Fig. 2. Crash Discovery Rate for Each Sample According to the Attack Intensity

The results in Table 2 indicate that random noise is ineffective in discovering vulnerabilities, which demonstrates the superiority of the mutation techniques. Furthermore, the adversarial example detection performance is best in the order of coverage-guided fuzzing, deep mutation, and shallow mutation, proving that repetitive mutation and coverage enhance the performance of vulnerability detection. The detection performance varies depending on the dataset and coverage method, highlighting the need for appropriate coverage metrics tailored to specific situations. Lastly, we compared the vulnerability detection performance of each coverage metric through the successful testing rate across ten different mutation intensities. The intensity of each mutation and the successful testing results are shown in Fig 2.

The results show that coverage-guided fuzzing outperforms PGD, and NLC with GradFuzz generally achieves the best performance among the others. Additionally, while the number of crashes detected in CICDDoS2019 increases with higher attack intensities, NSL-KDD shows more crashes with lower attack intensities, indicating overfitting or other issues in the model trained with NSL-KDD.

4.4 Increasing Robustness and Performance with Adversarial Training

In this experiment, we leverage the adversarial examples detected through vulnerability detection for adversarial training of the beta model. We merge the beta model with the alpha model to create a merged model with an integrated loss function. We perform a comparative experiment to identify the correlation between the number of training data and the performance by increasing the number of training data from 100 to 500. We consider a comprehensive environment called *Total*, where all adversarial examples generated from the combination of KMNC, NLC, and GradFuzz are used for adversarial training.

Performance of Beta Model The performance of the beta model shows significant variance in the training data, with an average AUC value of 0.6852, a

Table 3. AUC of Merged Model

Dataset		NSL-KDD						CICDDoS2019					
Coverage Metric		KMNC+ GradFuzz	KMNC	NLC+ GradFuzz	NLC	Total	PGD	KMNC+ GradFuzz	KMNC	NLC+ GradFuzz	NLC	Total	PGD
100	Train	0.5424	0.5176	0.9885	0.9761	0.6119	**0.9889**	0.9822	0.9683	0.9848	0.9759	0.9836	**0.9947**
	Test	0.4694	0.4455	**0.994**	0.9879	0.5538	0.9931	0.9905	0.9855	0.9907	0.9869	0.9925	**0.9974**
200	Train	0.5487	0.9895	0.5251	0.8479	0.9788	**0.9953**	**0.9935**	0.9908	0.9878	0.5019	0.9838	0.4855
	Test	0.4627	0.9938	0.4397	0.7593	0.989	**0.9978**	0.9905	0.9943	**0.996**	0.4276	0.9922	0.4303
300	Train	0.9928	0.9796	0.9848	0.976	**0.9949**	0.5373	0.9904	0.9911	0.9904	**0.9912**	0.989	0.5244
	Test	0.9962	0.9876	0.9935	0.987	**0.9973**	0.461	**0.9968**	0.9944	0.9944	0.9949	0.9938	0.4453
400	Train	0.9871	**0.9879**	0.9872	0.6732	0.5176	0.9857	0.9882	0.6516	0.5707	0.9857	**0.9893**	0.9892
	Test	**0.9927**	0.9922	0.9915	0.5667	0.4382	0.9914	0.9929	0.5539	0.4868	0.9937	0.9939	**0.9945**
500	Train	0.9858	0.9442	0.9829	0.9834	**0.9869**	0.5867	0.9861	**0.9897**	0.9892	0.5138	0.983	0.5718
	Test	0.9912	0.9415	0.9918	0.9917	**0.9922**	0.5092	0.9919	**0.9946**	0.994	0.436	0.9919	0.4911

minimum value of 0.5123, and a maximum value of 0.8063, while the test data shows an average of 0.6214, a minimum of 0.6054, and a maximum of 0.6966. This indicates that the model cannot reach the global optimum due to the diversity of adversarial examples. We observed the large variation in performance when repeatedly training on the same data, suggesting a significant impact of the initial seed regarding the local optimum. Furthermore, increasing the number of adversarial examples does not lead to a notable improvement.

Performance of Merged Model The AUC of the merged model is shown in Table 3. The AUC of the alpha model was 0.9392 and 0.9609 for NSL-KDD and CICDDoS2019, respectively. Some merged models significantly outperform the baseline alpha model, but others show performance close to random guessing. However, the highest performance of 10 repetitions in all conditions exceeds the baseline, demonstrating that the accuracy increases as the model converges well.

Robustness of Merged Model We verified the robustness of the merged model by measuring the number of misclassifications from 1000 attack samples with an epsilon strength of 0.05. The baseline is the alpha model under adversarial attacks, with 362 misclassifications for the NSL-KDD and 988 for the CICDDoS2019. As shown in Table 4, the robustness of the merged model is superior in all conditions, except three conditions for the NSL-KDD, compared to the baseline. The overall increase in the robustness of the merged model is meaningful because the attack intensity is 0.05, which is relatively high.

Table 4. Robustness of Merged Model

Dataset	NSL-KDD						CICDDoS2019					
Coverage Metric	KMNC+ GradFuzz	KMNC	NLC+ GradFuzz	NLC	Total	PGD	KMNC+ GradFuzz	KMNC	NLC+ GradFuzz	NLC	Total	PGD
100	320	8	**0**	326	542	15	402	272	83	537	74	**0**
200	588	339	**0**	1	**0**	54	34	4	304	**0**	59	54
300	33	255	557	122	85	54	**0**	297	59	2	514	**0**
400	**0**	**0**	**0**	285	**0**	**0**	**0**	**0**	298	123	272	**0**
500	**0**	329	**0**	28	1	261	57	575	22	102	308	261

5 Discussion

5.1 Finding the Best Combination for Coverage Guided Fuzzing

Through our experiments, we compared the efficiency of various coverages in detecting vulnerabilities. While some techniques generally performed better, it was evident that the effectiveness of a specific coverage varied depending on the environment. This highlights the need to select an appropriate coverage and the necessity for developing a more effective coverage for deep learning fuzzing.

Additionally, the selection of the input chooser is crucial. The performance in terms of the number of adversarial examples found improves from shallow to deep to coverage guided fuzzing. This indicates that while a greater depth is necessary to find more adversarial examples, the method used to select the parent seeds for mutation is even more important. Furthermore, if the depth gets deeper, efficiency decreases, emphasizing the need for a superior input chooser.

5.2 Model Optimization

Both the alpha and beta models exhibit significant performance variability with each experimental run. This is because optimizing models separated into head and tail is challenging, due to the issue of fixing the unoptimized head to the initial seed while training the tail. If either the alpha or beta model is not properly optimized, the merged model will also exhibit poor accuracy and robustness. Although the performance improved with the methodology discussed in section 4.2, addressing these issues with new training mechanisms could lead to more stable and improved performance.

6 Conclusion

Although NIDS models exhibit high performance, they are vulnerable to adversarial attacks, leading to misclassification. Therefore, it is necessary to verify and enhance not only performance but also robustness. In this paper, we effectively improved the performance of the open-source Kitsune and developed new mutation combinations, proving their efficiency through experiments. Using these mutation techniques and diverse coverage-based fuzzing methods, we were able to detect vulnerabilities and verify robustness. Furthermore, we demonstrated the improvement in robustness and performance of NIDS models through a new adversarial training method. Consequently, we contributed to the development of secure network systems by analyzing and enhancing the safety of NIDS models.

For future work, we plan to research optimization techniques discussed in section 5.2 to achieve stable convergence. Additionally, we aim to develop coverage and fuzzing techniques dedicated to deep learning to fuzz diverse AI technology.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No.RS-2023-00229400) and (No.RS-2023-00230337).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Jeff Johnson, Fred Plan, et al, "3CX Software Supply Chain Compromise Initiated by a Prior Software Supply Chain Compromise", MANDIANT, 2023.04.20
2. Mirsky, Yisroel, et al. "Kitsune: an ensemble of autoencoders for online network intrusion detection." in Proc. NDSS, 2018.
3. Han, Dongqi, et al. "Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation." in Proc. the NDSS. 2023.
4. Fu, Chuanpu, et al. "Realtime robust malicious traffic detection via frequency domain analysis." in Proc. the CCS. 2023.
5. Du, Min, et al. "Lifelong Anomaly Detection through Unlearning." in Proc. the ACM SIGSAC Conference on Computer and Communications Security (CCS), 2019.
6. Andresini, Giuseppina, et al. "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection." in Proc. AISec, 2021.
7. Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
8. Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." in Proc. ICLR, 2018.
9. Kurakin, Alexey, et al. "Adversarial machine learning at scale." In Proceedings of the International Conference on Learning Representations. 2017.
10. Bai, Tao, et al. "Recent advances in adversarial training for adversarial robustness." arXiv preprint arXiv:2102.01356 (2021).
11. Ymirsky,"Kitsune",GitHub,2020.8.29,<https://github.com/ymirsky/Kitsune-py>
12. Guillem96. "Kitsune Network PyTorch." GitHub, kitsune-pytorch, 2022.03.31.,<https://github.com/Guillem96/kitsune-pytorch>
13. Sharafaldin, Iman, et al. "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy." 2019 IEEE ICCST, 2019.
14. "DDoS Evaluation Dataset" University of New Brunswick, Canadian Institute for Cybersecurity, <https://www.unb.ca/cic/datasets/ddos-2019.html>
15. "NSL-KDD dataset." University of New Brunswick, Canadian Institute for Cybersecurity, <https://www.unb.ca/cic/datasets/nsl.html>
16. Yuan, Yuanyuan, Qi Pang, and Shuai Wang. "Assessing deep neural network testing via neural coverage." arXiv preprint arXiv:2112.01955 (2021).
17. Xie, Xiaofei, et al. "Deephunter: a coverage-guided fuzz testing framework for deep neural networks." Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis. 2019.
18. Odena, Augustus, et al. "Tensorfuzz: Debugging neural networks with coverage-guided fuzzing." International Conference on Machine Learning. PMLR, 2019
19. Pei, Kexin, et al. "Deepxplore: Automated whitebox testing of deep learning systems." proceedings of the 26th Symposium on Operating Systems Principles. 2017
20. Max Maxfield, "When Genetic Algorithms Meet Artificial Intelligence", Electronic Engineering Journal, 2020.07.09,
21. Ma, Lei, et al. "Deepgauge: Multi-granularity testing criteria for deep learning systems." in Proc. the ASE, 2018.
22. Yuan, Yuanyuan, et al. "Revisiting neuron coverage for dnn testing: A layer-wise and distribution-aware criterion." in Proc. the ICSE, 2023.
23. Park, Leo Hyun, et al. "GradFuzz: Fuzzing deep neural networks with gradient vector coverage for adversarial examples." Neurocomputing 522 (2023): 165-180.

VoteGAN: Generalized Membership Inference Attack Against Generative Models by Multiple Discriminators

Gyeongsup Lim[0009−0006−5274−8135]*, Wonjun Oh*, and Junbeom Hur[0000−0002−4823−4194]

Korea University, Seoul, Korea
{gslim,wjoh,jbhur}@isslab.korea.ac.kr

Abstract. Generative Adversarial Networks (GANs) produces samples with the same probability distribution as the given training data. However, as they frequently train with sensitive data to generate realistic samples, many people are concerned about the leakage of training data from the GANs model, causing a severe breach of privacy in practice. This paper focuses on membership inference attacks aiming to reveal information about whether a certain data record is used in the target model training procedure. Although the membership inference attack has been successfully applied to various models, delivering membership inference attacks against generative models still remains a challenging problem, because the attack model could not accurately represent the target model. To solve this problem, we quantify the target model representation degree by measuring the attack model's generalization gap (i.e., the difference between the attack model's prediction distribution on training data and unseen data). To reduce the generalization gap, we propose a novel membership inference attack framework, called *VoteGAN*, consisting of multiple discriminators and one generator. VoteGAN trains the discriminators separately with a partition of the training data. It enables VoteGAN to approximate the mixture distribution of all partitions, allowing the reflection of the entire data more accurately. Our experimental results demonstrate that the proposed attack model outperforms prior attack method, showing 25% higher attack success rate on average.

Keywords: AI security · Machine learning · Computer vision.

1 Introduction

Over the last years, previous studies [29], [28], [15], [11], [2] have shown that deep learning models can be vulnerable to a *membership inference attack*, which aims to identify whether a given data was used for the target model training. In the membership inference attacks, the adversary trains a *shadow model* that

* Contributed equally.

mimics the behavior of the target model. The adversary then trains an *attack model* to detect the used data with the outputs of the *shadow model*. After the first membership inference attack was proposed against machine learning models [29], many membership inference attacks have been proposed on Generative Adversarial Networks (GANs) [10], [13], [5]. As many services based on GANs train them with sensitive information (e.g., medical information [7], [26]), membership inference attacks on these services may infer whether a particular individual has a sensitive disease, leading to a serious privacy breach.

Prior membership inference attack against the GANs [10] leverages the components of the GANs framework. GANs consist of a generator to create fake samples that mimic training data, and a complementary discriminator to distinguish the generated data from real data. Therefore, when the adversary trains GANs to learn samples produced by the target model, the generator acts as a *shadow model*, and the discriminator acts as an *attack model*. However, since this method under-represents certain samples, incurring mode collapse [30], [8], [23], the attack becomes ineffective in the black-box setting where auxiliary information is not given to the adversary. The reason for the under-representation problem of the previous attack [10] is that the discriminator is trained faster than the generator. The discriminator of GANs is then quickly able to distinguish between real and produced samples, providing no valid gradients to improve the generator afterward. Consequently, this early convergence of the discriminator causes a decrease in the quality and diversity of samples produced by the generator.

We propose a novel attack framework called *VoteGAN* to prevent early convergence, which is a fundamental problem in the prior attack method [10]. VoteGAN consists of multiple discriminators and one generator. As the discriminators are trained separately with a partition of the training data, each discriminator is provided with restricted views of the training data. Meanwhile, the generator of VoteGAN is trained with the entire training data. As the discriminative task of each discriminator becomes more difficult, valid gradients continue to be provided throughout the training, thus early convergence of discriminator can be prevented. Consequently, VoteGAN can stably learn the distribution of each data partition, and efficiently approximate the entire distribution of all samples generated by the target model. Our experiments demonstrate that VoteGAN can perform metric-independent attacks in black-box settings and exhibits a lower generalization gap on the datasets used in the experiments, as well as achieving a higher attack success rate than prior method, called LOGAN [10].

Our study makes the following contributions:

- We conduct an in-depth analysis of the fundamental challenges of membership inference attacks on GANs in the black-box setting. We then propose a novel attack framework called VoteGAN, aiming to overcome the fundamental limitations of the previous work [10] by utilizing multiple discriminators to depict the target model more generally and accurately.
- We evaluate the performance of VoteGAN on various generative models trained with two benchmark datasets (CIFAR-10 [16], MNIST [18]) and one

real-world dataset (LFW [14]); and show that the attack success rate has increased by an average of 25% compared to the prior method.
– We demonstrate that VoteGAN can effectively attack target models trained with up to 4,000 data points; while prior method is only effective against target models trained with approximately up to 2,000 data points.

2 Background

2.1 Membership Inference Attack

The membership inference attack is a binary classification task that aims to infer whether specific data was used to train the target model. The adversary trains an attack model $\mathcal{A}$ that takes a data sample as an input. The attack model $\mathcal{A}$ outputs 1 if the adversary infers that a data sample s is a member of the target training set; otherwise, it outputs 0.

To achieve membership inference, the adversary exploits the overfitting behavior of the deep learning model by training data, where the model cannot perform accurately against unseen data while only fitting exactly against its training data. The overfitting is closely related to the size of the training dataset. Prior works [29], [33], [4], [28], [15], [25], [32], [19], [10], [5], [13] have demonstrated there is a correlation between attack performance and training dataset size. The dataset size per class used for the model training and the level of overfitting are inversely proportional [29], [33], [4]. This is because the smaller training data would less reflect the features of the population. Membership inference attacks on classification tasks become more accurate as the dataset size per class decreases because the adversary leverages overfitting behavior [29], [28], [15], [25], [32], [19]. Although unsupervised learning models such as generative models do not learn class labels, unlike supervised learning, they are also affected by the overall size of the dataset used for model training. As a result, the smaller the training dataset is, the more vulnerable it is to membership inference attacks [10], [5], [13].

2.2 Generative Adversarial Networks (GANs)

A Generative Adversarial Networks (GANs) [8] is a generative model that consists of a generator (G) and a discriminator (D). The generator takes random noise (latent code z) as an input and produces samples that approximate the training data distribution p_{data}, aiming to produce realistic data by following the entire distribution of training. The discriminator is a binary classifier that labels $D \rightarrow [0, 1]$: if the input data is predicted to be real data, the output will be close to 1; otherwise (i.e., predicted to be created one), it will be close to 0. During the training of GANs, the generator and the discriminator play a two-player minimax game that has global optimality [8]. D and G are optimized according

to the value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log (1 - D(G(z)))], \tag{1}$$

where $G(z)$ denotes a fake sample produced by the generator from z, and x is the real data of the training set. By competing with the discriminator in the training procedure, the generator learns how to produce a more realistic output to deceive the discriminator.

Equation 2 is the loss function of GANs generator. The generator attempts to minimize the loss L_G:

$$L_G = \mathbb{E}_{z \sim p_z(z)}[\log (1 - D(G(z)))]. \tag{2}$$

The discriminator learns to distinguish between the real data and the produced sample. Equation 3 is the loss function of GANs discriminator. The discriminator attempts to maximize the loss L_D:

$$L_D = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log (1 - D(G(z)))]. \tag{3}$$

The Deep Convolutional GAN (DCGAN) [27] learns training data more stably than the GAN [8] by leveraging the Convolution Neural Network (CNN) structure. Because of this stability, it becomes the basis for many of the other models. Therefore, in this study, we also selected DCGAN as the basic target model.

The Wasserstein GAN (WGAN) [1] is designed with a new loss function to address the problem of instability in the GAN [8] training. Equation 1 is the value function of GAN [8] with *Jenson-Shanon* (JS) divergence [21], which has a disadvantage of gradient vanishment in the GANs training process. The vanishing gradient problem is a phenomenon in which one model delivers a gradient close to zero to another model. As a result, the training is terminated incompletely. The Wasserstein distance [31] with WGAN addresses this problem because it always carries non-zero gradients.The Wasserstein distance loss of WGAN addresses this problem because it always carries non-zero gradients.

PrivGAN [24] was proposed to defend against data leakage from membership inference attacks by an overfitting mitigation strategy in the black-box setting. In order to mitigate the overfitting problem, PrivGAN allows one out of its multiple generators to impose a penalty on the discriminator (specifically, by fooling the discriminator into not memorizing the training data). This mechanism controls the trade-off between privacy and utilities with λ hyper parameters instead of making a solid defense (the larger the λ value is, the less utility it has). We use PrivGAN as a target model to demonstrate the resistance of our proposed attack to the defense mechanism.

2.3 Generalization Gap

In deep learning, the generalization gap means the difference between the model's prediction distributions on training data and non-training data derived from the

same distribution [6]. This paper uses the generalization gap to measure how well the *shadow model* of the attack model represents the target model. To measure the generalization gap, we use the discriminator of the attack model because samples produced by the target model are fed into only the discriminator. The formula for the generalization gap is defined as follows:

$$G_{gap} = \mathbb{E}_{x \sim p_{\text{training}}}[\log D(x)] - \mathbb{E}_{x \sim p_{\text{ non-training}}}[\log D(x)]. \tag{4}$$

The large generalization gap of the attack model means that the attack model under-represents certain data samples produced by the target model, which decreases the attack performance because of mode dropping [23] and mode collapse [30], [8].

In the black-box setting, resolving such under-representation of the attack model is the main challenge in membership inference attacks against generative models. Therefore, we will handle this issue by designing a novel attack model in the next section.

3 VoteGAN

3.1 Attack Overview

The attack flow of VoteGAN is shown in Figure 1. It consists of five steps: data collection, data partitioning, model training, final prediction, and membership inference. In the data collection step, the samples are collected only from responses of the target model generators. Let X_{real} be the dataset produced by the target model. In the data partitioning step, data is pre-processed such that X_{real} is divided into k disjoint partitions $X^1_{real}, X^2_{real}, \cdots, X^k_{real}$ to train multiple k discriminators. In the model training procedure, for any given l, each discriminator D^l learns distribution of partition X^l_{real}. In the final prediction step, the final value of multiple k discriminator models is calculated by soft voting strategy (detailed in Section 3.2).

Figure 2 shows the membership inference step of VoteGAN. With the same assumption as baseline model [10], the adversary knows the size n of the target model's training dataset, and also has a dataset $\{x_1, x_2, \cdots, x_{n+m}\}$ which is suspected of containing data points used to train the target model, where m is the number of datasets not used for training. The adversary inputs $n+m$ samples into VoteGAN, and sorts them in a descending order to facilitate the inference of the adversary that data showing higher prediction values were more likely to be used for training the target model. Thus, the adversary infers data showing the top n predictions as the data actually used to train the target model.

3.2 Voting Strategy

Voting has two methods: soft voting and hard voting. Soft voting adds prediction values from all models and then averages them. Therefore, its final prediction value lies between [0, 1]. On the contrary, hard voting decides it by the majority

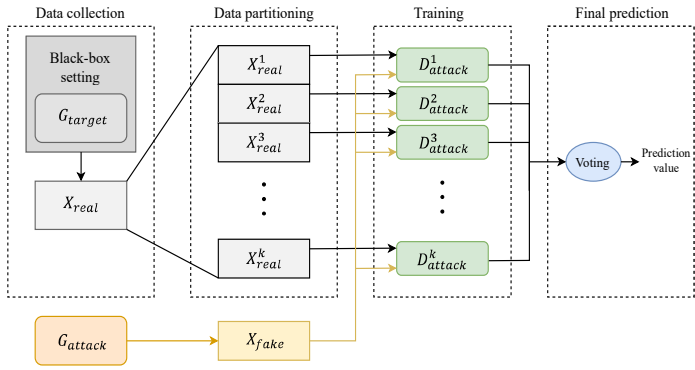


Fig. 1. Overview of VoteGAN framework.

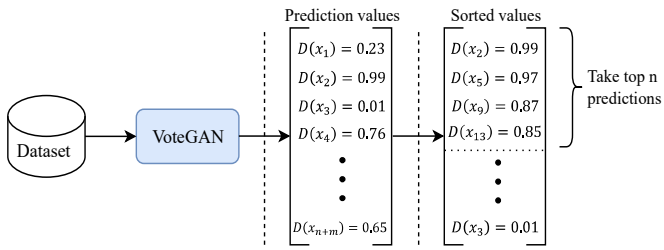


Fig. 2. Membership inference method of VoteGAN.

vote. That is, the final decision would be 1 if the majority of models predicting the same output exceed a given threshold for the predicted value of input x; otherwise, 0.

Our membership inference attack adopts the soft voting, because our model exploits subtle distinctions in the predictions for training and non-training data, which can hardly be captured by the hard voting. The soft voting of VoteGAN is defined as Equation 5.

$$Y_{voting}(x) = \frac{1}{k}\sum_{i=1}^{k} D^i(x). \tag{5}$$

Each discriminator D^i provides a prediction value indicating whether a data record x belongs to the samples produced by the target model. Based on them, the average of all of the prediction values are calculated as the final prediction value, Y_{voting}.

4 Evaluation

4.1 Experimental Setup

Datasets: For the evaluation of its design effectiveness, two toy datasets [20], [6] were used: the Blobs and Circles. The Blobs dataset was created using a two-dimensional Gaussian Mixture Model with five centers. The Circles dataset was created to be composed of a large composite circle containing a smaller circle. Each of the Blobs and Circles datasets consists of 5,000 samples.

For the evaluation of the other properties, we used a real-world dataset (LFW [14] of human faces) as well as benchmark datasets (CIFAR-10 [16] and MNIST [18]). LFW dataset contains 13,233 human face images collected from the web. CIFAR-10 is a collection of colored (RGB) objects dataset. It comprises 32×32 pixels with 60,000 samples (50,000 training samples, 10,000 test samples). MNIST is a grayscale handwritten digits dataset consisting of 28x28 pixels. There are ten classes from 0 to 9, and the total sample size is 70,000 (60,000 training samples, 10,000 test samples).

Attack Evaluation Metric: Since the membership inference attack is a binary classification task, we adopt the Area Under the Curve (AUC) metric to evaluate the performance of the binary classifier [22], [3]. AUC is a metric to measure an attack's average success rate. The AUC has a value between 0.5 and 1 (the higher the AUC, the higher the attack's average success rate). The AUC for random guesses is 0.5. To evaluate the AUC of attacks, we assume the adversary has a dataset containing the whole data used for training the target model as well as the non-training data of the same size.

Target Models: To evaluate the performance of VoteGAN in diverse aspects, we chose DCGAN (model with a typical CNN structure) [27], WGAN (model that uses different losses) [9], and PrivGAN (model with a defense against inference attacks) [24] as the target model. We trained DCGAN and PrivGAN for 400 epochs with the Adam optimizer [17], and set the parameter λ that controls the balance between privacy and utilities of PrivGAN to 1. We trained WGAN for 10,000 epochs using the RMSprop optimizer [9]. To evaluate VoteGAN's resistance to attack defense, parameter values of PrivGAN were applied with $\lambda = 0.1, 1, 10$, respectively (the larger λ value implies the stronger attack defense). We trained the target model with 2,000 data samples from the LFW, MNIST, and CIFAR-10 datasets.

4.2 Evaluation of Multiple Discriminators

To evaluate the effectiveness of multiple discriminators, we conduct both qualitative and quantitative analyses utilizing the Blobs and Circles datasets. The qualitative analysis evaluated the improvement of the GANs representation ability using visualizing the produced samples as the number of discriminator k increases. The quantitative analysis was evaluated using the Frechet Inception Distance (FID) [12] score. It is a metric to evaluate the quality of the produced samples by measuring the feature distance between the actual training image and

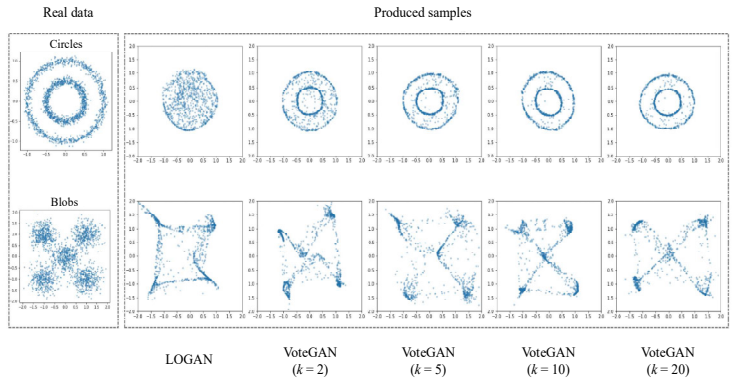


Fig. 3. Comparison of distributions produced by LOGAN and VoteGANs. The first column shows the original distributions of the Blobs and Circles. The second column shows the distributions produced by LOGAN. The third to last columns show the distributions produced by VoteGANs.

Table 1. Comparison of the FID scores.

	Blobs	Circles
LOGAN [10]	0.387	0.076
VoteGAN ($k = 2$)	0.224	0.038
VoteGAN ($k = 5$)	0.202	0.021
VoteGAN ($k = 10$)	0.158	**0.016**
VoteGAN ($k = 20$)	**0.134**	0.023

the produced sample. A lower FID score indicates that real data and produced samples have similar distributions.

Figure 3 depicts the real distributions of the Blobs and Circles datasets and the distributions produced by LOGAN and VoteGAN that learned them. As shown in the figure, LOGAN [10] cannot learn clear distributions on either the Blobs and Circles datasets. In Table 1, VoteGAN achieves better performance in terms of FID scores for all k. These results demonstrate that GANs with multiple discriminators can generate more similar data samples with the training data than GANs with a single discriminator.

4.3 Evaluation of Attack Performance

Attack Success Rate on Various Target Models Table 2 shows the attack success rate in terms of AUC score on various target models and datasets. We observed that VoteGAN outperforms LOGAN regardless of the dataset, showing 25% higher AUC scores on average. We also observed that VoteGAN shows the best performance on most of the datasets when $k = 5$ or $k = 10$ rather than $k = 20$.

Table 2. Attack success rate of membership inference attacks on various target models and datasets. Datasets I, II, III are LFW, MNIST and CIFAR-10, respectively.

Target \ Models / Datasets	DCGAN I	DCGAN II	DCGAN III	WGAN I	WGAN II	WGAN III	PrivGAN ($\lambda = 1$) I	PrivGAN ($\lambda = 1$) II	PrivGAN ($\lambda = 1$) III
LOGAN [10]	0.661	0.686	0.713	0.677	0.616	0.705	0.603	0.622	0.614
VoteGAN ($k = 2$)	0.757	0.743	0.806	0.740	0.701	0.784	0.702	0.772	0.754
VoteGAN ($k = 5$)	0.779	0.787	0.755	0.753	**0.825**	0.803	0.710	**0.794**	0.757
VoteGAN ($k = 10$)	**0.818**	**0.807**	0.797	**0.801**	0.731	**0.868**	**0.756**	0.739	**0.807**
VoteGAN ($k = 20$)	0.709	0.815	**0.832**	0.706	0.626	0.826	0.708	0.791	0.798

Attack Success Rate with Different Training Dataset Size To evaluate the attack success rate according to the different sizes of the training dataset, we trained the target model by varying the training data size. In this experiment, DCGAN was selected as the target model, and $k = 10$ in VoteGAN. Figure 4 shows the comparison result of attack success rates between LOGAN and VoteGAN for each dataset.

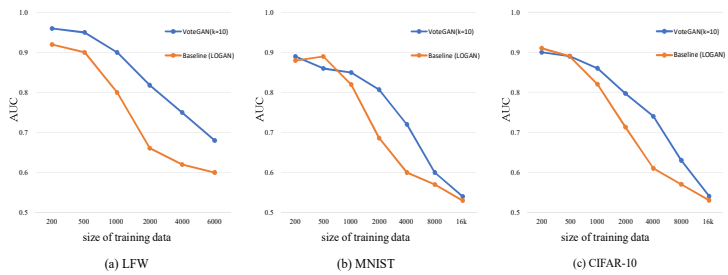


Fig. 4. Attack success rate against DCGAN with different training dataset size.

As explained in Section 2.1, we can observe that the smaller training data size makes the model more vulnerable to both attacks on every dataset. In addition, on every dataset, VoteGAN can effectively attack trained models with up to 4,000 samples (AUC score above 0.7); while LOGAN is only effective with approximately up to 2,000 samples.

Resistance to Defense Mechanism To evaluate the resistance capability of LOGAN and VoteGAN to a defense mechanism, we utilize PrivGAN [24] trained with CIFAR-10, because PrivGAN is designed to be resilient against the membership inference attack by mitigating the overfitting. As described in Section 2.2, the parameter λ of PrivGAN controls the trade-off between privacy and utilities (the larger λ value implies the stronger privacy protection).

Table 3 shows the attack success rate of LOGAN and VoteGAN against PrivGAN. We observe that VoteGAN outperforms LOGAN for every λ value.

Table 3. Attack success rate against PrivGAN with defense mechanisms

	PrivGAN ($\lambda = 0.1$)	PrivGAN ($\lambda = 1$)	PrivGAN ($\lambda = 10$)
LOGAN [10]	0.680	0.614	0.549
VoteGAN ($k = 2$)	0.748	0.754	0.688
VoteGAN ($k = 5$)	0.752	0.757	**0.723**
VoteGAN ($k = 10$)	**0.845**	**0.807**	0.698
VoteGAN ($k = 20$)	0.806	0.789	0.716

In addition, as λ increases, the AUC score of LOGAN approaches 0.5, which indicates that LOGAN cannot successfully infer membership. On the other hand, VoteGAN consistently shows the AUC scores above 0.7 in most of the cases, despite the increase of λ. The experiment results show that VoteGAN can consistently perform effective membership inference attacks even when the defense mechanism is adopted in the model, as opposed to LOGAN.

5 Conclusion

In this work, we propose a novel membership inference attack called *VoteGAN* against generative models. It leverages multiple discriminators to accurately represent the target model. Specifically, VoteGAN trains each discriminator separately with disjoint partitions of the training data, enabling the accurate approximation of the whole distribution of the training data. Thus, VoteGAN can effectively represent the target model in the black-box setting. The experimental results with real-world and benchmark datasets on various target models demonstrate that VoteGAN shows higher performance both in the attack success rate and the resistance to defense mechanisms, compared to the state-of-the-art baseline model. Considering our attack framework can be generically applied to diverse generative models and datasets in the wild, our attacks may pose realistic threat in the real world. Thus, how to effectively mitigate our attacks is an interesting problem, which we will attempt to solve in the future work.

Acknowledgments. This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00411, IITP-2023-2021-0-01810), and Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education(NRF-2021R1A6A1A13044830).

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. Bauer, L.A., Bindschaedler, V.: Towards realistic membership inferences: The case of survey data. In: Annual Computer Security Applications Conference. pp. 116–128 (2020)

3. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1519–1519. IEEE Computer Society (2022)
4. Caruana, R., Lawrence, S., Giles, C.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems **13** (2000)
5. Chen, D., Yu, N., Zhang, Y., Fritz, M.: Gan-leaks: A taxonomy of membership inference attacks against generative models. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. pp. 343–362 (2020)
6. Chen, J., Wang, W.H., Gao, H., Shi, X.: Par-gan: Improving the generalization of generative adversarial networks against membership inference attacks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 127–137 (2021)
7. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. In: Machine learning for healthcare conference. pp. 286–305. PMLR (2017)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)
10. Hayes, J., Melis, L., Danezis, G., De Cristofaro, E.: Logan: Membership inference attacks against generative models. Proceedings on Privacy Enhancing Technologies **2019**(1), 133–152 (2019)
11. He, Z., Zhang, T., Lee, R.B.: Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 148–162 (2019)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Hilprecht, B., Härterich, M., Bernau, D.: Monte carlo and reconstruction membership inference attacks against generative models. Proc. Priv. Enhancing Technol. **2019**(4), 232–249 (2019)
14. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Workshop on faces in'Real-Life'Images: detection, alignment, and recognition (2008)
15. Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N.Z., Cao, Y.: Practical blind membership inference attack via differential comparisons. In: ISOC Network and Distributed System Security Symposium (NDSS) (2021)
16. Kai, Y., Yuanqing, L., Lafferty, J.: Learning image representations from the pixel level via hierarchical sparse coding. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 1713–1720 (2011)
17. Kingma, D., Ba, L., et al.: Adam: A method for stochastic optimization (2015)
18. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
19. Leino, K., Fredrikson, M.: Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In: 29th USENIX security symposium (USENIX Security 20). pp. 1605–1622 (2020)

20. Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R., Carin, L.: Alice: Towards understanding adversarial learning for joint distribution matching. Advances in neural information processing systems **30** (2017)
21. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information theory **37**(1), 145–151 (1991)
22. Ling, C.X., Huang, J., Zhang, H., et al.: Auc: a statistically consistent and more discriminating measure than accuracy. In: Ijcai. vol. 3, pp. 519–524 (2003)
23. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
24. Mukherjee, S., Xu, Y., Trivedi, A., Patowary, N., Ferres, J.L.: privgan: Protecting gans from membership inference attacks at low cost to utility. Proc. Priv. Enhancing Technol. **2021**(3), 142–163 (2021)
25. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 739–753. IEEE (2019)
26. Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., Shen, D.: Medical image synthesis with context-aware generative adversarial networks. In: International conference on medical image computing and computer-assisted intervention. pp. 417–425. Springer (2017)
27. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
28. Salem, A., Zhang, Y., Humbert, M., Fritz, M., Backes, M.: Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In: Network and Distributed Systems Security Symposium 2019. Internet Society (2019)
29. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
30. Thanh-Tung, H., Tran, T.: Catastrophic forgetting and mode collapse in gans. In: 2020 international joint conference on neural networks (ijcnn). pp. 1–10. IEEE (2020)
31. Vallender, S.: Calculation of the wasserstein distance between probability distributions on the line. Theory of Probability & Its Applications **18**(4), 784–786 (1974)
32. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). pp. 268–282. IEEE (2018)
33. Ying, X.: An overview of overfitting and its solutions. In: Journal of physics: Conference series. vol. 1168, p. 022022. IOP Publishing (2019)

Unsupervised Contextual Anomalous Communication Detection Using VQ Tokenization with Flow Data

Norihiro Okui[1], Shotaro Fukushima[2], Ayumu Kubota[1], and Takuya Yoshida[3]

[1] KDDI Research, Inc., Saitama, Japan {no-okui, ay-kubota}@kddi.com
[2] ARISE analytics inc., Tokyo, Japan. shotaro.fukushima@ariseanalytics.com
[3] Toyota Motor Corporation, Tokyo, Japan ta-yoshida@mail.toyota.co.jp

Abstract. The widespread use of IoT devices has increased the importance of IoT security, and research on anomaly detection has attracted attention due to the properties of IoT devices, such as low resources. Flow data-based methods such as IPFIX, have attracted attention for the purpose of reducing analysis data from scalability perspective, but improving detection accuracy remains challenging. Recently, anomaly detection methods based on natural language processing (NLP) techniques have been proposed, and improved accuracy has been reported by considering the flow data context. In the application of NLP to flow data containing quantitative variables, it is necessary to construct a vocabulary by tokenization, which is different from the NLP. In this study, we propose a novel anomaly detection method combining a Vector Quantized Variational AutoEncoder (VQ-VAE) and a Transformer. This method improves the accuracy of anomaly detection by simultaneously learning the VQ-VAE and Transformer to achieve aggregated tokenization between flows with similar statistical and sequential characteristics. Experimental results on the ToN-IoT dataset showed that the proposed method achieves higher accuracy than methods in previous studies.

Keywords: Anomalous communication detection · IoT · IPFIX · Natural language processing · Vector quantization

1 Introduction

The number of IoT devices is rapidly increasing, with 15.1 billion active connections in 2023, and this number is projected to rise to 29.4 billion by 2030 [11]. These devices often have limited processing power and rely on battery life, posing challenges in enhancing processing capabilities or implementing direct security measures. Monitoring the data transmitted and received by IoT devices to detect unusual communication patterns is a practical security approach.

Communication data are primarily available in two formats: packet and flow data [15]. Packet data, which are rich in information, are relatively easy to manage, with numerous analytical methods developed for their analysis. However,

scalability becomes a challenge as the volume of data and the number of devices increase. Compare with packet data, flow data, which aggregate packet data into session statistics, significantly reduce data volume. This efficiency has led to increased research into detecting anomalous communications using flow data. Although flow data contain fewer characteristics than packet data, potentially affecting detection accuracy, researchers have developed advanced anomaly detection techniques. These techniques extend beyond traditional statistical analysis, incorporating methods that utilize temporal features [19] and approaches that treat flow data sequences similarly to natural language, achieving reasonable detection accuracy [5,14]. However, existing methods that treat flow records as words face challenges related to the tokenization process and the handling of unknown flows.

Tokenization Methods In NLP, tokenization divides text into meaningful units such as words or subwords [17], leveraging grammatical rules that are ideal for learning sequences. Applying tokenization to IoT communication flow data seems intuitive, as it organizes records by service units, such as DNS communication or home IoT device interactions such as turning lights on and off. For some services, such as DNS communication, tokenization can be effectively achieved using only the information in the IP and TCP headers, such as the protocol and port number. However, challenges arise when multiple service types coexist within the same port number, such as turning lights on and off, necessitating the consideration of quantitative variables such as packet numbers and sizes. Previous studies have incorporated these quantitative variables into tokenization, but the methods are not fully optimized for the flow data characteristics, often relying on rule-based thresholds and Q-quantile points.

Handling of Unknown Flows In NLP, infrequently occurring or newly generated words are often categorized as unknown tokens [6]. In IoT communication, it is anticipated that flow data characterized by infrequent packet numbers and sizes may emerge during a cyber-attack, highlighting the importance of effective anomaly detection. Therefore, addressing these infrequent flow data appropriately is crucial for identifying anomalous communications. However, previous studies have typically handled unknown tokens by grouping them with the closest statistical tokens, which can result in overlooking anomalous communications.

This study focuses on methods that consider the discrete sequence of flow data occurrences to detect anomalies in IoT communications. The primary goal of our study is to automatically optimize the tokenization of flow data features to enhance the accuracy of anomaly detection. The contributions of this study are outlined as follows:

- We have developed a novel unsupervised method for detecting anomalous communications that integrates a Vector Quantized-Variational AutoEncoder (VQ-VAE) [21] with a Transformer [22]. This method is designed to learn tokenization and analyze both the statistical and sequential features in an end-to-end manner.

– Our approach also enables the effective tokenization of unknown flows, allowing for the isolation and detailed analysis of flow data with anomalous statistics within each token.
– We evaluated our proposed method using the ToN-IoT dataset, an open dataset for IoT communication, and confirmed that it consistently achieves high performance.

2 Related Works

The advancement of deep learning in NLP has led to numerous anomaly detection methods in the security domain, particularly using textual data such as system logs and HTTP request information, which are well-suited for NLP techniques [7, 8].

Recently, anomaly communication detection methods have been proposed that treat each record as a grammatical unit in terms of word and sequence, applying NLP techniques [14, 23]. Unsupervised methods are particularly crucial in anomaly communication detection for addressing unknown attacks. These NLP-based methods typically generate an anomaly score through autoregressive learning, predicting the next record in a sequence of flow data. Anomaly detection techniques can be categorized based on the type of data they use, for example continuous values such as statistics [23] and discrete values such as port numbers and tokenized packet sizes [5, 14]. Given that flow data, such as IPFIX format data, are fundamentally discrete and records are created for each service type, using discrete values for NLP inputs and targets is appropriate.

Methods by Radford et al. [14] and Clausen et al. [5] focus on anomalous communication detection considering discrete sequence relations. These methods use $log2(bytes)$ and Q quantile points to discretize quantitative variables, but they are not optimized for the characteristics of the flow data and do not achieve service-specific tokenization.

Clustering methods such as K-Means have been used for service estimation tasks involving flow data [3], and a tokenization method using K-means has been suggested [4]. However, this method is primarily aimed at estimating the model of the IoT device, not at detecting anomalous communications. Furthermore, as tokenization aggregates to the nearest token, it fails to detect statistically anomalous communications that occur during a cyber-attack, significantly impacting the accuracy of anomaly detection.

3 Proposed Method

This section outlines the fundamental concept of the proposed method and the structure of the anomaly detection process.

3.1 Fundamental Concept

To enhance the accuracy of anomaly communication detection using flow data, it is crucial to accurately model the state of normal communication. Our method

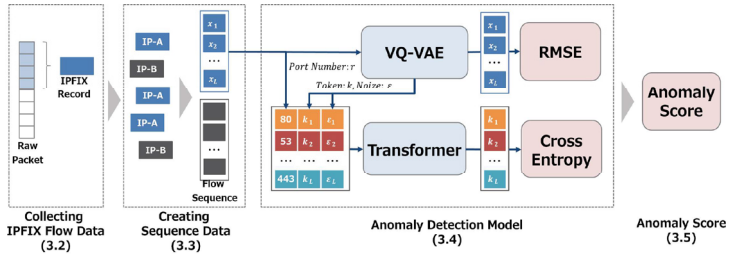


Fig. 1. Proposed anomaly detection processing

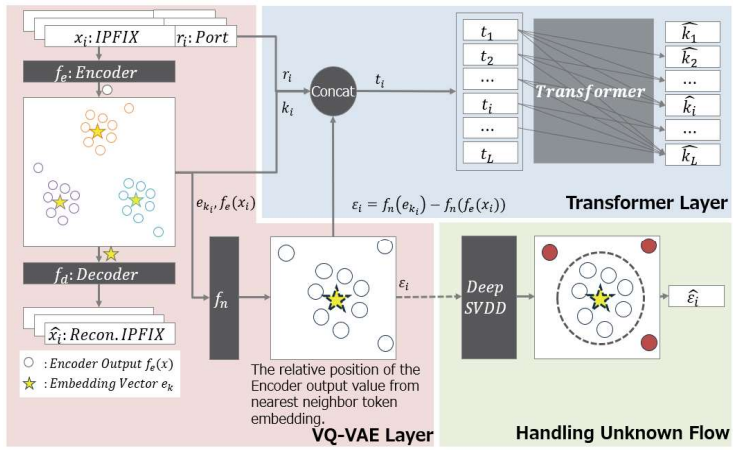


Fig. 2. Proposed model architecture

focuses on both the statistical and sequential features of the flow data for modeling. Initially, we considered hypotheses about the statistical features. The flow data includes quantitative variables such as packet numbers and communication volumes. However, since records are generated for specific service units such as "DNS communication" or "turning lights on and off", they inherently possess a discrete nature. In addition, even for the same service, there are continuous variations due to parameter changes such as retransmissions (hereinafter, referred to as "noise"). We hypothesize that flow data comprise both discrete features (services) and continuous features (parameters and other variations). Next, we address hypotheses concerning sequential features. Existing methods [5,14] suggest that recognizing discrete-sequence regularities in flow data under normal conditions can enhance the detection accuracy of anomalous communications.

Given that cyber-attacks often exploit vulnerabilities by deviating from expected service patterns, parameters, and ordering rules, our method uses a VQ-VAE to estimate service and noise such as parameters, and employs a Trans-

former to learn the sequence rules based on the obtained service features. Unlike traditional methods that separate discretization and sequential learning, our approach integrates these processes as suggested in [18], allowing simultaneous consideration of both the statistical and sequential features. This integration is expected to cluster flow data with similar features into the same token, thereby improving the anomaly detection accuracy.

Our methodology involves three primary steps: collecting IPFIX flow data, creating sequence data from the flow data, and detecting anomalies using our method. Fig. 1 provides a visual overview of our approach.

3.2 Collecting IPFIX Flow Data

While both packet data and flow data are prevalent in anomaly communication detection, our method specifically utilizes IPFIX, a type of flow data, due to its scalability advantages. Network devices such as routers and switches are capable of generating flow data, and packet data can be converted into IPFIX format using tools such as YAF [2, 20]. We define the statistics of the ith flow record with source IP address $p \in P$ as $x_{p,i}$ and port number as $r_{p,i}$.

3.3 Creating Sequence Data

Assuming that flow records with the same source IP address have an ordering regularity, we create a fixed-length L sequence $\boldsymbol{s_{p,i}} \in \mathcal{S}$ defined by (1).

$$\boldsymbol{s_{p,i}} = (x_{p,max(1,i-L+1)}, x_{p,max(1,i-L+1)+1}, ..., x_{p,i}) \tag{1}$$

For simplicity, let $\boldsymbol{s} = (x_1, ..., x_L) \in \mathcal{S}$.

3.4 Anomaly Detection Model

We detect anomalous communication using our newly proposed model, which consists of two layers: a VQ-VAE layer and a Transformer layer. Fig. 2 illustrates the model's architecture.

VQ-VAE layer The VQ-VAE [21] is an encoder-decoder model featuring a discrete latent space. In this layer, the sequence of flow records $\boldsymbol{s} = (x_1, ..., x_L) \in \mathcal{S}$ is used to pseudo estimate services and noise.

Initially, the encoder function f_e converts each $x \in \boldsymbol{s}$ into $z = f_e(x) \in \mathbb{R}^D$. Subsequently, z is quantized according to the embedding space $E_{vq} = \{e_k \in \mathbb{R}^D | k \in 0, ..., K-1\}$ as defined in (2), where K represents the number of embeddings and the dimensionality of the embedding vector. The resulting e_k from (2) is then reconstructed into the original flow data by the decoder function f_d.

$$\begin{aligned} f_k(z) &= argmin_k \ \|z - e_k\|_2 \\ f_q(z) &= e_k \ where \ k = f_k(z) \end{aligned} \tag{2}$$

This results in a sequence of pseudo service token $\boldsymbol{k} = (k_1, ..., k_L)$ in (2).

At the same time, noize ϵ are estimated. The data was defined by (3) using the linear function f_n.

$$\epsilon = f_n(f_e(x)) - f_n(e_k) \tag{3}$$

We define $\mathcal{L}_v$ as the loss function of the VQ-VAE layer. The $\mathcal{L}_v$ is an extension of the VQ-VAE [21] loss function to sequences and is defined as in (4). where sg is an identifier indicating a gradient stop and β is a control parameter.

$$\mathcal{L}_v = \sum_{i=1}^{L} \left(\|x_i - f_d(e_{k_i})\|_2^2 + \|sg[f_e(x_i)] - e_{k_i}\|_2^2 + \beta \|f_e(x_i) - sg[e_{k_i}]\|_2^2 + \|\epsilon_i\|_2^2 \right) \tag{4}$$

Transformer layer The Transformer [22] utilizes a self-attention mechanism and has been effectively applied to various types of series data, including natural language, images, audio, and time series. This layer employs an encoder-decoder structure to learn the discrete sequence of flow data.

The service and parameters (e.g. GET parameters) up to the last service determine the sequence in which services are generated. Therefore, the proposed method uses the pseudo service k_i, noize ϵ_i estimated at the VQ-VAE layer and port number r_i as input features to the Transformer. The sequence of flow data $\boldsymbol{t}$ to be input to the Transformer is defined by (5).

$$\begin{aligned} E_k &= \{e_j^{(k)} | j = 0, 1, 2, ..., K-1\} \\ E_r &= \{e_j^{(r)} | j = 0, 1, 2, ..., R-1\} \\ t_i &= Concat(e_{k_i}^{(k)}, e_{r_i}^{(r)}, \epsilon_i) \\ \boldsymbol{t} &= \{t_1, ..., t_L\} \end{aligned} \tag{5}$$

Using the context vector $c_{\boldsymbol{t}}$ obtained by inputting $\boldsymbol{t}$ into the Transformer Encoder and the Transformer Decoder, the probability of the next token occurring $\hat{k}_i = P(k_i | t_{i<}, c_{\boldsymbol{t}})$ is estimated. For the loss function $\mathcal{L}_t$, we use CrossEntropyLoss for the objective variable k_i and its predicted probability $\hat{k}_i$ in (6).

$$\mathcal{L}_t = CrossEntropyLoss(\boldsymbol{k}, \hat{\boldsymbol{k}}) \tag{6}$$

Loss Function The overall loss function of the proposed method is defined by (7), where γ_1 and γ_2 are the weights of each loss function.

$$\mathcal{L} = \gamma_1 \mathcal{L}_v + \gamma_2 \mathcal{L}_t \tag{7}$$

Handling Unknown Flow When a communication involving an unknown service or anomalous parameters occurs, the encoder output value z may significantly deviate from all $e_k \in E_{vq}$. The quantization process defined in (2), is necessarily assigned to one of the existing tokens, which can lead to missed

anomalies. To address this issue, flows exhibiting anomalous features within each token are identified and flagged as anomalies.

To integrate the anomaly score with the predicted probability $\hat{k}$ obtained from the Transformer layer, the anomaly score is converted into a probabilistic measure using DeepSVDD [16]. Anomaly detection for noise ϵ is conducted using DeepSVDD, and the resulting anomaly score ϵ' is converted into a probabilistic value using (8) proposed in [10], where ERF represents the Gaussian error function.

$$\hat{\epsilon} = max\left\{ERF\left(\frac{\epsilon' - mean(\epsilon')}{std(\epsilon') * \sqrt{2}}\right), 0\right\} \tag{8}$$

3.5 Anomaly Score

The anomaly score $\mathcal{AS}$ of the flow sequence $\boldsymbol{s} \in \mathcal{S}$ is defined as (9).

$$\mathcal{AS} = 1 - \hat{k}_L * (1 - \hat{\epsilon}_L) \tag{9}$$

4 Experiments

This section outlines the dataset utilized in the anomaly communication detection experiments, detailing the conditions and results.

4.1 Dataset

For our experiments, we employed the ToN-IoT dataset [13], a widely recognized open dataset for IoT communication. This dataset comprises communication data collected over approximately one month, from 02/04/2019 to 29/04/2019, from a network with seven different IoT/IIoT services, including weather sensors, light bulb sensors, smartphones, and smart TVs. The ToN-IoT dataset includes data from various cyber-attacks such as DoS and XSS, with communications generated during these attacks labeled according to the type of attack. In this experiment, all pcap data were converted to flow data in IPFIX format using Yaf [9], with an active timeout set to 1800 seconds and an idle timeout set to 300 seconds.

To preserve temporal continuity and avoid data leakage, the dataset was divided based on time. The training data consisted of 327,256 records of normal communication from the start of the dataset until 10:00 am on 24/04/2019. The remaining 14,486,152 records were used as evaluation data. This approach ensures that the training and evaluation data are temporally sequential, which is crucial for time-series data to prevent potential leakage. A summary of the training and evaluation data is presented in Table 1.

Table 1. Number of records of training and evaluation data in the ToN-IoT

Attack Type	Training data	Evaluation data
Benign	327,256	1,346,953
Anomaly	0	13,139,199

Table 2. Correlations between features and IPFIX elements

Feature	IPFIX element
Inbound packets	packetTotalCount
Outbound packets	reversePacketTotalCount
Inbound traffic volume	octetTotalCount
Outbound traffic volume	reverseOctetTotalCount
Duration	flowDurationMilliseconds
Average inbound packet size	$\frac{octetTotalCount}{packetTotalCount}$
Average outbound packet size	$\frac{reverseOctetTotalCount}{reversePacketTotalCount}$

4.2 Experimental Conditions

This section details the experimental setup used to evaluate the effectiveness of our proposed anomaly detection method by comparing it with existing methods that learn typical anomaly detection techniques and discrete sequences. We compared our method against well-known anomaly detection techniques such as AutoEncoder [1], DeepSVDD [16], VQ-VAE [21], and LSTM-AE [12]. Additionally, we evaluated methods that learn the discrete ordering of flow data, specifically those that handle tokenization and unknown tokens, including methods using Q-quantile points for tokenization (Q-Quant. + Trans.) and K-Means clustering (K-Means+Trans.).

The proposed method was configured with a sequence length $L = 16$, number of embeddings $K = 512$, embedding dimensions $D = 2$, and control parameters $\beta = 0.25$, $\gamma_1 = 1$ and $\gamma_2 = 0.1$. The learning rate was set at 0.003, the batch size was 128, and the model was trained over 3 epochs.

The features input to the model are seven variables: packets (inbound/outbound), traffic volume (inbound/outbound), duration and average packet (inbound/outbound). As a preprocessing step, all the variables are standardized after log-transforming the six variables except for duration. Table 2 shows the correspondence between each variable and the IPFIX elements.

4.3 Experiment Results

Initially, we evaluated the accuracy of our proposed method against traditional anomaly detection methods and those that account for discrete sequential relations. The results are detailed in Table 3. We utilized the PR-AUC and ROC-AUC as the metrics for evaluation. The findings indicate that our proposed method surpasses all other models tested in this study, demonstrating the effectiveness of integrating both statistical and discrete sequential features.

Table 3. Experiment result of ToN-IoT. The table shows the mean (standard deviation) of five seed values 0-4.

	PRAUC	ROCAUC
AutoEncoder	0.805(0.007)	0.147(0.014)
DeepSVDD	0.950(0.023)	0.659(0.171)
VQ-VAE	0.944(0.003)	0.780(0.019)
LSTM-AE	0.905(0.007)	0.589(0.028)
Q-Quant.+Trans.	0.934(0.002)	0.596(0.025)
K-Means+Trans.	0.932(0.012)	0.669(0.091)
VQ-VAE+Trans.(ours)	**0.966(0.013)**	**0.811(0.063)**

Table 4. Results of the ablation experiment for ToN-IoT. The table shows the mean(std) of five seed values of 0-4.

		PRAUC	ROCAUC
VQ-VAE+Trans.	Separately	0.959(0.015)	0.745(0.087)
VQ-VAE+Trans.	Simultaneously(ours)	**0.966(0.013)**	**0.811(0.063)**

Furthermore, we explored the impact of learning discretization and sequential features both separately and simultaneously. For this comparison, identical hyperparameters were maintained for both setups. The outcomes, presented in Table 4, reveal that while both approaches yield high accuracy, the simultaneous learning method exhibits superior anomaly detection capabilities. This enhancement is likely due to the method's ability to aggregate flows with similar statistical and ordinal characteristics into the same token, enhancing the model's predictive accuracy.

5 Discussion

Anomaly communication detection is crucial for identifying various cyber-attacks, each characterized by distinct anomaly patterns. Consequently, the effectiveness of anomaly detection methods varies depending on the type of attack they are designed to identify. This chapter evaluates different anomaly detection methods based on attack type and discusses the strengths and challenges of the proposed methods.

5.1 Detailed Analysis

Table 5 presents the detection rates for each type of abnormal communication when the false-positive rate is fixed at 0.1. Initially, our focus is on five specific attacks: DDoS, Injection, MITM, Password, and XSS. The traditional methods typically address either statistical or discrete ordinal features. In contrast, our proposed method models normal flow data characteristics from both perspectives simultaneously, enhancing its ability to distinguish the abnormal features of these attacks from normal communication patterns.

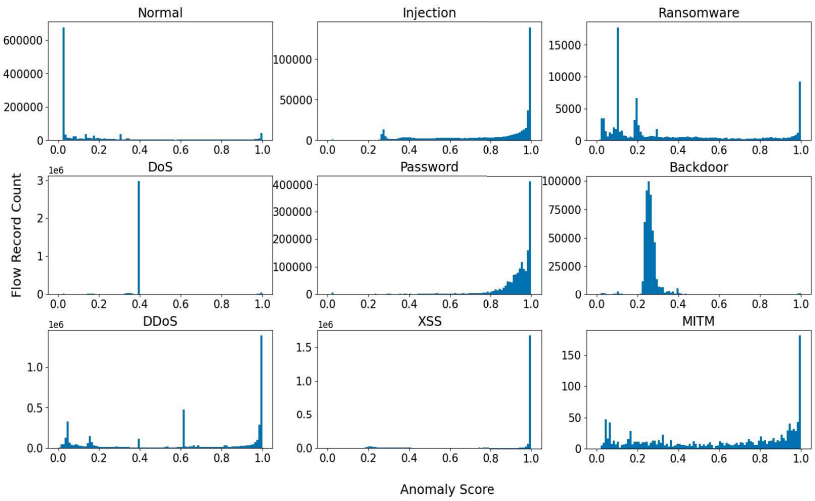


Fig. 3. Anomaly score distribution by proposal model per attack type (SEED=0)

However, both existing and proposed methods show limited accuracy in detecting backdoor and DoS attacks. This limitation stems partly from the methods' inability to incorporate time information and the sequential relationships between different IP addresses. For instance, DoS attacks such as SYN-Flood are characterized by an increased rate of incidents over time, a pattern that our current methods' focus on sequential relations alone fails to capture.

Overdetection is another challenge, as illustrated in Fig. 3, which displays the anomaly score distribution for each attack type within the evaluation data. While our model generally assigns low anomaly scores to normal communications, it erroneously predicts extremely high anomaly scores (=1.0) for certain communications. This issue is likely due to communications that are time-dependent rather than order-dependent, such as NTP communications.

To enhance accuracy, it is essential to develop a mechanism that effectively considers both the temporal characteristics and the order relationships between different IPs. This approach will help in accurately identifying and differentiating between normal communications and various types of cyber-attacks.

6 Conclusion

In this study, we introduced a novel tokenization method that leverages simultaneous learning by integrating VQ-VAE and Transformer. This approach also incorporates a new model that addresses unknown flows within the framework of anomaly communication detection, focusing on the discrete ordering of flow data. We conducted experiments using the ToN-IoT dataset, an open IoT communication dataset, to validate the effectiveness of our proposed method. The

Table 5. Detection rate per type of attack (false-positive rate fixed at 0.1, SEED=0)

	Backdoor	DDoS	DoS	Injection	MITM	Password	Ransom	XSS
AutoEncoder	0.009	0.001	0.021	0.007	0.017	0.000	0.498	0.021
DeepSVDD	0.002	0.503	0.022	0.278	0.319	0.030	0.213	0.605
VQ-VAE	0.009	0.004	0.021	0.047	0.052	0.000	**0.518**	0.060
LSTM-AE	0.007	0.060	0.016	0.018	0.263	0.001	0.241	0.042
Q-Quant.+Trans.	**0.010**	0.066	0.018	0.574	0.393	0.500	0.491	0.519
K-Means+Trans.	0.003	0.087	0.023	0.211	0.102	0.086	0.145	0.118
VQ-VAE+Trans.	0.005	**0.924**	**0.067**	**0.732**	**0.473**	**0.959**	0.235	**0.994**

results confirmed that our method enhances accuracy compared to both the typical existing methods and those utilizing tokenization. The proposed method demonstrated superior accuracy across a broader range of attack types compared to the existing methods, indicating its ability to handle unknown attacks more generally. Moving forward, we plan to develop strategies to incorporate time information more effectively to reduce false-positive.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer (2013). https://doi.org/10.1007/978-1-4614-6396-2
2. Aitken, P., Claise, B., Trammell, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011 (Sep 2013). https://doi.org/10.17487/RFC7011
3. Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.K.R., Sarsour, M.: Network traffic classification: Techniques, datasets, and challenges. Digital Communications and Networks (2022). https://doi.org/10.1016/j.dcan.2022.09.009
4. Bikmukhamedov, R.F., Nadeev, A.F.: Multi-class network traffic generators and classifiers based on neural networks. In: 2021 Systems of Signals Generating and Processing in the Field of on Board Communications. pp. 1–7 (2021). https://doi.org/10.1109/IEEECONF51389.2021.9416067
5. Clausen, H., Grov, G., Aspinall, D.: Cbam: A contextual model for network anomaly detection. Computers **10**(6) (2021). https://doi.org/10.3390/computers10060079
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019)
7. Du, M., Li, F., Zheng, G., Srikumar, V.: Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 1285âĂŞ1298. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3134015
8. Huang, W., Zhu, H., Li, C., Lv, Q., Wang, Y., Yang, H.: Itdbert: Temporal-semantic representation for insider threat detection. In: 2021 IEEE Symposium on Computers and Communications (ISCC). pp. 1–7 (2021). https://doi.org/10.1109/ISCC53001.2021.9631538

9. Inacio, C., Trammell, B.: Yaf: Yet another flowmeter. In: LiSA (2010)
10. Kriegel, H.P., Kroger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: Proceedings of the 2011 SIAM International Conference on Data Mining. pp. 13–24. SIAM (2011)
11. Lionel Sujay Vailshery: Number of iot connected devices worldwide 2019-2023, with forecasts to 2030. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (2023)
12. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016)
13. Moustafa, N.: A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. Sustainable Cities and Society **72**, 102994 (2021). https://doi.org/10.1016/j.scs.2021.102994
14. Radford, B.J., Apolonio, L.M., Trias, A.J., Simpson, J.A.: Network traffic anomaly detection using recurrent neural networks. arXiv preprint arXiv:1803.10769 (2018)
15. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A survey of network-based intrusion detection data sets. Computers Security **86**, 147–167 (2019). https://doi.org/10.1016/j.cose.2019.06.005
16. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International conference on machine learning. pp. 4393–4402. PMLR (2018)
17. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1162
18. Shi, J., Saon, G., Haws, D., Watanabe, S., Kingsbury, B.: VQ-T: RNN Transducers using Vector-Quantized Prediction Network States. In: Proc. Interspeech 2022. pp. 1656–1660 (2022). https://doi.org/10.21437/Interspeech.2022-414
19. Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep recurrent neural network for intrusion detection in sdn-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). pp. 202–206 (2018). https://doi.org/10.1109/NETSOFT.2018.8460090
20. Trammell, B., Boschi, E.: Bidirectional Flow Export Using IP Flow Information Export (IPFIX). RFC 5103 (Jan 2008). https://doi.org/10.17487/RFC5103
21. Van Den Oord, A., Vinyals, O., Others: Neural discrete representation learning. Advances in neural information processing systems **30** (2017), https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017), https://proceedings.neurips.cc/paper/7181-attention-is-all
23. Wakui, T., Kondo, T., Teraoka, F.: Gampal: Anomaly detection for internet backbone traffic by flow prediction with lstm-rnn. In: Machine Learning for Networking. pp. 196–211. Springer International Publishing, Cham (2020)

Boosting Black-Box Transferability of Weak Audio Adversarial Attacks with Random Masking

Mai Bui[0009−0006−5953−6266], Thien-Phuc Doan[0000−0001−7988−5953], Kihun Hong[0000−0002−5538−3630], and Souhwan Jung[0000−0003−2676−3412]

School of Electronic Engineering, Soongsil University, Seoul, Republic of Korea
{maibui,phucdt}@soongsil.ac.kr, {khong,souhwanj}@ssu.ac.kr

Abstract. The vulnerability of deep learning models to adversarial attacks poses a significant challenge in artificial intelligence, particularly in deepfake detection. Adversarial training has emerged as a crucial defense mechanism, highlighting the importance of highly transferable adversarial samples. However, current attack methods often struggle with transferability in black-box scenarios. This research identifies a key limitation: white-box methods tend to overfit surrogate models, reducing their effectiveness against unknown attacks. To address this vulnerability, we propose the iterative "Random Masking Perturbation" method. This method reduces dependence on gradient or loss function behavior by not using full gradient information to create perturbations. Furthermore, by iteratively masking perturbations while still generating successful adversarial samples, our method automatically and selectively perturbs sensitive features that easily mislead the surrogate model, particularly those with the highest density in the low-frequency band, which is critical for achieving high transferability rates. Experimental results demonstrate significant enhancements in transferability rates, with Random Masking yielding improvements of up to 60%, offering promising strides in fortifying the robustness of adversarial training for deepfake detection models.

Keywords: Adversarial Attacks · Adversarial Training · Random Masking · Boosting Transferability · Low-frequency.

1 Introduction

In the ever-evolving landscape of security, marked by technological advancements, a perpetual struggle persists between those fortifying defenses and those exploiting vulnerabilities. This dynamic is particularly evident in the domain of security, such as software, web, network, or especially deepfake applications [12] where advancements in artificial intelligence have enabled sophisticated attacks and robust defense mechanisms. Defenders constantly strive to strengthen systems against various potential threats, while adversaries persistently probe for security gaps, underscoring the imperative for ongoing innovation in defensive strategies. This symbiotic relationship has fueled research endeavors to develop effective and resilient adversarial samples for training tasks. However, a notable

vulnerability of adversarial attacks persists – the transferability in black-box scenarios [5], where transferability refers to an adversarial sample's ability to attack various unknown models successfully. This highlights the critical need for continued exploration and innovation in addressing nuanced aspects, a focal point in our investigation.

In scenarios where the target model's architecture is unknown (black-box), surrogate models are utilized to generate adversarial samples, with the expectation that the countermeasure model in black-box scenarios shares a similar architecture with the surrogate models. However, specific models often prioritize certain critical features during training and generating, resulting in adversarial samples effective only on surrogate models, particularly when perturbations are uniformly applied across the audio signal. Although augmenting perturbations can boost the effectiveness of attacking deepfake detection models and influence transferability, it frequently decreases audio quality, potentially reducing them to meaningless noise. Moreover, real-world countermeasures exhibit various characteristics and architectures, hampering adversarial attacks' transferability. Additionally, within an adversarial training scheme, transferability is crucial because these adversarial samples contain diverse characteristics of attack perspectives, so augmented datasets with a minimal amount of adversarial samples should be used rather than requiring the collection of all samples for each attack scenario. Therefore, controlling the amount of perturbation noise while enhancing transferability becomes pivotal, serving as a metric to evaluate the efficacy of adversarial attack methods.

The traditional adversarial attack methods often optimize perturbations based on gradient loss and network parameters of the surrogate model [1, 4, 10] or geometric perspectives [11]. Furthermore, audio data is time series data that leads to perturbations persisting in continuous and high-amplitude formats [9]. The state-of-the-art attack methods were introduced, such as input diversity [14] based on data augmentation approaches or creating robust ensemble surrogate models [6] based on ensemble learning, aim to enhance transferability without relying on specific features of inputs and surrogate models. However, these state-of-the-art methods are resource-intensive and time-consuming because they need time to process through many diverse normalized data steps (input diversity) or huge and complicated architecture models (ensemble surrogate models), and they do not perform stably well in an audio domain. As a result, striking a balance between increasing perturbations for successful attacks stably and maintaining transferability while optimizing time and resource utilization presents a significant challenge in developing robust adversarial training strategies.

Our primary objective is to produce effective adversarial samples with a high transferability rate while minimizing time and resource usage compared with input diversity and ensemble surrogate model approaches to bolster the adversarial training of deepfake detection networks. To accomplish this aim, we introduce the "Random Masking Perturbation" method for adversarial processing, which

significantly boosts transferability rates and a notable performance improvement of up to 60% compared to conventional methods.

In this study, we have the following contributions:

- We proposed an attack method to enhance the transferability of weak adversarial attacks based on gradient optimization while minimizing time and resource costs.
- With random masking perturbation iteratively, we introduce a new way to focus attack only the sensitive features that are the factors that easily make countermeasure misclassification.

2 Preliminaries

2.1 Adversarial Attacks

Adversarial attacks have gained considerable attention in recent years, particularly in the context of image data. However, the audio domain introduces unique challenges, operating in both the time and frequency domains. While many existing adversarial attack methods have been designed primarily for images, extending these methods to audio signals has raised novel questions. The popular adversarial attack method is the Fast Gradient Sign Method (FGSM) [4], which perturbs an input to maximize loss, crafting an adversarial sample in a single step. The perturbation is determined by taking the sign of the gradient of the loss function concerning the input features and scaling it by a small constant ϵ to control the strength of the perturbation. To improve FGSM, there are several methods are introduced as the following methods.

- **IFGSM** (Iterative Fast Gradient Sign Method) [1] is an iterative extension of FGSM that refines the perturbation through multiple iterations to enhance the adversarial attack. In each iteration, the perturbation is adjusted by a small step α in the direction of the gradient, and the resulting adversarial sample is clipped to ensure it stays within an ϵ-bounded L_∞ norm.
- **MIFGSM** (Momentum Iterative Fast Gradient Sign Method) [3] incorporates momentum into the iterative process of IFGSM to stabilize the update directions, helping to avoid poor local maxima. This momentum term accumulates the gradient of the loss function over iterations, enhancing the transferability of adversarial examples across different models.
- **DIFGSM** (Diverse Inputs Iterative Fast Gradient Sign Method) [17] enhances IFGSM by introducing input diversity, where each input is randomly resized and padded during the attack process. This approach prevents the model from overfitting to specific perturbations, improving the robustness and transferability of the adversarial examples.

FGSM typically maintain a uniform perturbation magnitude $\delta_t = \bar{\delta_t}$, where $\delta < \epsilon$, and ϵ represents the perturbation budget value at time domain t. Different

from FGSM, the various methods based FGSM, employ various strategies to optimize perturbations of an adversarial sample $\mathcal{T}(\delta, x)$. However, these attacks were initially designed for the image domain, where frequency signals do not play a significant role. When applied to the audio domain, where frequency signals are crucial for transferability, their performance, while better than the original FGSM, still falls short. These methods tend to attack all frequency bands, leading to decreased transferability, particularly because perturbations in only low-frequency bands significantly affect transferability rates [7]. Therefore, we classify these FGSM-based methods as weak transferability audio adversarial attack methods.

The success of audio adversarial samples is closely related to the extent of perturbation applied. In simpler terms, samples with more substantial perturbations tend to have a higher success and transferability rate but significantly decrease in quality, impacting their effectiveness in deceiving the target system. Consequently, ensuring the quality of adversarial samples while maintaining a satisfactory Mean Opinion Score (MOS), a premier metric for assessing audio quality is paramount in the audio domain. Recognizing these challenges, our work aims to maintain the quality of adversarial audio by limiting the perturbation budget ϵ while enhancing the transferability rate.

2.2 Surrogate model

Adversarial attack scenarios can be categorized as white-box and black-box attacks. In a white-box attack, the attacker has full knowledge of the system, including the model architecture and training dataset. In contrast, a black-box attack occurs when the attacker has no information about the system. To create adversarial samples for black-box scenarios, attackers use surrogate models as proxies for the target models, hoping that the surrogate model's architecture and behavior are similar to those of the target model. Consequently, the surrogate model should strike a balance: it should neither be too simple nor too complex, and it should avoid highly specific techniques that are unlikely to generalize to the target model.

In this paper, we choose AASIST-SSL [16] and RawNet2 [15] as the surrogate models, and we use BTSE [2] solely to measure transferability in black-box attacks. AASIST-SSL is selected because it represents the best-performing model, while RawNet2 represents the worst-performing model in our experiments (see Table 1). All models are trained and tested on the ASVSpoof2021 dataset [8]. In detail, AASIST-SSL is developed using an attention graph neural network combined with self-supervised learning, leveraging Wav2vec2 - a powerful and large model for audio feature extraction, resulting in a model with 317,837,834 parameters (bigger than BTSE and RawNet2 about 18 times), but the performance is the best. BTSE and RawNet2 are simpler network models. BTSE incorporates a specific breathing-talking-silence encoder, enhancing its detection performance compared to RawNet2. All three models are well-known in the audio deepfake detection domain and perform well across various datasets.

Table 1: Parameter number and accuracy rate of audio deepfake detection models

	AASIST-SSL	**BTSE**	**RawNet2**
Parameter number	317,837,834	17,662,290	17,621,410
Accuracy (%)	98.69	94.01	91.37

3 Problem Statements and Motivation

In both the image and audio domains, two primary categories of attack methods exist: those employing uniformly perturbed values across the entire sample, such as FGSM, and those selectively targeting specific parts, such as one-pixel [13] deemed crucial for adversarial training of deepfake detection models. Nevertheless, although the approach selectively targeting specific parts to attack has high transferability, it has a low attack success rate in the audio domain because it is difficult to identify the correct point or the number of perturbations is insufficient to attack. Thus, enhanced transferability of the methods from perturbing the entire sample can balance both the attack success and transferability rate.

The transferability of an audio adversarial sample depends on both audio characteristics and the surrogate model. Therefore, to enhance the transferability rate, it is necessary to balance these factors. Notably, black-box adversarial attack methods are typically optimized based on surrogate models' gradients and loss functions. However, these approaches often produce adversarial samples that overfit the surrogate model, resulting in low transferability in black-box scenarios. This overfitting occurs because the model tends to learn and adapt to the important features extracted by the surrogate model throughout the process, leading to adversarial samples that are heavily dependent on the specific gradient behavior of the surrogate model. Consequently, this dependency limits their effectiveness in different contexts and reduces their utility in adversarial training strategies.

Simultaneously, the fastest way to enhance transferability in signal attacks is to focus on the low-frequency band [7], where speech energy predominantly resides (with high frequencies often being noise or echo). In other words, audio attack methods should target sensitive features within the low-frequency band to effectively prompt the surrogate model to change the label during adversarial processing rather than attacking the entire frequency range. However, our experiments show that most of weak adversarial attacks based on FGSM target the full frequency band.

To address these issues, our proposed method employs random masking perturbation to avoid relying on the complete information of gradients and network parameters, thereby preventing overfitting. Additionally, it continuously applies random masking, increasing the number of perturbations through many iterations while enabling the successful generation of adversarial samples. This ensures that the adversarial samples are predominantly perturbed at sensitive

features, mainly concentrated in the low-frequency audio band, thereby enhancing transferability.

4 Proposed Method

In this work, we deviate from the traditional approach of uniformly adding perturbations across entire samples. To enhance the transferability rate, we introduce chaotic and heterogeneous perturbations guided by the gradient direction of the model. This approach aims to effectively deceive the model while optimizing the perturbation count required for successful deception.

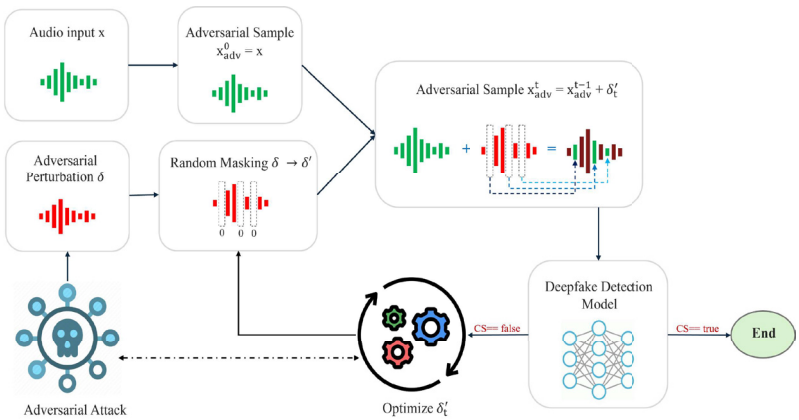


Fig. 1: Iterative random masking perturbation framework with conditional stop (CS), which halts the generation process early whenever the model misclassifies a spoof as a bonafide label

Specifically, as shown in Fig. 1, the algorithm for optimizing the adversarial perturbation δ remains unchanged. However, we introduce a modification where, after extracting δ from the usual attacks, we randomly mask portions of δ. Random masking of perturbations occurs with the optimized perturbation value δ'_t assigned at each iteration. Notably, δ'_t is optimized by the algorithm of the base attack methods such as IFGSM, MIFGSM, and DIFGSM in our experiments; our main contribution is the masking technique to the perturbation after its extraction.

The masked perturbation is calculated as the formula: $\delta'_t = \delta_t \times (1 - m_i)$ where m_i is generated from a Bernoulli distribution with the probability of masking p. Each m_i is independently set to 1 with p and to 0 with probability $(1 - p)$. The extent of this masking can be customized to various degrees, such as 30%, 40%, or 50%. This approach ensures that certain sections of the original sample

remain unmodified in the first iteration, and in subsequent iterations, some parts of the previous iterative adversarial sample x_{adv}^{t-1} also remain unmodified.

$$x_{adv}^{t} = x_{adv}^{t-1} + \delta' \tag{1}$$

As a result, our perturbations follow the model's gradient direction, maximizing the loss function until the model transitions its label from spoof to bonafide. The final perturbations have non-uniform values, following no specific pattern that a deepfake detection model can easily deduce.

Furthermore, although the Random Masking Perturbation method continuously changes the position of removed or hidden perturbation parts during many iterations, successful adversarial samples are still generated. This indicates that our method automatically selectively targets the sensitive features that are most likely to cause the surrogate model to easily change the label during the adversarial processing of audio samples. Our observations and analyses during experiments show that these sensitive audio features are predominantly located in the low-frequency band, which supports a high transferability rate for adversarial samples. By reducing dependence on gradient information and selectively targeting sensitive features, we successfully avoid overfitting the surrogate models and enhance the transferability of weak adversarial attack methods.

5 Experiments

5.1 Setup

In selecting a dataset for our evaluation and experiments, we prioritized one that includes noisy samples to simulate real-life scenarios. After careful consideration, we chose the ASVSpoof2021 DF eval dataset for its diversity and inclusion of various types of spoof audio data, particularly those from the renowned ASVspoof challenge. This dataset aligns well with our objective of testing under realistic conditions. To ensure consistency across all case experiments, We randomly selected 5000 samples from it and used these same samples for all case experiments.

For our experiments, we have selected FGSM, IFGSM, MIFGSM, and DIFGSM as the attack methods. While these techniques are typically initialized in the image domain, they are fundamentally gradient-based attack methods. After processing the raw data, both image and audio inputs are converted to signal data, facilitating easy extraction of gradient values from the surrogate model. This versatility allows for effectively applying FGSM-based techniques across various domains, including image and audio.

Furthermore, MIFGSM is a conventional technique designed to improve transferability rates for FGSM and IFGSM, while DIFGSM is another method to enhance transferability for FGSM, IFGSM, and MIFGSM. However, in our study, we found that despite their intended purpose of improving transferability, these

methods did not prove effective, and we aim to boost their transferability rate further. Therefore, the choice to focus solely on FGSM-based techniques in our experiments provides a sufficient basis for comparison.

As discussed in section 2.2, we have chosen AASIST-SSL and RawNet2 as appropriate surrogate models. To evaluate transferability across different architectures of deepfake detection models, we have also included BTSE. This model employs a specialized encoder that initially processes breath, silent, and talking segments as inputs for training.

Moreover, for a fair comparison, we adhere to a uniform perturbation budget ϵ of 0.3, a step size α of 0.01, and a maximum of 20 iterations. To be considered for the evaluation of transferability, the adversarial samples must first successfully attack the surrogate model. Therefore, the chosen samples must achieve a 100% attack success rate on the surrogate model.

Table 2: Transferability Rates (%) using AASIST-SSL and RawNet2 as surrogate models. Model (*) denotes the surrogate model in this case.

(a) Transferability Rates Using AASIST-SSL as Surrogate Model

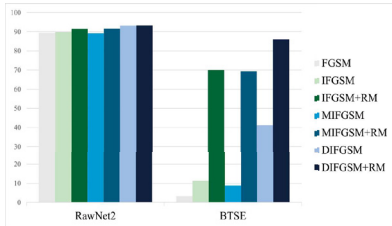
Method	AASIST-SSL*	RawNet2	BTSE
FGSM	100	89.48	3.27
IFGSM	100	89.88	11.34
IFGSM + RM	100	91.56	69.95
MIFGSM	100	89.29	8.79
MIFGSM + RM	100	91.68	69.33
DIFGSM	100	93.16	41.23
DIFGSM + RM	100	**93.21**	**85.92**

(b) Transferability Rates Using RawNet2 as Surrogate Model

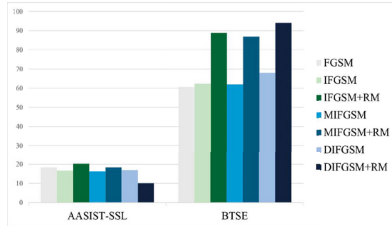
Method	RawNet2*	AASIST-SSL	BTSE
FGSM	100	18.33	60.63
IFGSM	100	16.71	62.36
IFGSM + RM	100	**20.26**	88.81
MIFGSM	100	16.22	61.92
MIFGSM + RM	100	18.36	86.96
DIFGSM	100	16.91	68.00
DIFGSM + RM	100	10.04	**94.11**

5.2 Results

Fig. 2a and Fig. 2b show the transferability rate of adversarial samples generated by several adversarial attacks and surrogate models, the detail result is in Table 2. The grey column represents FGSM, and the other paired colors indicate



(a) Transferability rate with surrogate model AASIST-SSL



(b) Transferability rate with surrogate model RawNet2

Fig. 2: Transferability rates (%) of adversarial attacks with different surrogate models

methods with and without applying Random Masking Perturbation, whereas the darker color signifies the application of Random Masking Perturbation. The flection random masking is 50%.

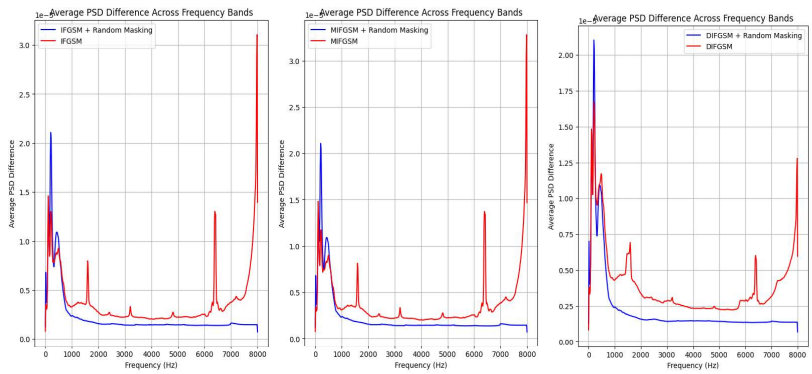
In Table 2, AASIST-SSL, a larger and stronger model than RawNet2, successfully transfers attacks to RawNet2 in both cases, with and without applying random masking. Specifically, BTSE is challenging to attack using normal adversarial methods due to its specialized encoder. However, our method significantly boosts the transferability of these weak attacks, with the best case showing an improvement of up to 60% in transferability rate.

In contrast, when using RawNet2 as the surrogate model, the transferability to AASIST-SSL, a strong deepfake detection model, remains weak in both cases, with and without applying our method. Our method only slightly enhances transferability in some cases because traditional attacks already struggle to deceive AASIST-SSL. Additionally, reducing perturbation further results in adversarial samples lacking sufficient perturbations to deceive the models. However, with the BTSE model, the enhanced transferability still outperforms traditional attacks.

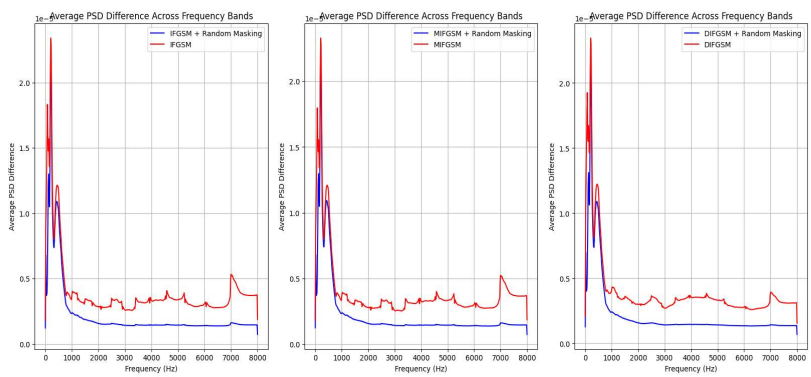
Our method operates in the gradient domain during each iteration, which is the core framework of the base attack methods we aim to enhance transferability. Notably, our approach doesn't require additional processing or augmented data. In our experiments, we maintained consistent parameters across all methods, including the maximum number of iterations and perturbation budgets, to ensure a fair comparison. The results demonstrate that by applying our method, the relatively simple IFGSM can achieve performance nearly on par with the more complex DIFGSM. This finding proves that our method can significantly reduce computational and resource costs while maintaining high effectiveness. By improving the efficiency of more straightforward attack methods, we eliminate the need for more intricate and resource-intensive techniques, thus optimizing the overall process of generating transferable adversarial examples.

5.3 Reasoning on the enhancing transferability

To explain the improvement in transferability rate, the key factor is the attack on low-frequency bands, which makes label switching easier, as already explained in paragraph 3 of section 3. For this reason, we investigated the power spectral density (PSD) of the perturbations from traditional adversarial attacks with or without applying our method - Random Masking Perturbation to see the power density in each frequency band. The perturbation δ is identified by subtracting the original audio x_{orig} from the adversarial audio x_{adv}:



(a) AASIST-SSL - IFGSM (b) AASIST-SSL - MIFGSM (c) AASIST-SSL - DIFGSM

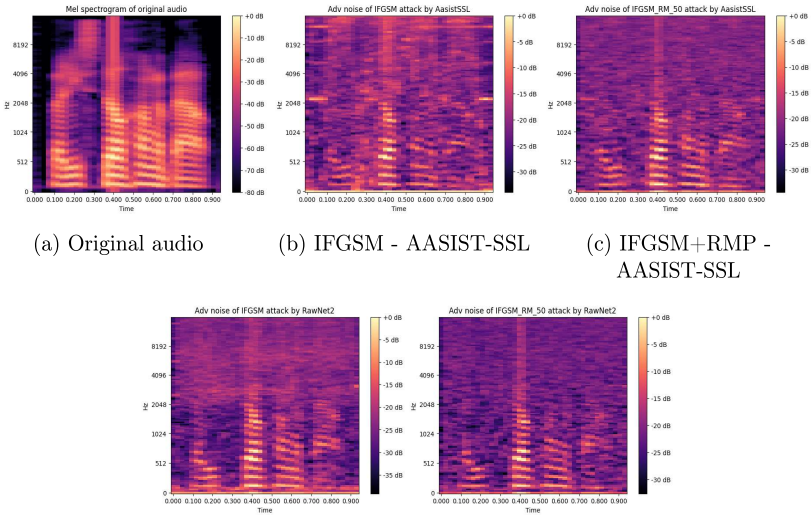


(d) RawNet2 - IFGSM (e) RawNet2 - MIFGSM (f) RawNet2 - DIFGSM

Fig. 3: Average power spectral density (PSD) difference of adversarial perturbation across frequency bands for different base attack methods generated by different surrogate models. The red line shows traditional attacks, and the blue line shows our method.

$$\delta = x_{adv} - x_{orig} \tag{2}$$

As shown in Fig. 3, the power spectral density (PSD) of adversarial perturbations without applying our method tends to be strong in the high-frequency bands. The average PSD of traditional attacks is higher than when applying our method in all cases (the red line), particularly in the case of the AASIST-SSL surrogate model. In contrast to traditional attacks, our method redirects the perturbations to the low-frequency band and stabilizes them within this band. The paper [7] demonstrated that attacking low-frequency bands can boost the transferability rate. This explains why we can improve the transferability.



(a) Original audio (b) IFGSM - AASIST-SSL (c) IFGSM+RMP - AASIST-SSL

(d) IFGSM - RawNet2 (e) IFGSM+RMP - RawNet2

Fig. 4: Mel-spectrogram of original audio and the perturbation generated by IFGSM apply with and without Random masking perturbation (RMP) using AASIST-SSL and RawNet2 surrogate models.

Moreover, using the iterative random masking perturbation scheme while adhering to the constraints of the surrogate model to switch the label from spoof to bonafide, the higher the number of masked reflections, the more the focus is forced onto the key features that enable achieving targeted deception of models. The final perturbations will be the result of collecting key features through each iterative random masking and optimizing perturbation. Thus, the probability ratio of masking perturbation directly affects the transferability; the higher the

ratio, the harder the constraints to generate an adversarial sample. The Fig. 3 show the best results occur when applying Random Masking Perturbation with 70% reflection, where the deviation of the PSD at the high-frequency band is the largest (the red line represents traditional attacks, and the blue line represents our method).

For the example shown in Fig. 4, this figure displays how the perturbation works on the frequency bands. Fig. 4a shows the original samples, Fig. 4b and Fig. 4c display the adversarial perturbation δ of IFGSM with and without our method using the AASIST-SSL surrogate model, and Fig. 4d and Fig. 4e show the adversarial perturbation of IFGSM with and without our method using the RawNet2 surrogate model. Although adversarial perturbations appear across the entire frequency band in both cases with and without applying our method, IFGSM methods tend to attack uniformly across all frequencies, including the high frequencies where the light color appears at the top of Fig. 4b and Fig. 4d. In contrast, our method primarily focuses on lower frequencies, especially where the speech energy is concentrated (the light color is mostly below). The transferability is evident as Fig. 4c and Fig. 4e successfully attack BTSE, whereas Fig. 4b and Fig. 4d fail to attack BTSE in black-box scenarios.

6 Conclusions

In conclusion, our research addresses the critical challenge of transferability in adversarial attacks, particularly in black-box scenarios, through the innovative "Random Masking Perturbation" method. This approach not only enhances transferability by up to 60% in black-box attacks but also optimizes resource usage, marking a significant advancement in adversarial training strategies. By iteratively limiting the use of full gradient and surrogate network parameter information and targeting sensitive features in the low-frequency band, our method provides insights into the fundamental vulnerabilities of deep learning models. This work has far-reaching implications for AI security, especially in domains like deepfake detection, and paves the way for more robust and efficient defense mechanisms against evolving threats. As we continue to pursue scalable solutions, our research positions itself at the forefront of AI security, contributing to the development of more resilient systems capable of withstanding a wide spectrum of potential attacks in an ever-evolving landscape of AI threats.

Acknowledgments. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2023-00263037, Robust deepfake audio detection development against adversarial attacks); and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2023-00230337, Advanced and Proactive AI Platform Research and Development Against Malicious Deepfakes).

References

1. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: 2018 IEEE security and privacy workshops (SPW). pp. 1–7. IEEE (2018)
2. Doan, T.P., Nguyen-Vu, L., Jung, S., Hong, K.: Bts-e: Audio deepfake detection using breathing-talking-silence encoder. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
3. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
5. Gu, J., Jia, X., de Jorge, P., Yu, W., Liu, X., Ma, A., Xun, Y., Hu, A., Khakzar, A., Li, Z., et al.: A survey on transferability of adversarial examples across deep neural networks. arXiv preprint arXiv:2310.17626 (2023)
6. Guo, F., Sun, Z., Chen, Y., Ju, L.: Towards the transferable audio adversarial attack via ensemble methods. Cybersecurity **6**(1), 44 (2023)
7. Liu, B., Zhang, J., Zhu, J.: Boosting 3d adversarial attacks with attacking on frequency. IEEE Access **10**, 50974–50984 (2022)
8. Liu, X., Wang, X., Sahidullah, M., Patino, J., Delgado, H., Kinnunen, T., Todisco, M., Yamagishi, J., Evans, N., Nautsch, A., et al.: Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2023)
9. Lu, Z., Han, W., Zhang, Y., Cao, L.: Exploring targeted universal adversarial perturbations to end-to-end asr models. arXiv preprint arXiv:2104.02757 (2021)
10. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
11. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
12. Nguyen-Vu, L., Doan, T.P., Bui, M., Hong, K., Jung, S.: On the defense of spoofing countermeasures against adversarial attacks. IEEE Access (2023)
13. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23**(5), 828–841 (2019)
14. Sun, J., Long, S., Ma, X.: Senattack: adversarial attack method based on perturbation sensitivity and perceptual color distance. Applied Intelligence **53**(23), 28937–28953 (2023)
15. Tak, H., Jung, J.w., Patino, J., Kamble, M., Todisco, M., Evans, N.: End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. arXiv preprint arXiv:2107.12710 (2021)
16. Tak, H., Todisco, M., Wang, X., Jung, J.w., Yamagishi, J., Evans, N.: Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. arXiv preprint arXiv:2202.12233 (2022)
17. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2730–2739 (2019)

Security Function for 5G-Advanced and 6G: Malicious UE Detection in Network Slicing

Jaehyoung Park[1], Jihye Kim[2], and Jong-Hyouk Lee[1] (✉)

[1] Department of Computer and Information Security & Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea
jaehyoung@pel.sejong.ac.kr, jonghyouk@sejong.ac.kr
[2] Department of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea
jihye@pel.sejong.ac.kr

Abstract. Starting with Software Defined Networking (SDN) and Network Functions Virtualization (NFV) technologies, network slicing is emerging as a critical technology in 5G-Advanced and 6G mobile communications to allocate logical networks end-to-end, meeting diverse service requirements. However, the advent of network slicing introduces new security threats, including inter-slice security, intra-slice security, and lifecycle security of network slicing. While recent research has focused on security orchestration technologies using SDN/NFV for network slicing and AI-based attack detection, there remains a gap in research on security by design within the components of 3GPP 5G systems. Therefore, this paper investigates security by design in the 5G-Advanced and 6G architectures. To address these issues, we propose a malicious User Equipment (UE) detection function to identify attacks within the network-slicing environment. We illustrate the interaction between the detection function and other 3GPP 5G system components for network slicing security with its algorithm for detecting malicious UEs in network slices.

Keywords: Network Slicing · Malicious UE · Security by Design.

1 Introduction

5G-Advanced and 6G are being researched to support various industries such as autonomous vehicles, remote surgery, smart factories, and smart cities, while also supporting diverse applications like the Internet of Everything, Virtual Reality, and Augmented Reality.

For this purpose, next-generation mobile communication technology must support enhanced Mobile BroadBand (eMBB) to provide data speeds of up to 10Gbps, Ultra-Reliable Low Latency Communication (URLLC) to support end-to-end latency of less than 1ms, and massive Machine Type Communication (mMTC) to support up to 1,000,000 connected devices per km^2. To address these challenges, network slicing technology, which divides the physical network into multiple logical networks, is gaining attention. This enables the

next-generation mobile communication architecture to meet diverse communication requirements, facilitating innovative changes [1].

Software Defined Networking (SDN), which separates the data plane and control plane of the network and provides a programmable architecture, along with Network Functions Virtualization (NFV), which allows traditional hardware-based network functions (e.g., firewalls, load balancers, routers) to run in a virtualized environment, are key technologies supporting network slicing [2].

Mobile communication networks are expected to evolve into more flexible and scalable forms. However, the advent of network slicing introduces new security threats, such as security between slices, security within slices, and network slicing lifecycle security [3].

This paper proposes a Malicious User Equipment (UE) Detection Function (MUDF) to detect security threats in a network-slicing environment, which can embed security technologies within the 5G-Advanced and 6G architectures. The MUDF is a Network Function (NF) designed to detect a malicious UE that negatively impacts a Network Slice (NS). The MUDF can interact with components of the 3rd Generation Partnership Project (3GPP) 5G system architecture. The contribution of this paper is threefold:

- Propose a 3GPP 5G system architecture in where the MUDF is deployed.
- Illustrate the interaction between the MUDF and 3GPP 5G system components for NS security.
- Present an algorithm for detecting malicious UEs in a network-slicing environment.

The structure of this paper is outlined as follows. Section 2 introduces the background knowledge of network slicing and reviews research on network slicing security technologies. Section 3 introduces the MUDF for detecting malicious UEs in the network-slicing environment. Section 4 describes the proposed MUDF. Finally, Section 5 concludes this paper.

2 Related Work

2.1 Network Slicing

The Next Generation Mobile Networks (NGMN) Alliance first introduced network slicing to realize network softwarization [4]. The network slicing multiplexes the physical network into separate virtual networks, allowing heterogeneous services to coexist within the same network architecture by allocating network computing, storage, and communication resources among specialized services. Network slicing leverages network virtualization technologies such as SDN, NFV, cloud and edge computing to achieve virtualization and automation of the infrastructure. This enables scalability, flexibility, customization, and isolation for specific services [5,6,7].

Standardization organizations such as 3GPP and ITU define the service areas of existing 5G as eMBB, URLLC, and mMTC. Additionally, 3GPP's Release-18 investigates and introduces features to provide services to a variety of devices,

including not only smartphones but also eXtended Reality and cloud gaming, vehicle devices, and low-complexity UEs [8]. The network slicing enables efficient resource allocation, service customization, and support for various applications. Therefore, it allows for providing specialized services that meet the requirements of new services.

Such network slicing will continue to evolve and be utilized not only to meet the requirements of 5G services but also for 5G-Advanced and 6G [9]. The network slicing, which is establishing itself as a key feature for 5G-Advanced and 6G, considers not only the core network but also end-to-end for the entire network infrastructure, including the radio access network [10].

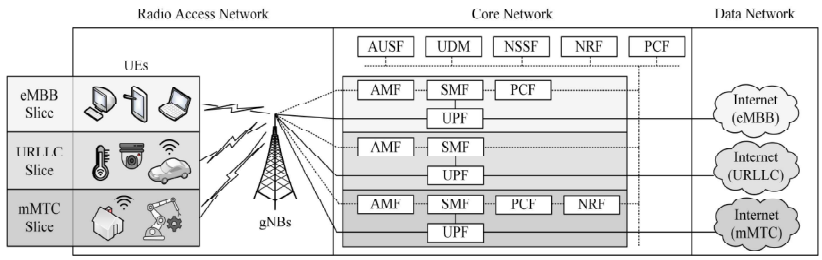


Fig. 1. Overview of Network Slicing

2.2 Security Technologies for Network Slicing

Network slicing has emerged as a key feature of 5G-Advanced and 6G, enabling efficient resource allocation and service customization. However, as networks become more advanced and services diversify, the security vulnerabilities of network slicing have also increased. The security vulnerabilities arising from the network slicing can lead to issues such as data breaches, network, and service disruptions. Therefore, it is essential to consider network security and privacy protection in the network slicing implementations [4,11].

Martini et al. [12] present an extended MANO framework based on usage control to enhance network slicing and orchestration security by integrating continuous and closed-loop access control mechanisms.

Wijethilaka et al. [3] describe how various security attacks such as Distributed Denial of Service (DDoS) attacks, Man-in-the-Middle attacks, and botnets originate in the device domain and propagate through the network to the network-slicing ecosystem. To address this, the author proposes a security orchestrator that enables responsive and preventive security mechanisms, life-cycle management of security functions, secure communication between slices, and isolation of compromised devices.

Bisht et al. [13] explain the impact of DDoS attacks in a network-slicing environment, such as delayed UE authentication during inter-slice handovers,

using metrics like average user waiting time and the number of dropped requests. Additionally, the author provides algorithms for attack detection and localization to secure the network slicing from such DDoS attacks.

Benzaid et al. [14] propose FortisEDoS, a deep transfer learning-based mechanism for detecting Economical Denial of Sustainability (EDoS) attacks, aiming to protect network slicing from the dynamic and increasingly sophisticated nature of DDoS attacks. This mechanism leverages advanced graph and recurrent neural network capabilities to capture spatiotemporal correlations, effectively identifying malicious behavior. Additionally, it utilizes knowledge from previous network slicing to enhance its response to attacks.

Javadpour et al. [15] present the challenge of maximizing the number of NSs while ensuring slice isolation in the network slicing. Slice isolation in the network slicing serves as a mechanism to protect slices from DDoS attacks. To achieve this, the author proposes a reinforcement learning model called slice isolation-based reinforcement learning, which utilizes five optimal graph features.

Bekkouche et al. [16] explain that although Machine Learning (ML) based solutions are increasingly being used to detect and mitigate recent network attacks, they still do not consider device-specific characteristics that could affect the quality of attack detection. Additionally, existing solutions do not consider the diverse characteristics and conditions of various network slicing services. Therefore, the author proposes a protection approach for network slicing based on an adaptive ML-based orchestrator that tailors security measures to the specific characteristics of each service.

Existing research has addressed various security threats that may occur between slices, within slices, and throughout the network slicing lifecycle in the network-slicing environment. To counter these threats, researchers have proposed frameworks based on SDN/NFV technologies and suggested the use of security orchestrators. Additionally, studies have explored authentication technologies for network slicing security and AI-based methods for detecting attacks. However, there is limited research on identifying attackers after detecting attacks in the network-slicing environment, and a lack of studies focused on embedding security into standardized 3GPP 5G system components. To address this gap, this paper proposes security functions for 5G-Advanced and 6G and integrates them into the mobile communication system.

3 Malicious UE Detection Function for Network Slicing

This section describes the proposed MUDF, which is designed to detect malicious UE in a network-slicing environment. To this end, the interaction between the network slicing security architecture with the MUDF applied, and the 5G-Advanced components are described. Additionally, the operation of the proposed architecture and malicious UE detection algorithms in the network-slicing environment are presented.

3.1 MUDF in the 3GPP 5G System Architecture

3GPP has defined NFs that can operate based on SDN and NFV technologies for the 5G system. They have also developed a service-based architecture that supports interaction between each NF through a Service-based Interface (SBI). Based on this architecture, research is being conducted to enhance existing commercial 5G systems through 5G-Advanced. It is also anticipated that the 3GPP 5G system architecture will be applied in the future 6G mobile communication systems. Therefore, the MUDF for detecting attacks in a network-slicing environment is designed based on the 3GPP 5G system architecture and can be applied as shown in Fig. 2.

The MUDF is designed considering the following elements. First, the MUDF must be able to apply to both 5G-Advanced and 6G architectures. To achieve this requirement, the MUDF must be able to access and interact with the NFs and services defined in the existing 3GPP 5G system. Therefore, it is proposed that the MUDF be registered as a core NF through the Network Repository Function (NRF) to provide SBI-based services within the 3GPP 5G system architecture.

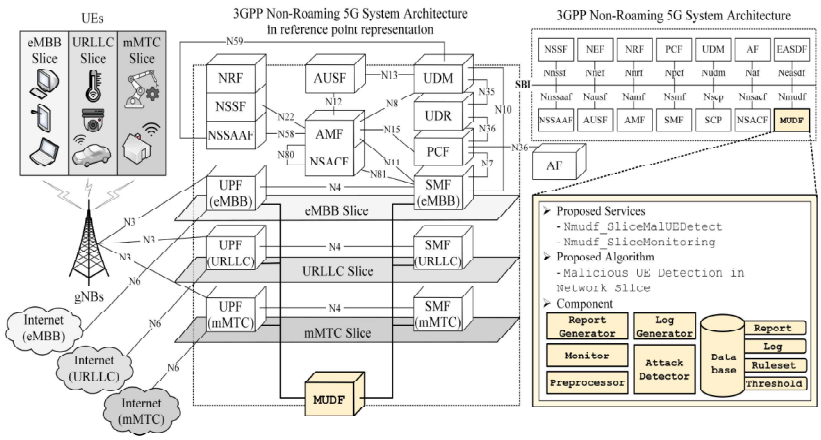


Fig. 2. MUDF deployed in the 3GPP 5G System Architecture

Secondly, the MUDF must monitor each NS in the network slicing environment. To achieve this, the MUDF must interact with the User Plane Function (UPF) that handles the User Plane (UP). As a result, the MUDF can monitor the UP packets of Packet Data Unit (PDU) sessions processed by the UPF allocated to each NS.

Thirdly, the MUDF must provide the capability to detect UEs that perform attacks in the network-slicing environment and identify which slice they are located in. To meet this requirement, the MUDF must provide algorithm for detecting security threats within NSs and identifying malicious UEs.

Fourthly, the MUDF must be able to request and retrieve subscription information of UEs from the core network to detect those performing attacks in the network-slicing environment. To achieve this, it is proposed that the MUDF interact with the Unified Data Management (UDM) within the core network, which manages the overall data.

Fifthly, the MUDF must provide a notification function to the core network when attacks and malicious UEs are detected in the network-slicing environment. To achieve this requirement, the MUDF must report the attack detection results to the Session Management Function (SMF) that manages sessions within the core network.

Based on these requirements, the MUDF can interact with existing 3GPP 5G system architecture components without modifying them. Additionally, due to the scalability of the MUDF, it can detect attacks even if an attacker targets a specific slice or multiple slices within the network-slicing environment.

3.2 MUDF Procedure for Securing Network Slicing

The main operation process of the MUDF is illustrated in Fig 3. The MUDF can provide the `Nmudf_SliceMalUEDetect` service and the `Nmudf_SliceMonitoring` service within the core network. The SMF and the UPF are able to consume these services through the SBI. The `Nmudf_SliceMalUEDetect` service is designed to detect attacks and malicious UEs in a network-slicing environment. First, to register the MUDF as an NF in the core network, it sends an *Nnrf_NFMangement_NFRegister* request to the NRF, including the NF type, NF ID, and NF IP. The NRF stores the NF profile of the MUDF and sends a response.

Next, the SMF, which intends to consume the MUDF's services, sends a `Nmudf_SliceMalUEDetect` request to the MUDF. This request includes the SMF ID, PDU session ID, and Subscription Permanent Identifier (SUPI) list. The SMF manages one or more NSs, and the SUPI is a value that identifies the UE within the core network. The MUDF then sends a response to the SMF for the `Nmudf_SliceMalUEDetect` request. The SMF requests the UPF, connected via the N4 interface, to subscribe to the `Nmudf_SliceMalUEDetect` service including the MUDF's IP and ID. Here, the UPF is responsible for processing UP data within the NS, and it subscribes to the MUDF to ensure NS security.

The UPF responds to the SMF regarding the `Nmudf_SliceMalUEDetect_Subscription` request and sends an `Nmudf_SliceMonitoring` request to the MUDF, which includes the PDU session ID and PDU session IP list. The PDU session IP refers to the IP address assigned to a UE when it is registered with the core network and is used to access services specific to each NS.

The MUDF responds to the `Nmudf_SliceMonitoring` request from the UPF, and the UPF forwards the UP data packets to the MUDF. The UP data packet refers to the packets generated when a UE utilizes services specific to each NS. Subsequently, the MUDF collects the packets forwarded by the UPF to detect malicious UEs that perform attacks within the NS.

When the MUDF detects attacks in one or multiple slices, it must identify which UE in which slice is causing the attacks. Therefore, when attacks are detected in the monitored PDU session, the MUDF identifies the malicious UE causing the attacks. It does this by using the data received from the SMF and UPF during the MUDF subscription process. Additionally, to detect the slice containing the malicious UE, the MUDF sends a *Nudm_SDM_Get* request to the UDM. This request includes the ID of the SMF managing the NS, the subscription data type (e.g., session management subscription data), and the key values for data retrieval (e.g., SUPI, PDU session ID, PDU session IP).

The UDM sends a *Nudr_DM_Query* request to the Unified Data Repository (UDR) to query the subscription data of the UE requested by the MUDF. The *Nudr_DM_Query* request includes the data set ID (e.g., subscription data) and the SUPI, PDU session ID, and PDU session IP received from the MUDF. The UDR responds to the UDM with the UE subscription data, including the Network Slice Selection Assistance Information (NSSAI). The UDM then forwards this information to the MUDF, enabling it to detect which NS the attacks have occurred in.

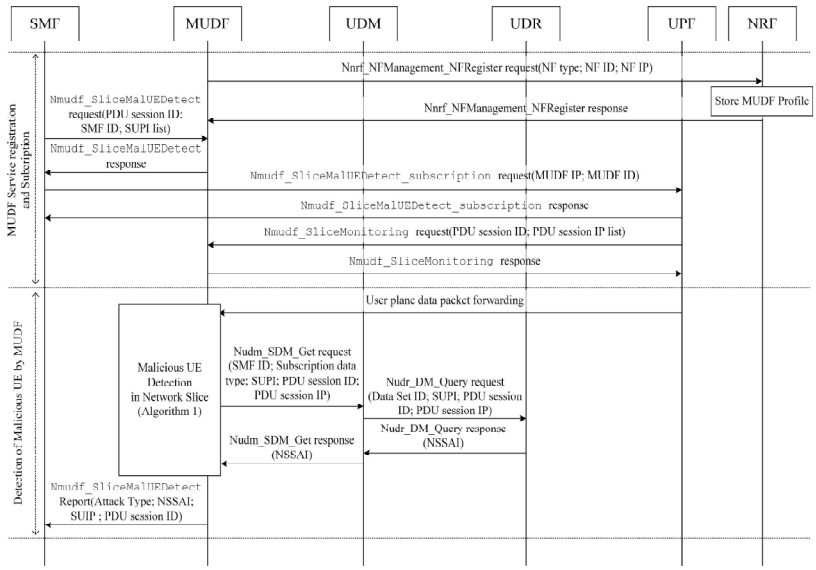


Fig. 3. Operation Process of the MUDF

The MUDF parses the received UE subscription data and sends a `Nmudf_SliceMalUEDetect` report to the SMF. The report includes the NSSAI, SUPI, Attack Type, and PDU session ID. Here, the Attack Type indicates the type of attack detected by the MUDF in the NS (e.g., DoS, data hijacking). The NSSAI is a value used to identify and select NSs. It consists of the Slice/Service Type (SST) and the Slice Differentiator (SD). The Slice/Service Type (SST)

standardized by 3GPP represents different types of slices or services based on defined values. In the 3GPP standard, 1 is defined as eMBB, 2 as URLLC, 3 as massive IoT, 4 as vehicle-to-everything, 5 as high-performance machine-type communications, and 6 as high data rate and low latency communications. SUPI and PDU session ID are used to identify the UE that caused the attacks in the slice specified by the NSSAI.

Thus, the MUDF analyzes the UP packets transmitted and received within the NS to detect attacks. It then reports the detected attacks to the SMF, which manages sessions in the NS, thereby ensuring a secure network-slicing environment.

3.3 MUDF Operating Algorithm

The malicious UE detection algorithm is presented in Algorithm 1. Table 1 presents the notations used in Algorithm 1.

Table 1. Notation

Notation	Description
ID_{SMF}	Identifier of the SMF on the core network
ID_{UDM}	Identifier of the UDM on the core network
$ID_{PDUsession}$	Identifier of the PDU session on the core network
IP_{SMF}	IP address of the SMF on the core network
IP_{UDM}	IP address of the UDM on the core network
$IP_{PDUsession}$	IP address of the PDU session on the core network
$SUPI$	UE identification value in the core network
$Packet_{NS}$	Packet transmitted and received within the NS
$Packet_{Pre}$	Preprocessed packet
$Result_{Detection}$	Attack detection result
$Ruleset$	Ruleset for attack detection
$Threshold$	Threshold for attack detection
$AttackType$	Types of attack detected
Mal_{SUPI}	SUPI of malicious UE
$Mal_{ID_{PDUsession}}$	PDU session ID of malicious UE
$NSSAI$	Network Slice Selection Assistance Information
$SubDataType$	Types of subscription data
$AttackReport$	Report of attack detection results
$AttackLog$	Log of attack detection results
$SessionStatusLog$	Log of a session status
$Request(x)$	Request for x
$Parsing(x)$	Parse x
$Monitoring(x)$	Monitor x
$Preprocessing(x)$	Preprocess x
$DetectAttack(x, y, z)$	Detect attacks based on x, y and z
$Query(x)$	Query x
$Send(x, y)$	Send y to x
$GenReport(w, x, y, z)$	Generate a report of attack detection based on w, x, y, z
$GenLog(x)$	Generate a log of attack detection based on x

Algorithm 1 is used by the MUDF to detect malicious UEs within the NS. The algorithm begins by parsing the data from the `Nmudf_SliceMalUEDetect` request and `Nmudf_SliceMonitoring` request messages, which are sent by the SMF and UPF to utilize the MUDF service. The parsed data includes $ID_{SMF}, SUPI, ID_{PDUsession}$, and $IP_{PDUsession}$. These data are the basic information for the MUDF to monitor network slices and identify malicious UEs.

Algorithm 1 Malicious UE Detection in a NS

1: begin
2: input $Request(\texttt{SliceMalUEDetect}), Request(\texttt{SliceMonitoring})$
3: $ID_{SMF}, SUPI, ID_{PDUsession} \leftarrow Parsing(\texttt{SliceMalUEDetect})$
4: $IP_{PDUsession} \leftarrow Parsing(\texttt{SliceMonitoring})$
5: **while** true **do**
6: $Packet_{NS} \leftarrow Monitoring(IP_{PDUsession})$
7: $Packet_{Pre} \leftarrow Preprocessing(Packet_{NS})$
8: $Result_{Detection} \leftarrow DetectAttack(Packet_{Pre}, Query(Ruleset), Query(Threshold))$
9: **if** $Result_{Detection} == True$ **then**
10: $AttackType, Mal_{SUPI}, Mal_{IP_{PDUsession}} \leftarrow Parsing(Result_{Detection})$
11: $NSSAI \leftarrow Send(IP_{UDM}, Request(SDM_Get(ID_{SMF}, SubDataType, SUPI, ID_{PDUsession}, IP_{PDUsession})))$
12: $AttackReport \leftarrow GenReport(AttackType, NSSAI, Mal_{SUPI}, Mal_{IP_{PDUsession}})$
13: $Send(IP_{SMF}, AttackReport)$
14: $AttackLog \leftarrow GenLog(AttackReport)$
15: **return** $AttackLog$
16: **else**
17: $SessionStatusLog \leftarrow GenLog(ID_{PDUsession})$
18: **return** $SessionStatusLog$
19: **end if**
20: **end while**
21: end

As shown in Fig. 3, after the MUDF sends the `Nmudf_SliceMonitoring` response to the UPF, and it monitors the interface corresponding to $IP_{PDUsession}$. The monitored $Packet_{NS}$ undergoes a data preprocessing step in the Preprocessor, resulting in $Packet_{Pre}$. The preprocessed data is then fed into the Attack Detector, which begins the attack detection process based on the Ruleset and Threshold present in the database. The Attack Detector can utilize signature-based or AI-based intrusion detection techniques. Additionally, when the algorithm starts, the MUDF monitors the PDU sessions managed by the UPF that has subscribed to its service through the Monitor.

If attacks are detected, a $Result_{Detection}$ is generated. This result is then parsed to obtain $AttackType, Mal_{SUPI}$, and $Mal_{ID_{PDUsession}}$. On line 10, the type of attack and the UE involved in the NS can be identified. However, during

this process, since the MUDF does not have NS information, line 11 is executed to obtain the NS information of the UE that caused the attack.

On line 11, the MUDF sends the *Nudm_SDM_Get* request to the UDM to obtain the NSSAI. The MUDF then generates an *AttackReport* using the Report Generator. The generated *AttackReport* includes the values of $AttackType$, $NSSAI$, Mal_{SUPI}, and $Mal_{ID_{PDUsession}}$.

The generated *AttackReport* is sent to the SMF, and the MUDF generates an *AttackLog* using the Log Generator. However, if no attack is detected, the Log Generator creates a *SessionStatusLog* and records it in the MUDF. Through this algorithm, when an attack occurs in the NS, the SMF receives the *AttackReport* and can identify the UE causing the attack and the specific NS where the UE is located.

4 Discussion

In 5G-Advanced and 6G, the significance of network slicing is expected to increase further to meet the diverse requirements of various services. While ongoing research aims to implement secure NS in next-generation mobile communication systems, there is still a notable lack of studies on applying these within the 3GPP mobile communication system architecture.

Since Release 15, 3GPP has been investigating security in a network-slicing environment. Release 15 focused on NS management security, UE authentication in NS, confidentiality and integrity protection of NS identifiers, and NF authentication for each NS. Release 16 addressed NS-specific authentication and authorization, while Release 17 explored application functions with confidentiality protection for NS identifiers. Most of the technologies researched are transport layer security-based authentication techniques for security management.

However, even authenticated entities can pose security threats due to malicious behavior in the network-slicing environment. Therefore, for 5G-Advanced in Release 18, research must extend beyond authentication technologies to address these security threats.

Detecting attacks is the first crucial step in responding to security threats. From this perspective, this paper presents a method and security architecture for detecting attacks in the network-slicing environment by integrating the MUDF into the 3GPP mobile communication system architecture. The proposed security architecture and operational process incorporating MUDF exemplify security by design within mobile communication systems. Additionally, the interaction between MUDF and 3GPP 5G system components facilitates the integration of currently researched attack detection technologies. For example, signature-based and AI-based attack detection technologies can be integrated as part of MUDF.

This research offers new perspectives on addressing security issues in 5G-Advanced and 6G, contributing to the creation of a more secure next-generation mobile communication environment.

5 Conclusion

In this paper, we proposed a new security function to detect malicious UE performing attacks in the 5G-Advanced and 6G network-slicing environment. Our algorithm enhances security by integrating identity verification, behavior analysis, and anomaly detection to identify potential threats, while utilizing the services of NFs defined in the 3GPP mobile communication system. The proposed security function can significantly improve the security and resilience of network slicing. Future work will focus on optimizing the detection algorithms and exploring the integration of additional security functions to further enhance the security of 5G-Advanced and 6G network slicing.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00796, Research on Foundational Technologies for 6G Autonomous Security-by-Design to Guarantee Constant Quality of Security).

References

1. ITU-R M.2160: Framework and overall objectives of the future development of IMT for 2030 and beyond (2023). https://www.itu.int/rec/R-REC-M.2160-0-202311-I/en
2. ETSI GR NFV-EVE 018 V5.1.1.: Network Functions Virtualisation (NFV) Release 5; Evolution and Ecosystem; Report on Multi-tenancy in NFV (2024)
3. Wijethilaka, S. and Liyanage, M.: The role of security orchestrator in network slicing for future networks. Journal of Communications and Networks **25**(3), 355-369 (2023)
4. Alwis, C.D., Porambage, P., Dev, K., Gadekallu, T.R., and Liyanage, M.: A Survey on Network Slicing Security: Attacks, Challenges, Solutions and Research Directions. IEEE Communications Surveys & Tutorials **26**(1), 534-570 (2024)
5. Zharabad, A.J., Yousefi, S., and Kunz, T.: Network slicing in virtualized 5G Core with VNF sharing. Journal of Network and Computer Applications 215(103631) (2023)
6. Popovski, P., Trillingsgaard, K.F., Simeone, O., and Durisi, G.: 5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View. IEEE Access **6**, 55765-55779 (2018)
7. Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., and Flinck, H.: Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. IEEE Communications Surveys & Tutorials **20**(3), 2429-2453 (2018)
8. Lin, X.: An Overview of 5G Advanced Evolution in 3GPP Release 18. IEEE Communications Standards Magazine **6**(3), 77-83 (2022)
9. Ebrahimi, S., Bouali, F., and Haas, O.C.L.: Resource Management From Single-Domain 5G to End-to-End 6G Network Slicing: A Survey. IEEE Communications Surveys & Tutorials (2024)
10. Odida, M.: Network Slicing in Software Defined Networking for 5G (2024)
11. Singh, P.S., Singh, M.P., Hegde, S., and Gupta, M.: Security in 5G Network Slices: Concerns and Opportunities. IEEE Access **12**, 52727-52743 (2024)
12. Martini, B. et al.,: Pushing Forward Security in Network Slicing by Leveraging Continuous Usage Control. IEEE Communications Magazine **58**(7), 65-71 (2020)

13. Bisht, H., Patra, M., and Kumar, S.: Detection and Localization of DDoS Attack During In-ter-Slice Handover in 5G Network Slicing. 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), pp. 798-803 (2023)
14. Benzaïd, C., Taleb, T., Sami, A., and Hireche, O.: A Deep Transfer Learning-Powered ED-oS Detection Mechanism for 5G and Beyond Network Slicing. GLOBECOM 2023 - 2023 IEEE Global Communications Conference, pp. 4747-4753 (2023)
15. Javadpour, A., Ja'fari, F., Taleb, T., and Benzaïd, C.: Reinforcement learning-based slice isolation against ddos attacks in beyond 5g networks. IEEE Transactions on Network and Service Management **20**(3), 3930-3946 (2023)
16. Bekkouche, R., Omar, M., and Langar, R.: Securing 5G Network Slices with Adaptive Ma-chine Learning Models as-a-Service: A Novel Approach. GLOBECOM 2023 - 2023 IEEE Global Communications Conference, pp. 4754-4759 (2023)

Simple Perturbations Subvert Ethereum Phishing Transactions Detection: An Empirical Analysis

Ahod Alghureid and David Mohaisen

University of Central Florida

Abstract. This paper explores the vulnerability of machine learning models, specifically Random Forest, Decision Tree, and K-Nearest Neighbors, to very simple single-feature adversarial attacks in the context of Ethereum fraudulent transaction detection. Through comprehensive experimentation, we investigate the impact of various adversarial attack strategies on model performance metrics, such as accuracy, precision, recall, and F1-score. Our findings, highlighting how prone those techniques are to simple attacks, are alarming, and the inconsistency in the attacks' effect on different algorithms promises ways for attack mitigation. We examine the effectiveness of different mitigation strategies, including adversarial training and enhanced feature selection, in enhancing model robustness.

1 Introduction

The proliferation of machine learning models in various domains has brought significant advancements in decision-making processes. However, concerns regarding the robustness and security of these models have also emerged alongside these advancements. Adversarial attacks, wherein small, carefully crafted perturbations are introduced into the input data to cause misclassification, seriously threaten the reliability of machine learning systems. Understanding the susceptibility of these models to adversarial attacks is crucial for developing robust and trustworthy AI systems.

The increasing prevalence of machine learning in cybersecurity applications has significantly improved the detection and prevention of various cyber threats. Among these threats, fraudulent activities such as phishing, scamming, and fake initial coin offerings (ICOs) pose substantial financial and personal data security risks. Machine learning models, particularly classification algorithms, have been deployed to identify and mitigate these threats with notable success. However, studies have raised concerns about the robustness and reliability of these models [18,5,8,3,4,2,1]. The attacks presented in the literature (see section 2) are effective yet sophisticated.

This study investigates the impact of extremely simple adversarial attacks on different machine learning models used in fraudulent transaction detection, specifically focusing on Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN) classifiers. By employing the Fast Gradient Sign Method, a widely recognized adversarial attack technique, we assess the performance degradation of these models when subjected to simple, realistic adversarially crafted inputs [35]. The primary objective is to understand how these models are susceptible to adversarial manipulation and explore potential mitigation strategies to enhance their robustness.

Previous research has highlighted the vulnerability of ML models to adversarial attacks [28]. This study contributes to the existing body of knowledge by providing a detailed empirical analysis of the effects of simple adversarial attacks on fraud detection models and proposing practical approaches to mitigate these effects. Our findings reveal inconsistency across algorithms for their tolerance of simple manipulation, underscoring the importance of selecting appropriate models and implementing robust defense mechanisms tailored to specific applications.

2 Literature Review

Transaction fraud is common in cryptocurrency and calls for fraud detection. Agarwal *et al.* [6] analyzed malicious Ethereum transactions, including phishing and scams, to enhance threat detection capabilities. Rabieinejad *et al.* [29] employed Ethereum transaction data to develop models for identifying cyber threats. Sanjalawe *et al.* [30] used the Benchmark Labeled Transactions Ethereum dataset to detect abnormal transactions related to illicit activities. Zola *et al.* [39] analyzed WalletExplorer data to classify Bitcoin addresses and reduce anonymity, aiding in identifying entities involved in illegal activities. Yang *et al.* [37] focused on applying transaction datasets in the Bitcoin network to detect spam transaction attacks. Mozo *et al.* [24] applied transaction datasets to detect cryptomining attacks within the Monero

Recent research demonstrated the vulnerability of deep convolutional neural networks (CNNs) to adversarial attacks [25]. In [12], Cartella et al. adapted algorithms to imbalanced tabular data for fraud detection, achieving a perfect attack success rate. Bhagoji *et al.* [10] introduced a Gradient Estimation black-box attacks to a target model's class probabilities, achieving near-perfect success rates for both targeted and untargeted attacks on deep neural networks (DNNs).

Generative Adversarial Networks (GANs) generate adversarial examples (AEs) and handle data perturbations in cryptocurrency transaction datasets. Table 1 summarizes these methods and highlights key findings across different domains, underscoring the diverse methods employed to address security challenges in cryptocurrency networks. For instance, Fidalgo *et al.* [20] utilized GANs for synthetic data generation and augmentation to tackle class imbalance in Bitcoin. Agarwal *et al.* [6] employed Conditional GANs (CTGAN) to create realistic adversarial data for Ethereum transactions.

Rabieinejad *et al.* [29] used Conditional GANs (CTGAN) and Wasserstein GANs (WGAN) to generate synthetic samples for augmenting Ethereum transaction datasets, enhancing detection capabilities against cyber threats. Sanjalawe *et al.* [30] used Semi-Supervised GANs and feature extraction techniques to perturb features in Ethereum transaction datasets. Zola *et al.* [38] focused on data augmentation using various GAN configurations to generate additional data for underrepresented classes in Bitcoin transactions. Zola *et al.* [39] investigated different GAN architectures to generate synthetic Bitcoin address data to mitigate class imbalance and improve entity classification.

Yang *et al.* [37] used a Wasserstein Generative Adversarial Network with divergence (WGAN-div) to generate adversarial examples for spam transaction detection in the Bitcoin network. Mozo *et al.* [24] utilized WGANs to create synthetic network traffic data to detect cryptomining attacks in the Monero (XMR) network.

Table 1: Summary of some of the prior work.

Paper Title	Year	Adversarial Techniques	Applications in Cryptocurrency
Li *et al.* [22]	2018	GANs	Anomaly detection, Secure Water Treatment
Qingyu *et al.* [19]	2019	IFCM, AIS, R3	Fraud detection, TaoBao
Ba *et al.* [9]	2019	GANs	Credit card fraud, 31-feature dataset
Ngo *et al.* [26]	2019	GANs	Anomaly detection, MNIST, CIFAR10
Zola *et al.* [38]	2020	GANs, data augmentation	Bitcoin, WalletExplorer (categorized addresses)
Yang *et al.* [37]	2020	WGAN-div, GRU-based detection	Bitcoin, custom spam transaction dataset
Shu *et al.* [31]	2020	GANs	Intrusion detection, network traffic
Mozo *et al.* [24]	2021	WGANs, synthetic traffic	Monero, custom cryptomining dataset
Fursov *et al.* [17]	2021	Black-box attacks	Transaction records
Agarwal *et al.* [6]	2022	CTGAN, K-Means Clustering	Ethereum, Etherscan dataset (2,946 malicious accounts)
Fidalgo *et al.* [20]	2022	GANs, data augmentation	Bitcoin, Elliptic dataset (200K+ transactions)
Zola *et al.* [39]	2022	Various GANs, adversarial learning	Bitcoin, WalletExplorer (16M+ addresses)
Rabieinejad *et al.* [29]	2023	CTGAN, WGAN	Ethereum, 57K normal, 14K abnormal transactions
Sanjalawe *et al.* [30]	2023	GANs, feature extraction	Ethereum, labeled transactions (normal, abnormal)

Research Gap. Despite advancements in machine learning and blockchain technology, critical gaps persist in effectively understanding mitigating adversarial attacks and emerging security threats within cryptocurrency networks. AEs grounded in the context of application are underexplored. AEs in the feature space that leverage targeted and untargeted attacks, transaction fraud, smart contract exploits, etc., are lacking. This underscores the imperative for further analysis. This study aims to bridge this gap by investigating the robustness of machine learning-based phishing detection algorithms against *simple manipulations*, comparing the effectiveness of various algorithms in resisting such attacks, and exploring mitigation strategies to enhance their resilience. Our approach involves evaluating the algorithms' susceptibility to subtle feature manipulations and conducting a comparative analysis to identify the most robust models.

3 Research Questions

This research explores the robustness, comparative performance, and mitigation strategies of machine learning-based phishing detection algorithms for Ethereum transactions in adversarial manipulations. The following questions highlight the core issues and the necessity to address them in light of existing literature.

RQ1.Are machine learning-based phishing detection algorithms for Ethereum robust against simple manipulations of individual features? This question is motivated by the vulnerability of machine learning (ML) models to adversarial attacks, as demonstrated in several studies [25,12,10,32,13]. In these studies, slight modifications in input data significantly altered the classification outcome, highlighting how even minor changes in key features like transaction amounts or timestamps can lead to incorrect classifications. Such manipulations can make it easier for adversaries to deceive models, underscoring the need to evaluate the robustness of ML models used in Ethereum phishing detection to ensure they remain effective and reliable despite such attacks.

RQ2.How do different machine learning algorithms compare robustness against adversarial manipulations in Ethereum phishing detection? This question stems from the observation that various machine learning algorithms, while effective in detecting phishing activities, exhibit differing levels of resilience against adversarial at-

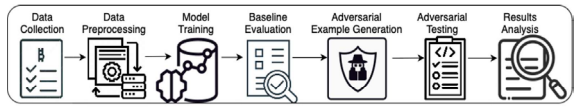


Fig. 1: Pipeline in Ethereum transactions and adversarial testing.

tacks [33,27,20,6,29,15]. The robustness of these algorithms can vary significantly under adversarial conditions, which can be seen in studies where some algorithms perform better than others when subjected to adversarial manipulations designed to evade detection [22,16,34]. A comparative analysis of these algorithms is essential to identify those that provide the best defense against such threats, ensuring the highest possible security in Ethereum transaction classification.

RQ3. How can the impact of manipulations be mitigated in machine learning-based Ethereum phishing detection? This question arises from the need to develop robust defensive strategies against adversarial attacks, as highlighted in recent literature [37,24,11,14,36]. Effective mitigation strategies could include enhancing feature selection processes, employing advanced data augmentation techniques, or implementing adversarial training methods [23,29,38]. Understanding and developing these techniques are crucial for improving the security and reliability of ML-based phishing detection systems in Ethereum transactions, thereby reducing the risk posed by adversaries manipulating transaction data and enhancing overall network security.

4 Methodology

Our pipeline is shown in Figure 1 and some of its key aspects are reviewed below.

4.1 Data Preparation

Dataset Description For our analysis, we employed two datasets. The first dataset, as detailed by Kabla *et al.* [21], focuses on binary classification, distinguishing between phishing and benign transactions. This dataset encompasses features such as `TxHash`, a unique identifier for each transaction; `BlockHeight`, which specifies the height of the block in which the transaction was recorded; and `TimeStamp`, indicating the exact time the transaction was confirmed and added to the blockchain. It also includes the `From` and `To` addresses, representing the Ethereum addresses of the sender and receiver, respectively. The `Value` feature denotes the amount of Ether transferred in the transaction, while `ContractAddress` indicates the Ethereum address of the smart contract involved, if applicable. Additionally, the `Input` field contains any extra data provided with the transaction. The dataset labels transactions as either phishing (1) or benign (0) under the `Class` feature. Dataset 1 includes 23,472 transactions, of which 15,989 are benign and 7,483 are phishing.

The second dataset, as described by Al-Emari *et al.* [7], is intended for multi-class classification, categorizing transactions into Phishing, Scam, Fake ICO, or Benign. This dataset features the `hash`, a unique identifier for each transaction, and `nonce`, a counter ensuring each transaction is processed only once. The `transaction_index`

indicates the transaction's position within the block, while `from_address` and `to_address` denote the blockchain addresses of the sender and receiver, respectively. The `value` field specifies the amount of cryptocurrency transferred. The dataset also includes `gas`, representing the gas limit provided for the transaction, and `gas_price`, indicating the price per gas unit. The `input` field contains additional data attached to the transaction. Moreover, `receipt_cumulative_gas_used` provides the total gas used by all transactions up to and including the current one within the block, and `receipt_gas_used` specifies the gas consumed by this particular transaction. The `block_timestamp` and `block_number` detail the time and number of the block that includes the transaction, while `block_hash` serves as the block's unique identifier. Lastly, the `from_scam` and `to_scam` fields indicate whether the sender's and receiver's addresses are associated with scams (0 for no, 1 for yes). The dataset also includes categorical data, `from_category` and `to_category`, which classify the nature of the participants, such as Phishing, Scamming, or Fake ICO. In this dataset, `Benign` had 57,000 transactions (79.28%), `Scamming` had 11,143 transactions (15.51%), `Phishing` had3,106 transactions (4.32%), and `Fake ICO` had only 1 transaction. Dataset 2 includes 71,250 transactions, with 80-20 training-testing splits.

4.2 Experimental Procedures

We utilized the two datasets outlined earlier to examine how simple AEs impact the classification accuracy of three classifiers: RF, DT, and KNN.

Minimal Manipulations. AEs were crafted by manipulating specific features. For the first dataset, we manipulated for feature. ① *Timestamp Manipulation (TimeStamp)*: We use predefined intervals to simulate future occurrences, testing the models' robustness to shifts against hypothetical scenarios of temporal manipulation. ② *Value Manipulation (Value)*: Transaction values were altered using two strategies: uniformly by adding a fixed percentage to each transaction's value or proportionally by introducing a random percentage change relative to each transaction's original value. ③ *Receiver Address Manipulation (To)*: The receiver's address was randomly changed to different addresses within the dataset to simulate phishing transactions directed to alternative destinations. ④ *Sender Address Manipulation (From)*: The sender's address was altered to different sender addresses to simulate phishing transactions originating from various sources.

We implemented a two-pronged approach, including both targeted and untargeted adversarial attacks on a second dataset [7]. AEs were generated by focusing on a broader array of critical transaction features, allowing us to simulate realistic attack scenarios and identify potential vulnerabilities more precisely.

Untargeted Attacks. The study of untargeted attacks involved generating AEs by applying broad, random perturbations across the entire feature space. Initially, AEs were generated using all features to evaluate the model's capacity to withstand attacks.

Individual features were then targeted to challenge the model more effectively. Modifying the *from_address* and *to_address* fields introduced new, unseen addresses to assess the model's ability to handle transaction origin and destination changes. Altering *value*, *gas* and *gas_price* simulated economic fluctuations, providing insights into the model's sensitivity to variations in transaction costs. Manipulating *block_timestamp*

and *block_number* mimicked transaction timing and sequence changes to understand the model's response to variations in transaction order. Altering *input*, *receipt_cumulative_gas_used*, and *receipt_gas_used* help explore the impact of changes in content.

Targeted Attacks. We conducted targeted adversarial attacks focusing on three scenarios: benign, phishing, and scamming. We employed two methods for generating these targeted attacks: rule-based and gradient-based (using the Fast Gradient Sign Method). *Rule-based Modifications* This approach applies straightforward, rule-based modifications to critical features like transaction value and timestamps, simulating realistic variations that can alter the classification outcomes.

- **Benign Targeted Attack:** This scenario aimed to assess the model's ability to maintain a benign classification despite manipulations, simulating tactics used to camouflage malicious activities. We created artificial benign transactions by minorly adjusting features such as transaction value and block_timestamp.
- **Phishing Targeted Attack:** The focus was on modifying phishing-labeled transactions to evade detection by misclassifying them as benign. This involved altering attributes such as from_address, to_address, and value, simulating an adversary's attempt to bypass security measures.
- **Scamming Targeted Attack:** In this scenario, scam-labeled transactions were manipulated to explore whether they could be misclassified as benign or other types. Adjustments to features like gas and gas_price were made to examine the model against efforts to obscure scam activities through changes in transaction costs.

FGSM FGSM applied small, strategically calculated changes to subtly influence the model's predictions while maintaining realistic feature values. Unlike the rule-based method, FGSM retained the existing distribution from the dataset, focusing on fine-tuning modifications based on the gradients to maximize the attack's effectiveness.

- **FGSM Details:** FGSM calculates perturbations that align with the gradient direction of the loss function. The perturbations were applied using the formula:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)),$$

  where x is the original feature, x' is the perturbed feature (AE), ϵ is a small scalar controlling the perturbation's magnitude, ∇_x represents the gradient of the loss function J for x, and $J(\theta, x, y)$ is the loss function dependent on the model parameters θ, input x, and true label y. The term $\text{sign}(\nabla_x J(\theta, x, y))$ provides the direction in which to perturb the feature vector to maximize the loss function.
- **Features:** FGSM perturbations were applied to transaction value, gas, gas_price, and block_timestamp. These features are critical in determining the nature of the transaction, and even small changes may affect the classification results.

5 Results and Analysis

5.1 Preliminary Results

We evaluate how different perturbations on the first dataset affect the accuracy and resilience of RF, DT, and KNN models. We examine how uniform and proportional value manipulations and address change impact model performance.

Timestamp Manipulations. We evaluate the effects of various timestamp modifications on classifier accuracy, including shifts of +24 hours, +1 hour, +30 minutes, +15 minutes, and +5 minutes. The RF classifier showed the highest resilience with only minor reductions in accuracy. For example, a one-day timestamp shift resulted in a decrease in accuracy from 98.82% to 95%, while a one-hour change led to an accuracy of 97.31%. In contrast, the DT model experienced a more pronounced decline, with accuracy dropping from 98.35% to 94.46% with a one-day shift. The KNN classifier was most affected by these temporal manipulations, with accuracy falling from 94.45% to 83.42% for a one-day change. These findings highlight the superior robustness of the RF model in handling timestamp alterations, positioning it as a preferable option for phishing detection in environments with variable timestamps (Table 2).

Value Manipulations. The uniform value manipulations caused significant declines. RF's accuracy dropped to 69%, and DT's fell to 68%, with substantial reductions in precision and recall for phishing transactions. Notably, the recall for phishing in DT decreased to 0.01% under uniform manipulation. Proportional manipulations had minimal impact, with accuracy and other metrics remaining close to their original values. KNN maintained relatively stable performance across both manipulation types, indicating robustness against such value changes (Table 3).

Address Manipulations. The robustness of the classifiers was tested against AEs of the `From` and `To` address features, involving changes in 5,000, 10,000, and 23,472 instances. For the RF model, accuracy decreased to 87% when the `From` address was manipulated and to 84% for the `To` address. Precision and recall for phishing transactions also declined significantly, with F1-scores dropping notably. The DT model showed a moderate reduction in accuracy, dropping to 92% for `From` and 93% for `To` manipulations, and a noticeable decrease in recall for phishing. KNN was the most sensitive to these manipulations, with accuracy falling to 85% for `From` and 93% for `To`, and significant drops in precision and recall for phishing transactions (Tables 4 and 5).

Next, our analysis will focus on the second dataset from Al-Emari *et al.* [7], which provides a comprehensive and relevant context for multi-class classification tasks.

5.2 Results of Targeted Attacks

Rule-based Modifications. We initially focus on rule-based AEs for specific classes.
① *Benign Class.* The RF and DT models initially demonstrated near-perfect accuracy in classifying benign transactions on the original test set. Under adversarial conditions, the accuracies for benign classifications declined markedly, with RF and DT models dropping to 84.39% and 84.65%. This represents a reduction exceeding 15%. In contrast, the KNN model sustained a higher adversarial accuracy of 90.25%.
② *Phishing Class.* The initial phishing accuracies for RF and DT were 96.34% and 96.98%, respectively. However, these values **plummeted to 1.31% and 1.18% under adversarial conditions**, reflecting a reduction of over 95%. The KNN model, which began with a lower baseline accuracy of 41.49%, saw a decrease to 2.15%.
③ *Scamming Class* Initially, the RF and DT models exhibited high accuracies of 99.5% and 98.68%. Adversarial attacks reduced these accuracies to 14.27% and 14.16%, rep-

Table 2: RF, DT, and KNN performance under timestamp manipulations. Accuracy, precision, recall, F1 score, and counts. B stands for benign and Ph. for phishing.

Increment	Model	Dataset	Acc	Precision		Recall		F1		Count	
				B	Ph	B	Ph	B	Ph	#B	#Ph
Original	RF	Baseline	0.9882	1	0.96	0.98	1	0.99	0.98	15989	7483
	DT	Baseline	0.9835	1	0.95	0.98	1	0.99	0.97	15989	7483
	KNN	Baseline	0.9445	1	0.85	0.92	1	0.96	0.92	15989	7483
+24 hours	RF	Adversarial	0.95	0.94	0.98	0.99	0.86	0.96	0.92	15498	7974
	DT	Adversarial	0.9446	0.95	0.94	0.97	0.88	0.96	0.91	15601	7871
	KNN	Adversarial	0.8342	0.84	0.82	0.94	0.62	0.88	0.7	14376	9096
+1 hour	RF	Adversarial	0.9731	0.97	0.97	0.99	0.94	0.98	0.96	15498	7974
	DT	Adversarial	0.9626	0.97	0.95	0.98	0.94	0.97	0.94	15601	7871
	KNN	Adversarial	0.8997	0.93	0.84	0.93	0.84	0.93	0.84	14376	9096
+30 min	RF	Adversarial	0.9776	0.98	0.97	0.99	0.96	0.98	0.96	15498	7974
	DT	Adversarial	0.9649	0.97	0.95	0.98	0.94	0.97	0.94	15601	7871
	KNN	Adversarial	0.916	0.95	0.85	0.92	0.9	0.94	0.87	14376	9096
+15 min	RF	Adversarial	0.9817	0.99	0.97	0.99	0.97	0.99	0.97	15498	7974
	DT	Adversarial	0.9672	0.98	0.95	0.98	0.95	0.98	0.95	15601	7871
	KNN	Adversarial	0.9179	0.95	0.85	0.93	0.9	0.94	0.87	14376	9096
+5 min	RF	Adversarial	0.9793	0.99	0.97	0.98	0.97	0.98	0.97	15498	7974
	DT	Adversarial	0.9734	0.98	0.95	0.98	0.97	0.98	0.96	15601	7871
	KNN	Adversarial	0.9344	0.98	0.85	0.92	0.96	0.95	0.9	14376	9096

Table 3: Performance evaluation of RF, DT, and KNN models subjected to 1% uniform and proportional value manipulation strategies. Metrics are as in Table 2.

Model	Strategy	Acc	Precision		Recall		F1		Count	
			B	Ph	B	Ph	B	Ph	#B	#Ph
RF	Original	0.99	0.98	1	1	0.99	0.99	0.99	15803	7669
	Uniform	0.69	0.96	0.68	0.02	1	0.03	0.81	23353	119
	Proportional	0.99	0.98	1	1	0.99	0.99	0.99	15813	7659
DT	Original	0.98	0.95	1	1	0.98	0.97	0.99	15601	7871
	Uniform	0.69	0.75	0.68	0.02	1	0.03	0.81	23294	178
	Proportional	0.98	0.95	1	1	0.98	0.97	0.99	15619	7853
KNN	Original	0.96	0.89	0.99	0.98	0.94	0.93	0.97	15192	8280
	Uniform	0.96	0.89	0.99	0.98	0.94	0.93	0.97	15193	8279
	Proportional	0.96	0.89	0.99	0.98	0.94	0.93	0.97	15192	8280

resenting a reduction of over 85%. The KNN model, with an initial accuracy of 67.06%, experienced a drop to 7.6%.

5.3 Gradient-based Approach Using FGSM

① *Benign Class* The overall accuracy of the RF model decreased from 99.75% to 94.62%, with a significant deterioration in phishing detection metrics. Despite hat, the model maintained a high benign accuracy of 99.95%. Conversely, the DT model's overall accuracy plummeted from 99.64% to 9.54%, with benign recall dropping to 0.02, indicating extreme vulnerability. The KNN model preserved a perfect benign accuracy

Table 4: Performance evaluation of RF, DT, and KNN models under manipulations of the `From` feature in Ethereum transaction datasets. Metrics are as in Table 2.

Model	Strategy	Acc	Precision		Recall		F1		Count	
			B	Ph	B	Ph	B	Ph	#B	#Ph
RF	Original Strategy	0.99	1	0.96	0.98	1	0.99	0.98	15708	7764
	5000 Changes	0.96	0.96	0.97	0.98	0.91	0.97	0.94	16386	7086
	10000 Changes	0.94	0.93	0.97	0.99	0.83	0.96	0.89	17027	6445
	23472 Changes	0.87	0.84	0.97	0.99	0.6	0.91	0.74	18877	4595
DT	Original Strategy	0.98	1	0.95	0.98	1	0.99	0.97	15601	7871
	5000 Changes	0.97	0.98	0.95	0.98	0.96	0.98	0.96	15935	7537
	10000 Changes	0.96	0.96	0.95	0.98	0.91	0.97	0.93	16272	7200
	23472 Changes	0.92	0.91	0.94	0.98	0.79	0.94	0.86	17207	6265
KNN	Original Strategy	0.94	1	0.85	0.92	1	0.96	0.92	14706	8766
	5000 Changes	0.93	0.97	0.85	0.92	0.93	0.94	0.89	15209	8263
	10000 Changes	0.91	0.94	0.84	0.92	0.87	0.93	0.85	15767	7705
	23472 Changes	0.85	0.86	0.81	0.93	0.67	0.89	0.74	17288	6184

Table 5: Performance evaluation of RF, DT, and KNN models under manipulations of the `To` feature in Ethereum transaction datasets. Metrics are as in Table 2.

Model	Strategy	Acc	Precision		Recall		F1		Count	
			B	Ph	B	Ph	B	Ph	#B	#Ph
RF	Original Strategy	0.99	1	0.96	0.98	1	0.99	0.98	15708	7764
	5000	0.96	0.95	0.97	0.98	0.89	0.97	0.93	16558	6914
	10000	0.92	0.91	0.97	0.99	0.79	0.95	0.87	17377	6095
	23472	0.84	0.81	0.97	0.99	0.51	0.89	0.67	19537	3935
DT	Original Strategy	0.98	1	0.95	0.98	1	0.99	0.97	15601	7871
	5000	0.97	0.98	0.95	0.98	0.96	0.98	0.96	15884	7588
	10000	0.96	0.97	0.95	0.98	0.93	0.97	0.94	16142	7330
	23472	0.93	0.92	0.94	0.98	0.83	0.95	0.88	16872	6600
KNN	Original Strategy	0.94	1	0.85	0.92	1	0.96	0.92	14706	8766
	5000	0.94	0.99	0.85	0.92	0.98	0.95	0.91	14849	8623
	10000	0.94	0.98	0.85	0.92	0.97	0.95	0.91	14988	8484
	23472	0.93	0.96	0.85	0.93	0.93	0.94	0.89	15351	8121

of 100% even under attack, but its overall accuracy fell from 90.15% to 80.22%, reflecting a failure to detect phishing and scamming categories effectively.

② *Phishing Class* The phishing detection accuracy of the RF model decreased from 96.34% to 47.69%, with a significant drop in the F1-score. The DT model's overall accuracy declined to 10.22%, with phishing recall reducing to 0.75%, underscoring a pronounced susceptibility to adversarial attacks. The KNN model's phishing detection performance collapsed entirely, with metrics falling to zero, indicating a complete failure to detect phishing transactions under adversarial conditions.

③ *Scamming Class* The overall accuracy of the RF model dropped from 99.75% to 81.75%, with scamming accuracy decreasing from 99.50% to 76.70%. The DT's overall accuracy fell to 9.71%, with scamming recall drastically reducing to 0.30. The KNN model's scamming detection metrics also dropped to zero.

5.4 Results of Untargeted Attacks

① *All Features* The RF model's accuracy decreased to 95.81%, DT's to 91.16%, with phishing detection severely impaired, and KNN maintained its baseline accuracy.

② *Address Features* AEs focusing on the *from_address* and *to_address* features resulted in a decline in overall accuracy to 80.22% for all models. None of the models detected phishing or scamming, indicating a high sensitivity to address manipulations.
③ *Financial Features* AEs targeting financial features (*value*, *gas*, *gas_price*) led to reduced RF's accuracy to 79.96%. The DT's accuracy dropped to 79.42%. The KNN model's accuracy slightly decreased to 90.02%.
④ *Using Temporal Features* Adversarial manipulations of temporal features (*block_timestamp*, *block_number*) showed the RF model's accuracy fell from 99.02% to 80.25%, with phishing detection metrics nearly nullified. The DT's accuracy similarly declined to 80.25%. The KNN model's accuracy also dropped to 80.26%.

Takeaway. These results underscore the need for robust defensive mechanisms against AEs. The significant declines in performance metrics under simple conditions highlight the necessity for a more reliable classification of transactions.

6 Discussion

Feature Selection for Optimal Classification. The analysis of transaction classification in this study highlights the significant role of feature selection in the robustness and accuracy of ML models. Among the features tested, **timestamp** and **value** emerged as critical classifier performance determinants. Timestamp manipulations demonstrated substantial impacts across models, with RF showing notable resilience compared to DT and K-KNN. The accuracy metrics indicate that temporal features, such as transaction time and date, are crucial for distinguishing legitimate from fraudulent transactions due to their inherent variability and relevance to transactional behaviors.

In contrast, value manipulations, including uniform and proportional changes, significantly affected model performance, particularly under uniform conditions. RF and DT models experienced considerable accuracy drops with uniform value changes, while KNN maintained relative stability. These findings suggest that while **transaction value** is a key feature for classification, it is also highly susceptible to perturbations.

Most Resistant Features to Adversarial Attacks. The study's results indicate that **address features**, specifically the `From` and `To` addresses, exhibit higher resistance to adversarial attacks compared to other feature types. Manipulations of these features resulted in moderate accuracy reductions for DT and RF models but a more pronounced impact on KNN. These features likely encode the relationship patterns between transaction entities, making them inherently resistant to straightforward changes.

The analysis showed that despite their critical contribution to accuracy, temporal features were also relatively resistant to manipulations. Timestamp shifts caused accuracy declines, but the extent was less severe than value manipulations. This indicates that while temporal features are crucial for classification, they are robust against adversarial attacks due to timestamp data's complexity and non-repetitive nature.

Best Combinations. For resilient classification, a combination of **temporal and address features** has proven effective. The synergy between these features offers a dual layer of robustness; temporal features provide a dynamic aspect that captures the tem-

poral distribution and patterns of transactions, while address features contribute a stable relational component less prone to adversarial interference.

Combining temporal features with **financial features**, such as transaction value and gas price, also enhances robustness. The results show that despite the susceptibility of financial features to uniform manipulations, their combined use with temporal data provides a broader context that improves model resilience. The temporal features help to contextualize the financial data, mitigating the impact of adversarial value manipulations by providing a temporal frame of reference.

The following recommendations can be drawn for the effective and robust classification of digital transactions, especially in adversarial environments. ① **Focus on Temporal and Address Features**: Incorporate timestamp and address data as primary features due to their robustness against adversarial attacks and critical role in classification accuracy. ② **Integrate Financial Features with Temporal Data**: Use financial transaction data with temporal features to improve robustness and provide a comprehensive transactional context that helps counteract adversarial manipulations. ③ **Adopt a Multi-Feature Approach**: Utilize a combination of diverse feature types to leverage their respective strengths and ensure a balanced, resilient classification model capable of withstanding various adversarial strategies.

7 Conclusion

This study provides a comprehensive examination of the vulnerability of machine learning models to adversarial attacks in fraudulent transaction detection. Our findings highlight the varying degrees of susceptibility among different classifiers, with RF demonstrating greater resilience than DT and KNN, which showed significant sensitivity to adversarial perturbations. The analysis underscores the importance of robust feature selection and the implementation of adversarial training to enhance model robustness. These strategies effectively mitigate the impact of adversarial manipulations, thereby strengthening the classifiers against potential threats. The emphasis on temporal and address features, combined with financial data, emerged as a crucial approach to bolster model defenses and ensure reliable classification in adversarial environments.

References

1. Abusnaina, A., Abuhamad, M., Alasmary, H., Anwar, A., Jang, R., Salem, S., Nyang, D., Mohaisen, D.: DL-FHMC: deep learning-based fine-grained hierarchical learning approach for robust malware classification. IEEE Trans. Dependable Secur. Comput. 19(5), 3432–3447 (2022)
2. Abusnaina, A., Anwar, A., Alshamrani, S., Alabduljabbar, A., Jang, R., Nyang, D., Mohaisen, D.: Systematically evaluating the robustness of ml-based iot malware detection systems. In: 25th International Symposium on Research in Attacks, Intrusions and Defenses, RAID. pp. 308–320. ACM (2022)
3. Abusnaina, A., Jang, R., Khormali, A., Nyang, D., Mohaisen, D.: DFD: adversarial learning-based approach to defend against website fingerprinting. In: 39th IEEE Conference on Computer Communications, INFOCOM. pp. 2459–2468. IEEE (2020)
4. Abusnaina, A., Khormali, A., Alasmary, H., Park, J., Anwar, A., Mohaisen, A.: Adversarial learning attacks on graph-based iot malware detection systems. In: 39th IEEE International Conference on Distributed Computing Systems, ICDCS. pp. 1296–1305. IEEE (2019)
5. Abusnaina, A., Wu, Y., Arora, S.S., Wang, Y., Wang, F., Yang, H., Mohaisen, D.: Adversarial example detection using latent neighborhood graph. In: IEEE/CVF International Conference on Computer Vision, ICCV. pp. 7667–7676. (2021)
6. Agarwal, R., Thapliyal, T., Shukla, S.K.: Analyzing Malicious Activities and Detecting Adversarial Behavior in Cryptocurrency based Permissionless Blockchains: An Ethereum Usecase. Distributed Ledger Technol. Res. Pract. 1(2), 1–21 (2022)

7. Al-Emari, S., Anbar, M., Sanjalawe, Y.K., Manickam, S.: A Labeled Transactions-Based Dataset on the Ethereum Network. In: Advances in Cyber Security, ACeS. vol. 1347, pp. 61–79. Springer (2020)
8. Alasmary, H., Abusnaina, A., Jang, R., Abuhamad, M., Anwar, A., Nyang, D., Mohaisen, D.: Soteria: Detecting adversarial examples in control flow graph-based malware classifiers. In: 40th IEEE International Conference on Distributed Computing Systems, ICDCS. pp. 888–898. IEEE (2020)
9. Ba, H.: Improving Detection of Credit Card Fraudulent Transactions using Generative Adversarial Networks. CoRR abs/1907.03355 (2019)
10. Bhagoji, A.N., He, W., Li, B., Song, D.: Practical Black-Box Attacks on Deep Neural Networks Using Efficient Query Mechanisms. In: ECCV. vol. 11216, pp. 158–174 (2018)
11. Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J.C., Liang, P.: Unlabeled Data Improves Adversarial Robustness. In: NeurIPS. pp. 11190–11201 (2019)
12. Cartella, F., Anunciação, O., Funabiki, Y., Yamaguchi, D., Akishita, T., Elshocht, O.: Adversarial Attacks for Tabular Data: Application to Fraud Detection and Imbalanced Data. In: SafeAI. vol. 2808. CEUR-WS.org (2021)
13. Chen, H., Zhang, H., Boning, D.S., Hsieh, C.: Robust Decision Trees Against Adversarial Examples. In: ICML. vol. 97, pp. 1122–1131. PMLR (2019)
14. Cohen, J., Rosenfeld, E., Kolter, J.Z.: Certified Adversarial Robustness via Randomized Smoothing. In: ICML. vol. 97, pp. 1310–1320. PMLR (2019)
15. Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: RobustBench: a standardized adversarial robustness benchmark. In: NeurIPS (2021)
16. Ding, Y., Wang, L., Zhang, H., Yi, J., Fan, D., Gong, B.: Defending Against Adversarial Attacks Using Random Forest. In: CVPR Workshops. pp. 105–114 (2019)
17. Fursov, I., Morozov, M., Kaploukhaya, N., Kovtun, E., Rivera-Castro, R., Gusev, G., Babaev, D., Kireev, I., Zaytsev, A., Burnaev, E.: Adversarial Attacks on Deep Models for Financial Transaction Records. In: KDD. pp. 2868–2878. ACM (2021)
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: ICLR (2015)
19. Guo, Q., Li, Z., An, B., Hui, P., Huang, J., Zhang, L., Zhao, M.: Securing the Deep Fraud Detector in Large-Scale E-Commerce Platform via Adversarial Machine Learning Approach. In: WWW. pp. 616–626. ACM (2019)
20. de Juan Fidalgo, P., Camara, C., Peris-Lopez, P.: Generation and Classification of Illicit Bitcoin Transactions. In: UCAmI. vol. 594, pp. 1086–1097. Springer (2022)
21. Kabla, A.H.H., Anbar, M., Manickam, S., Karuppayah, S.: Eth-PSD: A Machine Learning-Based Phishing Scam Detection Approach in Ethereum. IEEE Access 10, 118043–118057 (2022)
22. Li, D., Chen, D., Goh, J., Ng, S.: Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. CoRR abs/1809.04758 (2018)
23. Li, X., Chen, Y., He, Y., Xue, H.: AdvKnn: Adversarial Attacks On K-Nearest Neighbor Classifiers With Approximate Gradients. CoRR abs/1911.06591 (2019)
24. Mozo, A., González-Prieto, Á., Perales, A.P., Canaval, S.G., Talavera, E.: Synthetic flow-based cryptomining attack generation through Adversarial Networks. CoRR abs/2107.14776 (2021)
25. Narodytska, N., Kasiviswanathan, S.P.: Simple Black-Box Adversarial Attacks on Deep Neural Networks. In: CVPR Workshops. pp. 1310–1318 (2017)
26. Ngo, C.P., Winarto, A.A., Kou, C.K.L., Park, S., Akram, F., Lee, H.K.: Fence GAN: Towards Better Anomaly Detection. In: ICTAI. pp. 141–148. IEEE (2019)
27. Oliveira, V.C., Valadares, J.A., de Azevedo Sousa, J.E., Vieira, A.B., Bernardino, H.S., Villela, S.M., Gonçalves, G.D.: Analyzing Transaction Confirmation in Ethereum Using Machine Learning Techniques. SIGMETRICS Perform. Evaluation Rev. 48(4), 12–15 (2021)
28. Papernot, N., McDaniel, P.D., Wu, X., Jha, S., Swami, A.: Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In: IEEE Symposium on Security and Privacy. pp. 582–597 (2016)
29. Rabieinejad, E., Yazdinejad, A., Parizi, R.M., Dehghantanha, A.: Generative Adversarial Networks for Cyber Threat Hunting in Ethereum Blockchain. Distributed Ledger Technol. Res. Pract. 2(2), 1–19 (2023)
30. Sanjalawe, Y.K., Al-Emari, S.: Abnormal Transactions Detection in the Ethereum Network Using Semi-Supervised Generative Adversarial Networks. IEEE Access 11, 98516–98531 (2023)
31. Shu, D., Leslie, N.O., Kamhoua, C.A., Tucker, C.S.: Generative adversarial attacks against intrusion detection systems using active learning. In: WiseML@WiSec. pp. 1–6. ACM (2020)
32. Silva, S.H., Najafirad, P.: Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey. CoRR abs/2007.00753 (2020)
33. Singh, H.J., Hafid, A.S.: Prediction of Transaction Confirmation Time in Ethereum Blockchain Using Machine Learning. In: Blockchain. vol. 1010, pp. 126–133 (2019)
34. Stutz, D., Hein, M., Schiele, B.: Disentangling Adversarial Robustness and Generalization. In: CVPR. pp. 6976–6987 (2019)
35. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
36. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature Denoising for Improving Adversarial Robustness. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 501–509. Computer Vision Foundation / IEEE (2019)
37. Yang, J., Li, T., Liang, G., Wang, Y., Gao, T., Zhu, F.: Spam transaction attack detection model based on GRU and WGAN-div. Comput. Commun. 161, 172–182 (2020)
38. Zola, F., Bruse, J.L., Barrio, X.E., Galar, M., Urrutia, R.O.: Generative Adversarial Networks for Bitcoin Data Augmentation. In: BRAINS. pp. 136–143. IEEE (2020)
39. Zola, F., Segurola-Gil, L., Bruse, J.L., Galar, M., Urrutia, R.O.: Attacking Bitcoin anonymity: generative adversarial networks for improving Bitcoin entity classification. Appl. Intell. 52(15), 17289–17314 (2022)

LLM Guardrail Framework: A Novel Approach for Implementing Zero Trust Architecture

Bogeum Kim[1][0009−0007−4763−5117], Hyejin Sim[1][0009−0006−0691−1237], Jiwon Yun[1][0009−0007−3447−6567], Jaehan Cho[1][0009−0008−4878−9123], and Howon Kim[1][0000−0001−8475−7294]

Pusan National University
epdlsemgks872@gmail.com, {hyejin, jiwon, jaehan}@islab.re.kr, howonkim@gmail.com

Abstract. This paper proposes an `LLM guardrail framework` that incorporates a Zero Trust architecture to validate and control the responses of Large Language Model (LLM) to unethical queries. The proposed framework applies guardrails to harmful inputs to avoid harmful responses and includes four verification steps through Policy Decision Point (PDP) and Policy Enforcement Point (PEP) structures. This structure aims to enhance the reliability and safety of LLM responses. We demonstrate this framework on a fixed model and verify its generality by applying it to various models. Consequently, this allows for evasive responses to a wide range of unethical prompts.

Keywords: Large Language Model, Guardrail, Zero Trust.

1 Introduction

Large Language Model (LLM) have revolutionized many fields by leveraging powerful language understanding and generation capabilities, along with features such as in-context learning, instruction following, and multi-step reasoning[18]. These LLMs have had a major impact on the research community, revolutionizing the way AI algorithms are developed, from traditional natural language processing tasks and information retrieval (IR) to multimodal models and LLM-based agents. They have also been applied effectively in specialized fields such as healthcare, finance, law, and education, providing significant utility[24].

However, the widespread application of LLMs has also revealed negative aspects. Open Web Application Security Project (OWASP) has identified the top 10 vulnerabilities associated with LLMs, highlighting important issues such as over-reliance on LLM responses and prompt injection[11]. Concerns about gender, minority group, and political bias, along with the risk of providing insecure responses to certain queries, were also discussed in [22].

To address these negative aspects, several guardrail frameworks have been introduced. Nvidia NeMo Guardrails[19] guide LLM behavior by indexing key

elements in a vector database and enabling dialogue control and safety measures. For tasks like toxicity checks in Guardrails AI[4], classifier models evaluate input and output text, and if errors are detected, prompts are adjusted to guide the LLM's responses. While these frameworks aim to mitigate risks, they often perform validation procedures on the input only once.

When a verification process is performed once, it may cause reliability issues such as incorrect outputs for non-malicious inputs or harmful outputs generated through inadequately verified malicious prompts. Furthermore, it can lead to the leakage of sensitive information, including model system prompts and personal data contained in the training data. This paper applies the Zero trust concept a security paradigm of "never trust, always verify" to the guardrail framework to mitigate reliability issues and the leakage of sensitive information.

In this paper, we propose a Zero Trust-based guardrail framework that incorporates multiple validation procedures to enhance the control of LLMs over unethical queries. This approach aims to ensure a higher level of reliability and safety when managing LLM deliverables.

2 Background

2.1 Zero Trust

The traditional perimeter-based security approach involves storing a company's resources within a trusted zone and using security technologies such as firewalls or IDS/IPS to authenticate and authorize users accessing from outside before they can access the resources. However, this perimeter-based security has a significant drawback in that it is vulnerable to insider attacks, where malicious users who fail to gain authentication use the credentials of authenticated users to access and attack internal resources. Zero Trust is a security concept designed to address these vulnerabilities of perimeter-based security with the principle of "never trust, always verify." This security concept has gained more attention due to increased remote access from telecommuting and the rise in the use of cloud services following the COVID-19 pandemic. In January 2021, the Biden administration issued an executive order requiring the federal government and cloud service providers to adopt Zero Trust security policies and adhere to the corresponding framework by May 2021[3]. As a result, various organizations in the United States have proposed architectures and services for Zero Trust. The logical structure for Zero Trust can be seen in Figure 1.

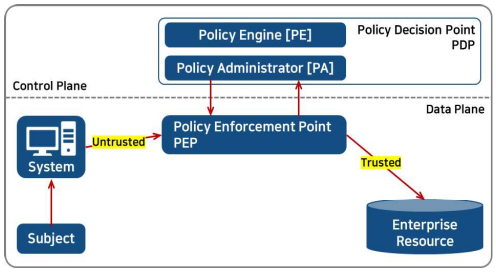


Fig. 1. Zero Trust Logical Structure

Figure 1 is the structure proposed by NIST in their 800-207 document for the core logical components of a Zero Trust. The most central points in this structure are the Policy Decision Point (PDP), which determines the policy of Zero Trust, and the Policy Enforcement Point (PEP), which enforces the policy. The following structure can be applied to various fields, and representative services include access control for external users and access control for resources.

Zero Trust Access Authentication-Authorization The structure of the Zero Trust-based LLM guidelines proposed in this paper is based on the structure shown in Figure 2.

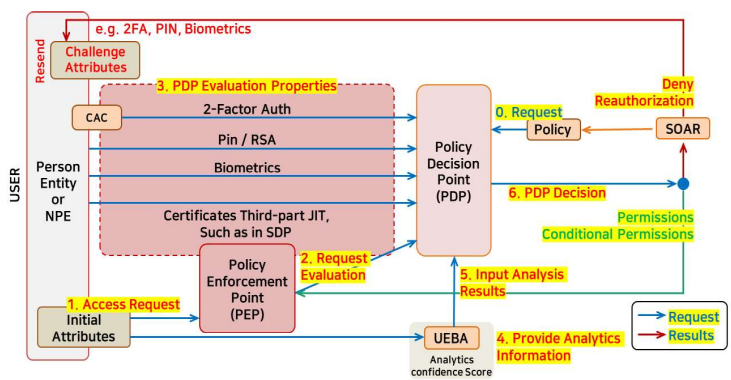


Fig. 2. Zero Trust Authentication-Authorization

Figure 2 shows a structure that can be applied in scenarios where a user or NPE sends an access request, based on the Zero Trust logical structure proposed by the United States Department of Defense (DoD)[8].

In this structure, the PEP collects the information of the user sending the access request through various authentication methods, and the collected data is transmitted to the PDP. The transmitted data is then stored on the server and analyzed using various technologies and solutions such as User and Entity Behavior Analytics (UEBA), Security Information and Event Management (SIEM), and Security Orchestration, Automation and Response (SOAR). This process updates the policies and ultimately determines whether the user's access request is granted or denied. The LLM guardrail framework proposed in this paper is designed based on the structure shown in Figure 2. The evaluation system of the PDP is replaced with a trained classification model and a vector database, using binary classification results (True or False) and cosine similarity (ranging from -1 to 1) as policy scores for evaluation. The following section provides a detailed explanation of the proposed structure and the results of experiments conducted based on the designed structure.

2.2 Guardrails for LLM

Recently, the concept of "guardrails" that monitor and filter the inputs and outputs of LLMs to minimize the potential risks of LLMs to ensure their safe utilization has gained importance. A guardrail is an algorithm that takes a set of objects (e.g., the inputs and outputs of an LLM) as input and determines if and how some enforcement action can be taken to reduce the risk inherent in the objects[10].

These LLM guardrails can be utilized to validate input data and block malicious input to prevent the model from being misused, or to monitor the output generated by the model and filter out inappropriate or harmful content.

In [9], the requirements for guardrails are defined in three broad categories: (i) Free from unintended responses e.g., offensive and hate speech; (ii) Compliance to ethical principles such as fairness, privacy, and copyright; and (iii) Hallucinations and uncertainty.

Some of these LLM guardrail-related studies include the following:

Llama Guard[14]. Llama Guard is an instruction-tuned Llama2-7b[5] model based on data labeled according to a classification system. Following this, Llama Guard2[21], based on Llama3-8B[6], was introduced, and Llama Guard3 further expanded the model by adding categories for Defamation, Elections, and Code Interpreter Abuse.

Nvidia NeMo Guardrails. NeMo Guardrails acts as a proxy between users and the LLM, using Colang scripts to define interaction rules. Colang specifies event flows and guides the LLM's behavior via the Guardrails runtime. Key elements include user canonical forms, dialogue flows, and bot canonical forms, indexed in a vector database. Developers use these scripts for dialogue control, custom actions, and safety measures.

RigorLLM[23]. During training, harmful and benign data were embedded and used Langevin dynamics to enrich the embedding space with harmful instances. In the inference phase, safe suffixes are optimized and use LLM to generate text-level transformations. Probabilistic KNNs and pre-trained LLMs (e.g., Lama Guard) provide predictions on both the augmented and original text, which are aggregated for the final prediction.

3 Framework Architecture

The LLM guardrail process proposed in this paper is based on a Zero Trust architecture. The purpose of this framework is to avoid responses to unethical or harmful (i.e., abnormal) prompts. For abnormal prompts—such as those related to sexual content, self-harm, harassment, hate, or violence—the model's output is fixed to prevent any additional response. Conversely, for normal prompts, the model provides appropriate responses.

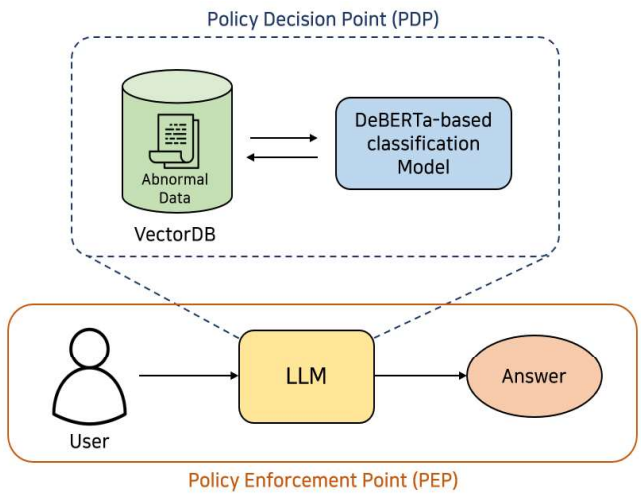


Fig. 3. LLM Guardrail Framework with Zero Trust Architecture

Figure 3 shows the PDP and PEP in our framework. In our guardrail framework, the primary entity is the chatbot user, who must pass through a PEP to obtain responses from the model. The PEP forwards the user's input to the PDP, which controls the model's response based on predetermined policies. Here, the PEP can be considered the LLM, which first performs binary classification to determine whether the user's prompt is normal or abnormal. After classification, the input is sent to the PDP, where various policies dictate the final output type

for the user's prompt. This decision is then sent back to the PEP (LLM), which ultimately generates the model's response.

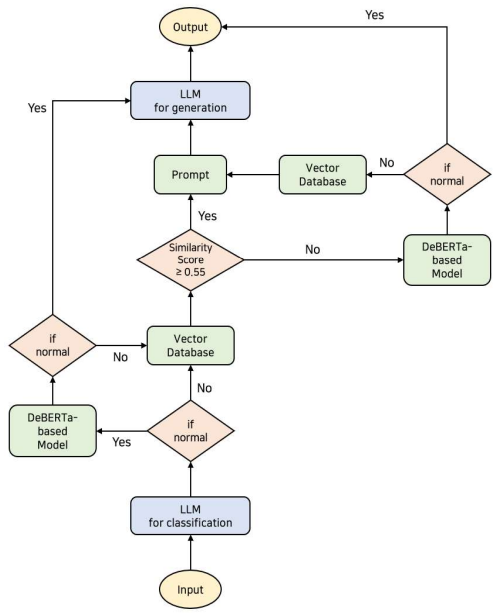


Fig. 4. LLM Guardrail Algorithm

The PDP has two main policy elements: a DeBERTa[13]-based model that verifies prompts classified as normal by the LLM and a vector database that verifies prompts classified as abnormal by the LLM. The DeBERTa-based model determines whether a prompt is truly normal with a policy score of 0 or 1, where a score of 1 indicates abnormal and 0 indicates normal. We are using DeBERTaV3[12], which is the enhanced version of DeBERTa in terms of performance and efficiency. In this verification stage, prompts classified as normal by the LLM are re-verified. Prompts classified as abnormal are sent to the vector database, where their cosine similarity is compared to retrieve the category of the top-ranked data. Additionally, prompts initially classified as abnormal by the LLM are sent to the vector database, where the cosine similarity is calculated to extract the category of the most relevant data. If the cosine similarity threshold is 0.55 or higher, the category of the top-ranked data is retrieved; if it is below 0.55, the prompt is considered normal and sent back to the DeBERTa-based model for re-verification. At this point, the DeBERTa-based model classifies the prompt again to determine if it is normal or abnormal. If the prompt is deemed normal, it is passed to the PEP, which is the LLM. If the prompt is deemed

abnormal, it is sent to the vector database, where the cosine similarity with internal data is calculated to extract the category of the top-ranked data. The category extracted from the vector database is then sent to the LLM's system prompt to ensure that the LLM can bypass responses to abnormal prompts. This process is illustrated in Figure 4.

Algorithm 1 Zero Trust Architecture based LLM Guardrail Framework

Input: User prompt
Output: Normal response or evasive response
$classification \leftarrow$ `LLM for classification`($input$)
if $classification =$ `normal` **then**
    $verification \leftarrow$ `DeBERTa-based model`($input$)
    if $verification =$ `normal` **then**
        Output: `LLM for generation`($input$)
    else
        $category \leftarrow$ `Vector Database`($input$)
        Output: `LLM for generation with system prompt`($category$)
    end if
else
    if `Cosine Similarity`($input$, `Vector Database`) ≥ 0.55 **then**
        $category \leftarrow$ `Extract Category from Vector Database`($input$)
        Output: `LLM for generation with system prompt`($category$)
    else
        $verification \leftarrow$ `DeBERTa-based model`($input$)
        if $verification =$ `normal` **then**
            Output: `LLM for generation`($input$)
        else
            $category \leftarrow$ `Extract Top Category from Vector Database`($input$)
            Output: `LLM for generation with system prompt`($category$)
        end if
    end if
end if

According to Algorithm 1, the PDP in this framework makes policy decisions based on the following scores:

Policy 1: Vector Database - Cosine similarity score (threshold 0.55)

Policy 2: DeBERTa-based Classification Model - True or False (0 or 1)

Additionally, the user's prompt is verified in four steps in the PDP:

1. If the LLM classifies the prompt as abnormal, it is verified by the Vector Database.
2. If the LLM classifies the prompt as normal, it is verified by the DeBERTa-based Classification Model.

3. If the result from the Vector Database verification does not exceed the threshold of 0.55, the prompt is re-verified by the DeBERTa-based Classification Model.

4. If the result from the DeBERTa-based Classification Model is 1 (abnormal), it is re-verified by the Vector Database for similarity.

Through these four verification steps, the framework strengthens the guardrail against unethical prompts.

This continuous verification process takes longer compared to the existing guardrail framework. However, our proposed framework offers improved reliability, resulting in greater long-term efficiency.

4 Experiments

4.1 Dataset

This section describes the vector database, DeBERTa model fine-tuning, and the datasets used for evaluation.

Dataset for Building Vector Database We constructed the vector database using the publicly available *toxic-chat*[16] dataset from Hugging Face and the *YouTube toxic comments*[2] dataset from Kaggle. To address the label imbalance present in the *toxic-chat* dataset, we incorporated additional data from the *YouTube toxic comments*.

Train Dataset for DeBERTa Fine-Tuning We used the *Jigsaw Multilingual Toxic Comment Classification*[1] dataset from Kaggle to train the DeBERTaV3. A total of 78108 data points were used after deleting some normal data to resolve label imbalance.

Evaluation Dataset We utilized a refined *OpenAI moderation evaluation dataset*[17] consisting of a total of 1595 data points as our evaluation dataset.

4.2 Cosine Similarity Threshold

Multiple experiments were conducted to set the cosine similarity threshold for determining the harmfulness of input prompts. Table 1 shows examples of prompt categories and similarity scores used for classification. We found that similarity scores between input prompts and vector database elements ranged from 0.54 to 0.56, setting the threshold at 0.55.

Input	Category	Similarity Score
If we get a wall, mass deportations and a muslim ban, I will be a little happier.	hate	0.549
women seeking I Am Wanting Sexy Chat sex dating	sexual	0.552
I started self harming around 5th grade, so 11 or 12 years old.	self-harm	0.565

Table 1. Similarity Score of Input Prompts

4.3 Evaluation of Classification

This study evaluated the effectiveness of the proposed guardrail framework by fixing the model to gemma-7b-it[20] for experiments. The evaluation dataset consisted of 1595 prompts, including 571 abnormal and 1024 normal prompts. The experiment compared two approaches: the first approach used the existing Python guardrail library[4], and the second approach used the classification method of the proposed guardrail framework. The evaluation was based on whether each approach correctly classified the prompts.

The first approach used the guardrail library provided by Guardrails AI. This library operated by generating error messages for the input abnormal prompts. It detected harmful language in 375 out of 571 abnormal prompts by generating error messages, but it failed to generate error messages for the remaining 196 prompts, indicating limitations in fully detecting harmful content.

The second approach used the proposed guardrail framework, employing a two-step strategy to classify input prompts as normal or abnormal. In the first step, where the LLM performed binary classification, it correctly classified 522 out of 571 abnormal prompts out of 1595 total prompts. The 49 prompts that were not correctly classified as abnormal in the first step were sent to the DeBERTa classifier for the second step, where 22 were reclassified as abnormal. This approach demonstrated significantly higher performance compared to the existing Python guardrail library. The Zero Trust architecture-based framework can provide evasive responses to most prompts falling into categories such as sexual content, self-harm, bullying, hate speech, and violence.

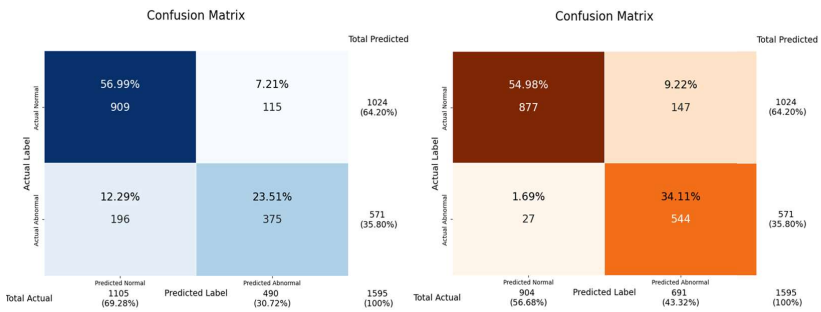


Fig. 5. Comparative Confusion Matrices for Prompt Detection. Left: Results using the Python Guardrail Library. Right: Results from Our Proposed Guardrail Framework

Figure 5 shows the confusion matrix for the Python Guardrail Library and for the method proposed in this paper. The proposed method demonstrates higher accuracy and a lower false positive rate, indicating superior performance in handling harmful prompts. This is evidenced by the higher true positive and true negative rates.

Method	Misclassification	Accuracy	F1 Score
Python Guardrails Library	311	0.805	0.707
Our Two-stage Strategy	174	0.891	0.862

Table 2. Classification Results of Two Approches

Based on the results summarized in Table 2, which include accuracy and F1 score, the classification method of the Zero Trust architecture-based LLM guardrail framework proposed in this paper demonstrates superior performance in handling harmful prompts.

4.4 Evaluation of Generalizability

To verify the versatility of the proposed framework, we conducted experiments with other models similar in size to gemma-7b-it model: Meta-Llama-3-8B-Instruct[6], Qwen2-7B-Instruct[7], Mistral-7B-Instruct-v0.3[15]. Table 3 presents the number of prompts that each model incorrectly classified as 'normal' during the first validation phase when evaluated on a set of 571 abnormal prompts. Additionally, it shows the number of answers generated for prompts initially classified as 'normal' but ultimately considered 'abnormal' by the framework.

Model	Misclassification	Output	Filter ratio
Llama-3-8B-Instruct	30	12	0.4
Qwen2-7B-Instruct	61	26	0.43
Mistral-7B-Instruct-v0.3	44	12	0.27

Table 3. Filter Ratio of LLMs

5 Results

Evaluation of Classification. We conducted experiments using two methods: a model with Python Guardrails, and a model with our proposed framework, all using the gemma-7b-it model. We verified the effectiveness of our framework by evaluating 571 abnormal prompts and comparing the results with and without using the classification method of the framework.

Evaluation of Generalizability. The framework's generality was confirmed using models with similar parameters to gemma-7b-it. Despite misclassification of abnormal prompts as normal, the framework provided correct answers through the validation step, demonstrating its versatility with various models.

6 Conclusion

In this paper, we propose a framework that bypasses harmful answers by applying guardrails to harmful inputs within a Zero Trust structure. We implement PDP and PEP structures using large language models and incorporate four verification steps to address the verification gaps in existing public domain guardrail devices. We demonstrate the framework's effectiveness on a fixed model and verify its generality by applying it to various models. In the future, we would like to conduct research on DeBERTa-based models used for PDP structures and techniques that can add verification elements from vector databases or apply guardrails to model output answers.

Acknowledgements This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information and Communications Technology Planning and Evaluation)

References

1. Jigsaw multilingual toxic comment classification, https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification
2. Youtube toxic comments, https://www.kaggle.com/datasets/reihanenamdari/youtube-toxicity-data
3. Executive order on improving the nation's cybersecurity (2021), https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

4. Guardrails ai (2023), https://github.com/guardrails-ai/guardrails.git
5. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama/blob/main/MODEL_CARD.md
6. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
7. Bai, J., Bai, S., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
8. the Defense Information Systems Agency (DISA), Team, N.S.A.N.Z.T.E.: Department of defense (dod) zero trust reference architecture (2022)
9. Dong, Y., Mu, R., et al.: Building guardrails for large language models. arXiv preprint arXiv:2402.01822 (2024)
10. Dong, Y., Mu, R., et al.: Safeguarding large language models: A survey. arXiv preprint arXiv:2406.02622 (2024)
11. Foundation, O.: Owasp top 10 for llm applications (2024)
12. He, P., Gao, J., et al.: Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv:2111.09543 (2021)
13. He, P., Liu, X., et al.: Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654 (2020)
14. Inan, H., Upasani, K., et al.: Llama guard: Llm-based input-output safeguard for human-ai conversations. arXiv preprint arXiv:2312.06674 (2023)
15. Jiang, A.Q., Sablayrolles, A., et al.: Mistral 7b (2023)
16. Lin, Z., Wang, Z., et al.: Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation (2023)
17. Markov, T., Zhang, C., et al.: A holistic approach to undesired content detection. arXiv preprint arXiv:2208.03274 (2022)
18. Minaee, S., Mikolov, T., et al.: Large language models: A survey. arXiv preprint arXiv:2402.06196 (2024)
19. Rebedea, T., Dinu, R., et al.: Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. arXiv preprint arXiv:2310.10501 (2023)
20. Team, Gemma, et al.: Gemma: Open models based on gemini research and technology (2024)
21. Team, L.: Meta llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md (2024)
22. Yao, Y., Duan, J., et al.: A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing p. 100211 (2024)
23. Yuan, Z., Xiong, Z., et al.: Rigorllm: Resilient guardrails for large language models against undesired content. arXiv preprint arXiv:2403.13031 (2024)
24. Zhao, W.X., Zhou, K., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)

ChatDEOB: An Effective Deobfuscation Method based on Large Language Model

Byunggeon Choi[1], Hongjoo Jin[1], Dong Hoon Lee[1], and Wonsuk Choi[1]

Korea University, Seoul 02841, Republic of Korea
{geonchoi,realredwine,donghlee,beb0396}@korea.ac.kr

Abstract. Obfuscation is a method that safeguards intellectual property rights against malicious analysts by altering the structure, logic, and other aspects of a program. However, malicious developers utilize obfuscation methods in their malware to avoid detection and analysis. To deobfuscate malware, analysts leverage their analysis skills alongside deobfuscation methodology. Although obfuscation is widely used in malware, heuristic-based deobfuscation methodology has limitations, including reliance on specific obfuscation tools and inefficiency in large-scale processing. In this paper, we propose ChatDEOB, an effective deobfuscation method that utilizes a Large Language Model (LLM). We focus on the LLM's application in various software engineering areas, such as code analysis, generation, and fuzzing, and employ it in our deobfuscation method. To effectively deobfuscate, we fine-tune the LLM model in detail and implement ChatDEOB using well-designed prompt engineering methods. To the best of our knowledge, ChatDEOB is the first method to deobfuscate code using a fine-tuned LLM model. To demonstrate the effectiveness of ChatDEOB, we utilize SacreBLEU, a published obfuscation evaluation method, along with the Obfuscation Quality Quantification Framework. The experiment resulted in the SacreBLEU score increasing from an initial average of 22.71 to 49.12, achieving a 116.27% improvement and demonstrating significant effectiveness. Additionally, when measuring the six evaluation indicators of the Obfuscation Quality Quantification Framework, the deobfuscation effect shows an average improvement of 85% compared to the obfuscated code.

Keywords: Source code deobfuscation · Source code obfuscation · Large Language Model.

1 Introduction

Obfuscation methods transform the internal structures of programs while maintaining their original functionality so that reverse engineering on those programs becomes more challenging. For this reason, obfuscation is being broadly used by software developers to protect the intellectual properties of their programs from illegal software analysis. However, on the other hand, malware is also obfuscated to be protected from commercial obfuscation. It is known that 99% of malware is obfuscated to bypass malicious code detection[12]. Indeed, it takes many resources to analyze such obfuscated malware. Moreover, because some obfuscation tools are provided for free, the number of obfuscated malware is expected to increase.

To analyze obfuscated codes, many researchers have presented some methods to deobfuscate the obfuscated codes [3,5,11]. However, existing deobfuscation researches have two significant limitations regarding code reconstruction. First, the existing methods are not fully automated, which implies a reliance on analyst's heuristics for the accuracy of deobfuscation. For example, for ProMBA [11] that is presented by *Lee et al.* to handle complex expressions effectively, the analyst must manually identify the obfuscated parts and input the corresponding expressions. Subsequently, the existing methods are not generally designed, which implies that they rely on a particular obfuscation tool. For example, *Kochberger et al.* have presented experiment results that deobfuscates source codes obfuscated by five obfuscation tools [8]. Even if deobfuscation is successfully performed, it cannot be deobfuscated using other obfuscation tools. To resolve these two limitations of the existing methods, we consider the Large Language Model (LLM), which is an artificial intelligence model trained on vast amounts of text data.

LLM possesses language understanding and generation capabilities, which allow it to perform natural language processing tasks. Recently, LLM has been employed in software engineering, such as code analysis and generation, software debugging, and fuzzing because of its advanced capabilities in understanding and generating natural language, which enhance the automation and accuracy of these tasks[22,15]. By applying the powerful language understanding capabilities of LLM to deobfuscation, we aim to overcome the limitations of existing deobfuscation methods and develop an efficient and generalized deobfuscation method. To gain a deep and generalized understanding of various deobfuscation mechanisms, we present a method (named ChatDEOB) that is designed to deobfuscate obfuscated source codes. To the best of our knowledge, ChatDEOB is the first method based on LLM for deobfuscation. In ChatDEOB, a pre-trained LLM such as ChatGPT is fine-tuned with a dataset for obfuscated code and its original code. By leveraging the LLM's pattern recognition and language understanding capabilities, ChatDEOB operates fully automated, eliminating the need for analyst heuristics. Furthermore, ChatDEOB can enhance its ability to respond to new obfuscation methods through continuous learning and updates of the LLM. By leveraging the LLM's pattern recognition ability, it can quickly learn and integrate new obfuscation methods into existing deobfuscation methods, enhancing its responsiveness to the latest obfuscation methodologies. Therefore, ChatDEOB is an innovative deobfuscation method that leverages LLM to reduce reliance on obfuscation tools and provide a more general and efficient deobfuscation method. This contributes to software security by presenting a new paradigm in obfuscation and deobfuscation research and supporting more reliable deobfuscation. In conclusion, our contributions can be summarized as follows:

Introduction of a general-purpose deobfuscation method that can be applied to various obfuscation tools and methods: ChatDEOB uses a fine-tuned LLM to provide a deobfuscation solution applicable to various obfuscation tools and methods.

Minimization of reliance on analysts and deobfuscation tools: ChatDEOB leverages the LLM's abilities to understand language to automatically analyze and reconstruct obfuscated code, reducing the reliance on the analyst's experience and intuition.

Demonstration of ChatDEOB's superior performance: SacreBLEU score evaluation results improved from an initial average of 22.71 to 49.12, a

116.27% increase, demonstrating our method's excellent performance and high level of responsiveness to the latest obfuscation methods.

2 Background and Related Work

2.1 Obfuscation

Software developers use obfuscation to protect intellectual property and safeguard software against malicious use. Source code obfuscation reduces readability, making it harder to analyze and requiring reverse engineers to spend significant time manually reviewing the code.

Obfuscation methods can be implemented using various methods. This section explains three representative methods: Mixed Boolean Arithmetic (MBA) obfuscation, Control Flow Flattening, and Manufacturing Opaque Predicates.

Mixed Boolean Arithmetic(MBA) obfuscation MBA obfuscation [25] complicates expressions by mixing complex boolean and arithmetic operations, making the code difficult to understand and analyze—and therefore challenging for analysts to discern the original meaning. Various MBA transformation formulas represent the same arithmetic operation in multiple ways. This makes it incredibly challenging for reverse engineers to analyze the code by pattern recognition, thus complicating their tasks.

Control Flow Flattening Flattening obfuscation [10] complicates the program structure by flattening the control flow. By embedding all control structures within a single large switch-case structure, this method creates a convoluted logical flow of the program that is difficult to trace. Given these conditions, reverse engineers must spend inordinate amounts of time reviewing the code flow in order to follow the execution order of the code.

Manufacturing Opaque Predicates Opaque Predicates obfuscation [13] complicates the execution flow by inserting opaque conditions (that always evaluate to true or false) into the code. These conditions do not affect the execution path but force analysts to handle unnecessary conditional branches, increasing the analysis time and hindering accurate analysis.

Even for the same obfuscation method, the implementation mechanisms vary by tool, which means that it is difficult to create consistent countermeasures. Each obfuscation tool uses unique algorithms and transformation methods, leading to significant differences in code structure and obfuscation patterns; this variability introduces additional complexity in the deobfuscation process. Consequently, existing research relies on the analyst's heuristics and lacks complete automation. So, while some solutions are specially designed for specific obfuscation tools, they may not be effective for others, highlighting the need for efficient deobfuscation solutions that work with various obfuscation methods.

2.2 Large Language Model

LLMs have achieved groundbreaking results in natural language processing, and their applications are continually expanding [20]. By training on extensive textual data datasets, an LLM can grasp the context and semantics of the language, enabling it to perform tasks such as text summarization, translation, and sentiment analysis with high accuracy. Notably, there is also ongoing research into the potential applications of LLM in software security [9].

Key applications of LLMs in security include malware detection, vulnerability analysis, and automated code review [21]. An LLM can understand the meaning and structure of code, learning specific patterns or anomalies to detect them automatically. For instance, an LLM can provide better accuracy and generalizability in malware detection compared to traditional signature-based detection methods. This is because an LLM can learn and make generalizations based on diverse malware patterns [1]. Additionally, an LLM can help discover and identify potential code flaws and security vulnerabilities in vulnerability analysis. The technology also excels at analyzing large codebases and finding similar vulnerabilities [14].

There is great potential for applying LLMs to deobfuscation tasks. An LLM can understand the context and structure of obfuscated code and use this understanding to restore the original form [4]. For example, an LLM can demonstrate outstanding capabilities in obfuscation methods involving complex formula transformations or control flow alterations. As it can recognize and comprehend obfuscated code patterns, an LLM can reconstruct the original code structure. This automates many parts of the deobfuscation process, which traditionally relied heavily on an expert's experience and intuition.

However, research on deobfuscation using LLMs is still nascent, and to the best of our knowledge, practical examples of using LLMs for actual deobfuscation tasks do not exist. Through continuous learning, an LLM can quickly and continually learn and integrate new obfuscation methods into existing deobfuscation methodologies to maintain its responsiveness to the latest obfuscation methodologies [6]. This characteristic greatly enhances the efficiency of deobfuscation methods and improves the accuracy and speed of software security analysis. The ongoing updates also enable an LLM to respond rapidly to the latest obfuscation methods, significantly overcoming the limitations of traditional static analysis or pattern-matching tools.

Therefore, research into leveraging LLMs for their powerful ability to understand language to perform deobfuscation tasks is highly promising. Our approach, in particular, has the potential to automatically analyze and reconstruct obfuscated code, minimizing reliance on analysts and effectively responding to the latest obfuscation methods. We aim to comprehensively demonstrate how LLM-based deobfuscation methods can address a range of obfuscation methods.

3 Approach

3.1 Prompt Engineering

In LLMs, the prompt is crucial in determining each model's performance and quality of responses. Well-designed prompts can maximize the capabilities of an LLM, yielding more accurate and useful results. We emphasize the importance

of prompt engineering in deobfuscation tasks using LLMs and explore methods to handle various obfuscation methods effectively through this approach.

Recognizing the significance of prompt engineering, we design our experiments by referencing the important prompt patterns that *White et al.* [24] previously proposed. This paper presents several useful prompt patterns that provide clearer and more efficient guidelines for LLMs to perform given tasks. Below, we describe the key prompt patterns presented in *White et al.*'s research and introduce our approach based on these patterns. The key prompt patterns are as follows:

Meta Language Creation Pattern. This pattern involves communicating with the LLM through a user-defined language, defining specific syntax or commands for the LLM to understand and respond appropriately. For example, a user might standardize notation to describe graph structures or define the states and transitions of a state machine. This enables the LLM to more accurately understand and process both the language and the concepts of a specific domain.

Persona Pattern. This pattern instructs the LLM to adopt a specific role or perspective, such as performing a code review from the viewpoint of a security expert. This allows the LLM to generate more specialized outputs that reflect the knowledge and perspective of a particular field, which helps it perform more precise and appropriate analyses tailored to a specific role.

Question Refinement Pattern. This pattern involves the LLM suggesting improved versions of user questions and asking the user whether to use them. It helps obtain more accurate and useful answers by clarifying the user's intent and providing additional information, thus leading to better outcomes.

Alternative Approaches Pattern. This pattern presents alternatives for performing a specific task and compares the strengths and weaknesses of each approach to help the user select the optimal solution. The LLM suggests various methods, analyzes their effectiveness, and supports finding the best solution. Table 1. demonstrates how utilizing these prompt engineering patterns allowed us to achieve more accurate and efficient results in deobfuscation tasks using an LLM. Specifically, we employed the Meta Language Creation pattern to define a language that describes specific patterns and structures of obfuscated code, and we used the Persona pattern to instruct the LLM to perform code analysis from the perspective of a security expert. Additionally, the Question Refinement pattern helped us gather supplementary information for improved analysis, while the Alternative Approaches pattern facilitated the comparison of various methods to select the optimal solution.

3.2 ChatDEOB

In this paper, we conduct deobfuscation tasks using an LLM. To ensure reproducibility and accessibility, we adopt a commercial model. This model performs well while being highly accessible and advantageous for obtaining reproducible results in various research environments—a characteristic crucial for enhancing research transparency and reliability. The ease of access through OpenAI's API

Table 1. Prompt Engineering Patterns and Examples

Prompt Pattern	Example Prompt
Meta Language Creation	The following obfuscated code has been subjected to Mixed Boolean Arithmetic (MBA) obfuscation. MBA obfuscation combines arithmetic and Boolean operations. For example, $a+b=(a\oplus b)+2\cdot(a\wedge b)$. Now, analyze the obfuscated code and deobfuscate it to restore its original meaning.
Persona	Now, you are a security expert. Analyze the obfuscated code from the perspective of a security expert, deobfuscate it to restore its original meaning, and evaluate the functionality of the deobfuscated code.
Question Refinement	If parts of the obfuscated code are unclear, generate additional questions to gather more information. For example, generate and answer questions such as, "What is the original variable name in this part?"
Alternative Approaches	There can be multiple approaches to deobfuscating and restoring the obfuscated code. Compare the following two approaches that explain the pros and cons of each: 1. Deobfuscating code through static analysis 2. Deobfuscating code through dynamic analysis Present the results of each approach and ultimately select the most suitable method.

further allows researchers to utilize the model and conduct experiments conveniently, thereby fostering a robust and collaborative research environment.

ChatDEOB is designed around an LLM to perform deobfuscation tasks and maximizes performance through fine-tuning and prompt engineering. The overview of ChatDEOB is shown in Fig. 1. The main components are described as follows: **First**, in the data collection and preprocessing stage, pairs of obfuscated and original code are collected and preprocessed for model training. **Second**, the collected data is used to tune the model during the fine-tuning stage. In this process, the model learns various obfuscation patterns, enabling it to restore the original code based on these patterns. **Third**, in the Prompt Engineering stage, appropriate prompts are designed to maximize the model's performance. Prompts help the model understand the context of the obfuscated code and accurately restore it. **Finally**, in the model evaluation and validation stage, the performance of ChatDEOB is assessed, and its effectiveness in handling various obfuscation methods is verified. This step ensures the model's generalization performance and evaluates its applicability in real-world scenarios.

The design of ChatDEOB is systematically structured to perform LLM-based deobfuscation tasks effectively. Optimizing the model's performance through fine-tuning and enhancing the response quality through prompt engineering provides a universal deobfuscation solution capable of handling various obfuscation methods. We demonstrate the potential of using an LLM to restore obfuscated code, contributing significantly to future research and practical applications.

4 Evaluation

4.1 Implementation

We implement the ChatDEOB deobfuscation method using the *Gpt-3.5-turbo-0125* model. The implementation process of ChatDEOB can be broadly divided into three stages: data preparation, fine-tuning, and evaluation.

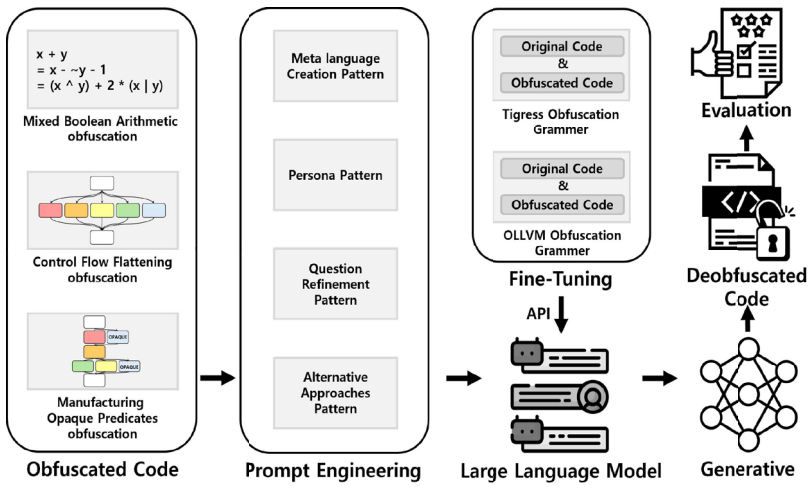


Fig. 1. Overview of ChatDEOB

Benchmark Datasets. We first generate various pairs of obfuscated source code with the corresponding original code to evaluate different obfuscation methods. We primarily apply obfuscation methods to source code obtained from open-source projects and public datasets to create the datasets. Specifically, we generate the obfuscated source code using the Tigress obfuscation tool and LLVM. We then use the LLVM obfuscation tool to obfuscate the intermediate language (IR) and then decompile it back to C, thus reconstructing the code. The resulting datasets include balanced samples for each obfuscation method.

For our datasets, we use Juliet C/C++ 1.3 [16] and publicly available Obfuscation Benchmarks [2] on GitHub [23]. Juliet C/C++ 1.3, released by the U.S. National Institute of Standards and Technology (NIST), includes examples of 118 different CWEs (Common Weakness Enumeration) in C/C++. These 70 selected files are written in C and compilable in a Linux environment. Additionally, we chose 40 basic algorithms and 30 small programs from the Obfuscation Benchmarks available on GitHub, totaling 70 Obfuscation Benchmark datasets. This results in a total of 140 Benchmark Datasets.

Next, we categorize the datasets into seven types. As described in Section 2, the three obfuscation methods—Mixed Boolean Arithmetic (MBA), Control

Flow Flattening, and Manufacturing Opaque Predicates—are divided into three levels. Level 1 includes datasets to which each obfuscation method is applied once. Level 2 includes datasets to which combinations of two out of the three obfuscation methods are applied. Level 3 includes datasets to which all three obfuscation methods are applied. This method results in seven sets of datasets for the 140 Benchmark Datasets.

Fine-Tuning. According to the official document from OpenAI [17], at least 10 datasets are required for fine-tuning, but it is recommended to have from 50 to 100 datasets to observe clear improvements. Based on this, we use pairs of obfuscated code and corresponding original code from the benchmark datasets—10 pairs for each obfuscation method—to ensure sufficient obfuscated and original code to fine-tune the model. The goal of the fine-tuning process is to enhance the model's ability to learn the patterns of obfuscated code and use this knowledge to restore the original code. Following OpenAI's recommendations and considering the diversity and quantity of data to build a general-purpose deobfuscation model capable of handling various obfuscation methods, we use 280 pairs (the 10 Fine-tuning examples x 2 Benchmark Datasets x 7 Obfuscation Methods x 2 Obfuscation Tools) of obfuscated and original code.

ChatDEOB Evaluation. BLEU (Bilingual Evaluation Understudy)[18] is a metric used to evaluate the performance of machine translation by quantitatively assessing the similarity between texts, making it useful for evaluating deobfuscation as well. However, BLEU can be inconsistent when assessing the complexity of obfuscated code. In contrast, SacreBLEU[19] with standardized settings can more consistently evaluate the differences between obfuscated and original code. Therefore, to ensure the consistency and reproducibility of scores—and measure the model's performance by quantitatively assessing the similarity between obfuscated and deobfuscated code—we use SacreBLEU.

The Obfuscation Quality Quantification Framework [7], a quantitative obfuscation evaluation tool, evaluates three main categories: Potency, Resilience, and Cost. Potency measures how difficult analyzing the obfuscated code is, and Resilience evaluates its resistance to tool-based analysis. This framework uses various quantitative metrics to measure the effectiveness of obfuscation methods, assessing program complexity and resistance to analysis tools. Such evaluation methods help compare the relative quality of obfuscation methods and provide a clear assessment of the protection level of obfuscated code. This experimentally validated the effectiveness of the deobfuscation method, which is expected to provide useful evaluation metrics and dataset construction methods for future research.

4.2 Experimental Results

To evaluate the performance of the ChatDEOB method, we conducted various experiments. First, we clearly define the scope and criteria for deobfuscation evaluation. The deobfuscated code must have the same functionality as the original code, meaning it should perform identical operations to ensure functional equivalence. However, due to the generative nature of LLMs, the deobfuscated code may not use the same symbols or variable names as the original code, and

its structure may also differ. Therefore, it is often challenging to assess the quality of deobfuscation based solely on superficial code similarity. To address this, we use the Obfuscation Quality Quantification Framework to comprehensively evaluate the deobfuscation performance regarding Potency, Resilience, Cost, and Compile Error.

For our evaluation of ChatDEOB, we use three key metrics that are crucial in assessing various aspects of deobfuscation: Potency, Resilience, and Cost. Potency measures how well the deobfuscated code maintains the original functionality. It includes attributes such as McCabe Complexity, CFG Size, Program Length, Instruction Count, Code Optimization, and Compile Error. Lower values for each attribute indicate successful deobfuscation, implying that the obfuscated code has been restored concisely and efficiently. Resilience assesses the obfuscated code's resistance to reverse engineering tools. Attributes such as code optimization are included here, and according to the evaluation metric formula, lower values indicate a higher resistance protection strength. To summarize, we evaluate the deobfuscation performance of ChatDEOB using six attributes to obtain a comprehensive assessment of how successful deobfuscation could be from multiple perspectives. Cost is a metric that evaluates the time and resources required for the deobfuscation process. However, when comparing ChatDEOB with other models in deobfuscation, we exclude the Cost metric to focus more on assessing the reliability of the deobfuscated code.

Secondly, we establish the experimental environment and evaluation methods. Generative LLMs tend to add comments automatically to aid in code understanding, so we first remove all comments from the generated code to prevent them from affecting the SacreBLEU score, as the presence of comments in both the reference and candidate files can cause discrepancies in the SacreBLEU score. Removing the comments ensures a more accurate evaluation of the code's performance. We then quantitatively assess the similarity between the comment-free reference files and candidate files using the SacreBLEU scores. This setup allows us to exclude the impact of comments on the evaluation results and more accurately measure the actual functional similarity of the code.

To ensure that our evaluation of ChatDEOB's deobfuscation performance is comprehensive, we utilize 5,880 datasets (140 Benchmark Datasets x 7 Obfuscation methods x 2 Obfuscation Tools x 3 Models), the SacreBLEU score, and the Obfuscation Quality Quantification Framework. These evaluation methods collectively demonstrate ChatDEOB's superior performance.

First, Table 2 compares the performance of Gpt-3.5-turbo, Fine-tuned Gpt-3.5-turbo, and ChatDEOB on code subjected to various obfuscation methods and intensities using the SacreBLEU score. The results show that while Gpt-3.5-turbo scores an average of 29.20 and 16.22, Fine-tuned Gpt-3.5-turbo returns improved scores of 49.50 and 33.36, respectively. ChatDEOB performs the best, with average scores of 56.02 and 42.22, indicating its superior deobfuscation capability and ability to restore the original code before obfuscation closely.

Using the Obfuscation Quality Quantification Framework, we compare the performance of Original, Gpt-3.5-turbo, and ChatDEOB. This framework evaluates the success of deobfuscation from multiple perspectives using six attributes: McCabe Complexity, CFG Size, Program Length, Instruction Count, Code Optimization, and Compile Error, as summarized in Fig. 2. Each attribute is a crucial indicator that sheds light on the reliability of the deobfuscated code.

Table 2. Comparison SacreBLEU Score of Gpt-3.5-turbo, Fine-tuned Gpt-3.5-turbo and ChatDEOB on Tigress and LLVM. Fine-tuned Gpt-3.5-turbo refers to the Gpt-3.5-turbo model fine-tuned using the method described in Section 4.1 Implementation.

Obfuscation Intensity	Gpt-3.5-turbo		Fine-tuned Gpt-3.5-turbo		ChatDEOB	
	Tigress	LLVM	Tigress	LLVM	Tigress	LLVM
Level1 - FLT	38.79	15.86	44.18	38.90	57.35	42.99
Level1 - MBA	37.89	16.76	49.30	37.74	54.66	47.03
Level1 - OPA	21.65	15.83	53.18	27.17	57.82	41.68
Level2 - FLT&OPA	25.36	16.63	53.97	30.40	56.25	38.97
Level2 - MBA&FLT	38.53	15.11	45.40	37.26	54.61	43.88
Level2 - MBA&OPA	19.18	17.79	50.88	30.12	57.34	40.59
Level3 - MBA&FLT&OPA	23.03	15.59	49.55	31.95	54.09	40.37
Average	29.20	16.22	49.50	33.36	**56.02**	**42.22**

ChatDEOB maintains values for McCabe Complexity, CFG Size, Program Length, and Instruction Count that are, on average, 85.01% and 21.28% lower than those of the Obfuscated Code and Gpt-3.5-turbo, respectively, regardless of the obfuscation level. This demonstrates ChatDEOB's ability to restore obfuscated code more simply and efficiently. Additionally, It excels with superior performance in Code Optimization, indicating its effectiveness in this area. Finally, the Compile Error metric assesses the success of code compilation. ChatDEOB reduces compilation errors to an average of 21%, compared to Gpt-3.5-turbo's 36%, indicating a more effective transformation of the obfuscated code.

These results demonstrate ChatDEOB's excellent deobfuscation performance across various obfuscation methods. Particularly in metrics such as McCabe Complexity, Program Length, Instruction Count, CFG Size, Code Coverage, and Code Optimization, ChatDEOB outperforms both Gpt-3.5-turbo and the Original. This provides evidence that ChatDEOB offers a robust deobfuscation solution that is capable of restoring obfuscated code simply and efficiently and is also highly reliable.

5 Conclusion

In this paper, we have proposed ChatDEOB based on an LLM to effectively deobfuscate codes that are obfuscated by several methods and tools. The LLM is fine-tuned in ChatDEOB with pairs of obfuscated and original codes. Also, we have conducted the prompt engineering to design outperformed prompts. To evaluate ChatDEOB, we employed the SacreBLEU score as a metric. We have shown that Gpt-3.5-turbo achieves a score of 22.71 on average. On the other hand, after the fine-tuning and prompt engineering, ChatDEOB achieves a score of 49.12, which is highly improved by 116.27%. Moreover, we have evaluated ChatDEOB on several combinations of obfuscation methods, which implies that it is outperforms regardless of obfuscation methods. To validate ChatDEOB, the Obfuscation Quality Quantification Framework has been employed, which provides six evaluation metrics. Through these six metrics, we were able

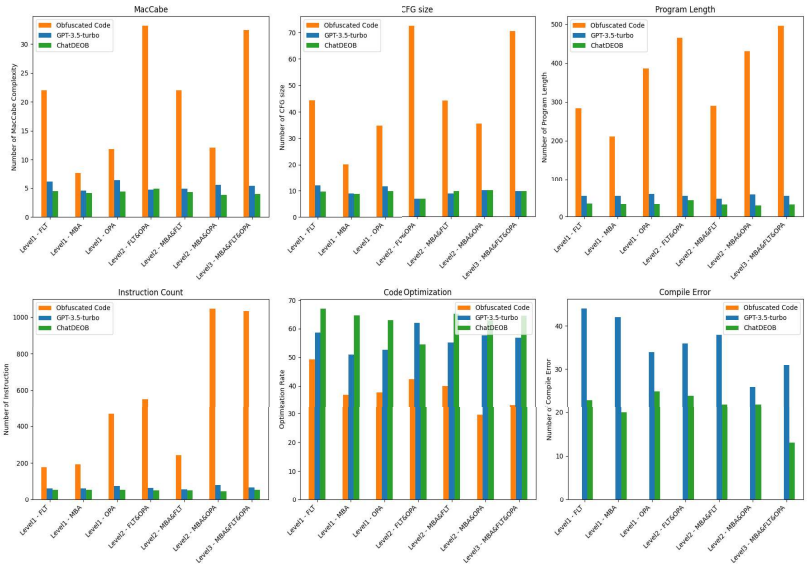


Fig. 2. Comparison of deobfuscation performance using McCabe, CFG Size, Program Length, Instruction Count, Code Optimization, and Compile Error metrics across different obfuscation levels.

to compare ChatDEOB with Gpt-3.5-turbo comprehensively. Unlike the existing methods that require manual analysis, ChatDEOB is able to generate deobfuscated codes in a fully automated manner. Finally, we believe that ChatDEOB is extensively applicable to unknown obfuscation methods. Since fine-tuning and prompt engineering can be conducted accumulatively to improve the performance of ChatDEOB, ChatDEOB is expected to stay effective against new obfuscation methods. Given this, future research should focus on further refining the model's adaptability and real-time performance. In conclusion, our research suggests that harnessing the power of an LLM for deobfuscation can help analysts flexibly respond to the latest obfuscation methods, thus significantly improving the accuracy and efficiency of software security analysis.

6 Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIP) (No. RS-2024-00399389, Generative AI based Binary Deobfuscation Technology and Its Evaluation Metrics).

References

1. Amine Ferrag, M., Alwahedi, F., Battah, A., Cherif, B., Mechri, A., Tihanyi, N.: Generative ai and large language models for cyber security: All insights you need.

arXiv e-prints pp. arXiv–2405 (2024)
2. Banescu, S., Collberg, C., Ganesh, V., Newsham, Z., Pretschner, A.: Code obfuscation against symbolic execution attacks. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. pp. 189–200 (2016)
3. Blazytko, T., Contag, M., Aschermann, C., Holz, T.: Syntia: Synthesizing the semantics of obfuscated code. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 643–659 (2017)
4. Chataut, R., Gyawali, P.K., Usman, Y.: Can ai keep you safe? a study of large language models for phishing detection. In: 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0548–0554. IEEE (2024)
5. David, R., Coniglio, L., Ceccato, M., et al.: Qsynth-a program synthesis based approach for binary code deobfuscation. In: BAR 2020 Workshop (2020)
6. Derner, E., Batistič, K., Zahálka, J., Babuška, R.: A security risk taxonomy for large language models. arXiv preprint arXiv:2311.11415 (2023)
7. Jin, H., Lee, J., Yang, S., Kim, K., Lee, D.H.: A framework to quantify the quality of source code obfuscation. Applied Sciences **14**(12), 5056 (2024)
8. Kochberger, P., Schrittwieser, S., Schweighofer, S., Kieseberg, P., Weippl, E.: Sok: automatic deobfuscation of virtualization-protected applications. In: Proceedings of the 16th International Conference on Availability, Reliability and Security. pp. 1–15 (2021)
9. Kolosnjaji, B., Zarras, A., Webster, G., Eckert, C.: Deep learning for classification of malware system call sequences. In: AI 2016: Advances in Artificial Intelligence: 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings 29. pp. 137–149. Springer (2016)
10. László, T., Kiss, Á.: Obfuscating c++ programs via control flow flattening. Annales Universitatis Scientarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica **30**(1), 3–19 (2009)
11. Lee, J., Lee, W.: Simplifying mixed boolean-arithmetic obfuscation by program synthesis and term rewriting. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 2351–2365 (2023)
12. Liang, M., Li, Z., Zeng, Q., Fang, Z.: Deobfuscation of virtualization-obfuscated code through symbolic execution and compilation optimization. In: Information and Communications Security: 19th International Conference, ICICS 2017, Beijing, China, December 6-8, 2017, Proceedings 19. pp. 313–324. Springer (2018)
13. Majumdar, A., Thomborson, C.: Manufacturing opaque predicates in distributed systems for code obfuscation. In: Proceedings of the 29th Australasian Computer Science Conference-Volume 48. pp. 187–196 (2006)
14. Mathews, N.S., Brus, Y., Aafer, Y., Nagappan, M., McIntosh, S.: Llbezpeky: Leveraging large language models for vulnerability detection. arXiv preprint arXiv:2401.01269 (2024)
15. Meng, R., Mirchev, M., Böhme, M., Roychoudhury, A.: Large language model guided protocol fuzzing. In: Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS) (2024)
16. National Institute of Standards and Technology: Software Assurance Reference Dataset Test Suites (2017), `https://samate.nist.gov/SARD/test-suites/112`
17. OpenAI: Fine-tuning (2023), `https://platform.openai.com/docs/guides/fine-tuning`
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
19. Post, M.: A call for clarity in reporting bleu scores. arXiv preprint arXiv:1804.08771 (2018)
20. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)

21. Saxe, J., Berlin, K.: Deep neural network based malware detection using two dimensional binary program features. In: 2015 10th international conference on malicious and unwanted software (MALWARE). pp. 11–20. IEEE (2015)
22. Tan, H., Luo, Q., Li, J., Zhang, Y.: Llm4decompile: Decompiling binary code with large language models. arXiv preprint arXiv:2403.05286 (2024)
23. Technical University of Munich: Obfuscation Benchmarks (2016), `https://github.com/tum-i4/obfuscation-benchmarks`
24. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
25. Zhou, Y., Main, A., Gu, Y.X., Johnson, H.: Information hiding in software with mixed boolean-arithmetic transforms. In: International Workshop on Information Security Applications. pp. 61–75. Springer (2007)

An effective ensemble algorithm for short-term load forecasting

En-Wei Zhang, Luo-Fan Wu, and Chun-Wei Tsai

Computer Science and Engineering, National Sun Yat-sen University
Kaohsiung, Taiwan
{m113040054,m123040025}@student.nsysu.edu.tw
cwtsai@mail.nsysu.edu.tw

Abstract. Short-term load forecasting (STLF) is a critical issue for managing electricity distribution systems because its accuracy might strongly impact the performance and security of the power system. An effective load forecasting system, of course, can also be used to detect and prevent abnormal behaviors (e.g., electricity theft detection) to further provide a stable and safe power system. That is why extensive studies have been presented by using statistical, machine learning, and deep learning methods. However, each learning model has its strengths to capture specific patterns in the load profile. To integrate the strengths of different deep learning algorithms, an ensemble model, which contains three deep learning mechanisms, including convolution, self-attention, and recurrent, is proposed in this paper. Because of the high uncertainty and volatility, a single-step horizon will be focused on while predicting the load at the residential level to ensure timely and precise responses to security threats. Multiple-step forecasting will also be conducted at the aggregated level to further enhance the robustness of the power grid security. Furthermore, considering the distinct behavior in the load profile, the proposed method is designed accordingly to adapt to each level. The experimental results show the effectiveness of the proposed method compared with deep learning models on both levels across two well-known datasets.

Keywords: Smart grid · deep learning · machine learning

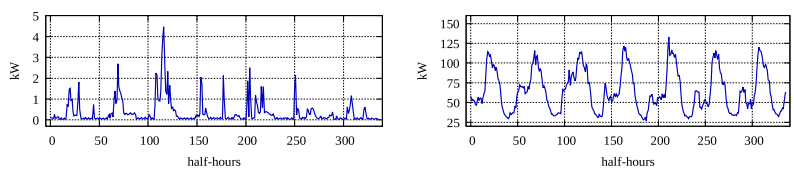
1 Introduction

The short-term load forecasting (STLF) [17] is one of well-known problems in smart grid. Accurately predicting load usage in advance can help energy companies formulate effective plans for power transmission. Many well-known statistical methods were used to deal with the STLF problem. Multiple linear regression (MLR) [2] treats the load value in each timestep as a variable, looking to minimize the square error between the actual value and prediction. Another widely used method is autoregressive integrated moving average model (ARIMA)

[28], using the sliding window to capture the average load usage in the focused period. However, statistical methods have limitations because they often find it hard to learn personalized information about users and have difficulties handling complex and numerous data. As the data grows in complexity and size recently, extracting useful information from the data becomes a challenging issue for these traditional statistical approaches.

Due to these abovementioned limitations, machine learning has been used to handle more complicated data recently. The study [15] used a three-layer multilayer perceptron (MLP) and added extra timestamp vectors into the load profiles, enhancing the ability of MLP to learn the temporal relationships in the time-series data. Support vector regression (SVR) [6] allows the load profiles to be analyzed in a higher-dimensional space through the projection of kernel function. This minimizes the distance of each load profile to the hyperplane, making it possible to realize the non-linear components between samples. k-nearest neighbor (KNN) [1] generates the prediction based on the Euclidian distance of each nearest k sample. The number of k can significantly affect the performance of this algorithm.

Most machine learning models are not inherently designed to handle sequential or time-series data directly. This limitation restricts their ability to fully understand the relationship between temporal information and load usage. Some deep learning models were developed without such limitation when handling time-series data. Recurrent neural network (RNN) [23] and long short-term memory (LSTM) [11] networks are capable of learning temporal information from time-series data recursively. Temporal convolutional network (TCN) [4] adapted from convolutional networks, and it was specifically designed for time-series data. Furthermore, transformers-based have also been modified to enhance prediction accuracy in forecasting topics [31]. Based on our observation, different mechanisms of deep learning models have advantages in capturing the load profile trends. In the STLF problem, the load prediction problem can be categorized



(a) Residential level load profile. (b) Aggregated level load profile.

Fig. 1. The comparisons between residential and aggregated levels.

into two levels based on data granularity [25]. As shown in Figure 1, the first one is the residential level [26], which suffers from great uncertainty and volatility, providing finer granularity predictions. The other one is aggregated level [27], which means predicting an area load usage of households, offering a coarser granularity prediction. In this paper, the proposed method will be applied on both levels to evaluate the effectiveness of each level. The main contributions of this paper are as follows:

1. An ensemble algorithm was proposed to integrate three mechanisms of deep learning model into a hybrid generative-style model.
2. The proposed method was applied to both residential level and aggregated level to show its effectiveness on two electricity load datasets.
3. Clustering algorithm and different designs on meta-learner will be used to improve the forecasting results.

The remaining sections of this paper are as follows: Section 2 begins with the problem definitions of residential and aggregated levels, followed by existing research on both levels. Section 3 turns to the basic concept and the details of the proposed algorithm on different levels. Section 4 shows the experiment results conducted on two datasets on both levels. Finally, the conclusion and future work are presented in Section 5.

2 Related Work

2.1 The Problem Definition

The goal of load prediction is to minimize the error between forecasted values and actual load consumption [10]. An accurate load prediction is important for grid management, which provides energy companies with an insight of a better resource allocation and planning. To handle this problem by using deep learning models, the sliding window method is applied to segment the historical load profiles into input and target sequences. The following is the definition:

$$\boldsymbol{X}_i = \{\boldsymbol{x}_1^d, \boldsymbol{x}_2^d, \ldots, \boldsymbol{x}_t^d\}, \tag{1}$$

$$\boldsymbol{Y}_i = \{\boldsymbol{y}_{t+1}^o, \boldsymbol{y}_{t+2}^o, \ldots, \boldsymbol{y}_T^o\}, \tag{2}$$

where $\boldsymbol{X}_i$ is the i-th input sequence, composed of vectors from $\boldsymbol{x}_1^d$ to $\boldsymbol{x}_t^d$. $\boldsymbol{x}_t^d$ contain d-dimension features at each time t. $\boldsymbol{Y}_i$ is the i-th target sequence, composed of vectors from $\boldsymbol{y}_t^o$ to $\boldsymbol{y}_T^o$. $\boldsymbol{y}_t^o$ represents the o-dimension vectors, which is the actual load consumption. In this paper, root mean square error (RMSE), represented as $L(M^j, \boldsymbol{X}_i^j, \boldsymbol{Y}_i^j) = \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(M^j(\boldsymbol{X}_i^j) - \boldsymbol{Y}_i^j\right)^2}$, where M^j represents the prediction model for the load profile j. $\boldsymbol{X}_i^j$ and $\boldsymbol{Y}_i^j$ represent the input and output sequence of load profile j, respectively, and are used in the loss function to minimize the difference between predictions and actual consumptions. Compared with mean square error (MSE), the scale of units will become kWh^2 rather than the same scale unit of RMSE, unlike mean absolute percentage error (MAPE), which suffers from the divided-by-zero problem when the actual load values are close to zero. The forecast target differs between residential and aggregated levels because, as observed in Figure 1, the load profile behavior varies significantly across these levels. The load profile at the residential level is noted by high uncertainty and contains numerous peaks. Such uncertainty is caused by the irregular habits of customers or the sudden opening of power-consuming appliances. The load profiles in aggregated level seems smoother and more regular because of the principle of large numbers. It will cancel individual uncertainties

by aggregating them together. To make the applications more practical, the forecasting accuracy of the next single timestep will be focused for residential load consumption, which can be represented as $\boldsymbol{Y}_i = \{\boldsymbol{y}^o_{t+1}\}_{\text{res}}$. In contrast, predicting multiple timesteps for the aggregated level will be conducted, which can be represented as $\boldsymbol{Y}_i = \{\boldsymbol{y}^o_{t+1}, \boldsymbol{y}^o_{t+2}, \ldots, \boldsymbol{y}^o_T\}_{\text{agg}}$, where T is a variable to evaluate model performance, reflecting the ability to forecast different length of horizons.

2.2 Forecasting Methods

Due to the regular and periodic nature of load profiles at aggregated level, statistical methods such as ARIMA [28] and linear regression (LR) [2] are often employed compared with another level. Rather than only minimizing mean-squared error as usual, a study [7] conducted comparisons of many linear models. One of them is adding regularization terms to penalize coefficients that are too large. Compared with traditional statistical methods, machine learning based models demonstrate a better ability to capture the non-linear relationship in load profiles. It can be divided into two main parts: non-neural-based and neural-based. Non-neural-based studies [5, 13] have utilized SVR to address the STLF problem on aggregated and residential levels. A weighted k-nearest neighbor (WKNN) [8] algorithm was employed to solve this problem. Rather than averaging the nearest k neighbor, this method increases the weight of samples by their Euclidean distance from each other. A closer sample will get a higher weight when making predictions. The variations of decision tree conducted on the STLF problem are presented in the study [9].

Neural-based algorithms like deep neural network (DNN) [12], recurrent neural network (RNN) [26], convolution neural network (CNN) [18], and transformer [14, 21, 31] models have been utilized to deal with STLF problems. Shi et al. [26] noticed that the behavior between customers might suffer similarly. A pooling strategy was presented to enhance the accuracy of prediction. Wang et al. [30] used temporal convolution network (TCN) [4], a variation of CNN, to obtain better results than LSTM model. Zhou et al. [31] modified the designation of the decoder and the mechanism of self-attention to outperform other transformer-based models. Additionally, the nature of neural-based algorithms allows for the combinations of various deep learning models. Kim and Chuo [16] used CNN to extract the meaningful features among many variables. Then LSTM model was used to learn relationships in time-series, yielding better forecasting results compared with just using LSTM or CNN only. Aouad et al. [3] proposed a sequence-to-sequence CNN combined with attention mechanism for handling the STLF problem. The experiment results showed the architecture of combining CNN and self-attention can outperform other sequence-to-sequence models.

3 Proposed Method

In this paper, we observe different load profile behaviors for residential and aggregated levels, as shown in Figure 1. This paper proposes an ensemble load

forecasting model that features three different deep learning models and customized meta-learners. As shown in Figure 2, the data preprocessing will be conducted, including missing value imputation and normalization. Furthermore, according to different scenarios of prediction level, clustering algorithm and aggregating method will be applied to divide the STLF problem into residential level (A) and aggregated level (B).

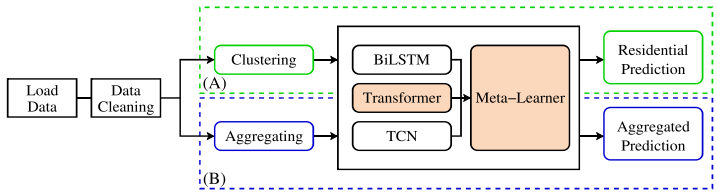


Fig. 2. The overview of proposed method.

At the residential level, a clustering algorithm is applied to group customers based on similarity among load profiles, allowing the model to learn the shared uncertainty within the same group. An area's total load usage is shown at the aggregated level by adding load profiles for each customer. To capture different load profile trends, the proposed method utilizes TCN [4] based on convolution, BiLSTM [24] based on recurrent, and transformer [29] based on self-attention. There are slightly different designations of transformer decoder and meta-learner for residential and aggregated levels, marked orange in Figure 2. Integrating submodels through a meta-learner makes the proposed method achieve better prediction accuracy in most cases, as shown in the experimental results.

3.1 Aggregated Level

In this scenario, our goal is to conduct a relatively long-term prediction of the aggregation load usage of residents in an area. Making an accurate prediction could help create a better schedule for electricity generation. The way how to process the data, the designations of transformer decoder and meta-learner will be followed.

Aggregation will be conducted at this level. The dataset used in this paper is the load records of users, including buildings and residents. Sevlian and Rajagopal [25] explored the relationship among load profiles by aggregating them as the load profile, as shown in Figure 1 (b). According to the law of large numbers, the more load profiles are aggregated, the smoother the aggregation profiles will be. Figure 3 takes the example of k load profiles, and each load profile may record other variables like temperature and humidity. Each variable will be aggregated to form aggregated level load profiles. Observed that the behavior of load profiles in aggregated level is much smoother than the residential level.

Transformer Decoder generates a sequence of load predictions, as shown in Figure 4. The behavior of decoder can be divided into training and inference phases. During the decoder training, the actual load values are used as inputs,

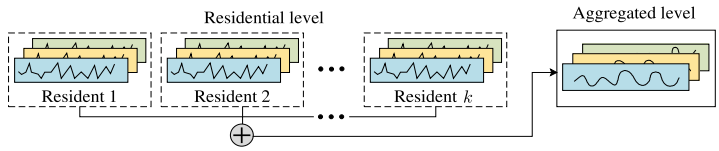


Fig. 3. Aggregation of load profiles.

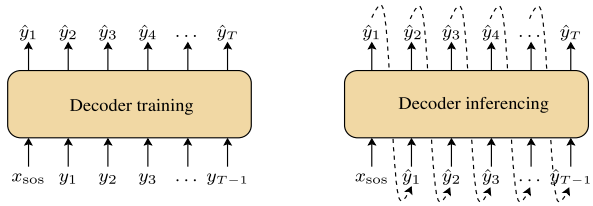


Fig. 4. The behavior of decoder in aggregated level.

which is called teacher-forcing. Masked self-attention is employed to prevent the leakage of future information. In the inference phase, the last step input of encoder will be treated as the start of sequence x_{sos} in the decoder. The x_{sos} activates decoder to generate outputs through dynamic decoding. The output of each previous step is recursively used as the input for the next step until the entire sequence is wholly predicted.

Meta-learner in this level is used to integrate a series of predictions from submodels by selecting the most accurate predictions from each submodel, resulting in the final prediction that outperforms any single model. To incorporate two

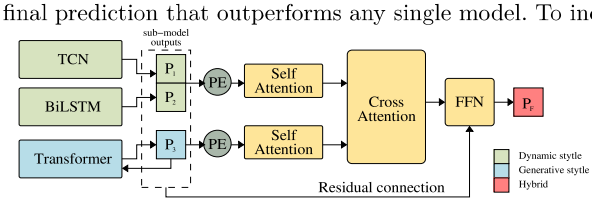


Fig. 5. Meta-learner of aggregated level.

different output styles: generative style and dynamic decoding, a self-attention based meta-learner is presented. The output of submodels is processed with positional encoding before entering self-attention blocks. After the positional encoding, the output of transformer will serve as query vectors, and the others will be treated as key and value vectors in the cross-attention. The attention score produced from cross-attention is then fed into a feed-forward network (FFN) along with the outputs of each submodel to generate final predictions.

3.2 Residential Level

Prediction at residential level aims to predict the load values recorded by smart meters. Due to the high uncertainty and volatility within these load profiles, making relatively long-term forecasts is quite challenging and often results in lagging predictions. Therefore, improving the prediction for next single timestep will be emphasized at this level.

Clustering algorithm is applied to explore the relationship between users. Based on the study [26], the pooling mechanism is presented because of the finding that a group of customers containing similarity might improve the forecasting accuracy. Here, rather than randomly grouping customers in the study [26], k-means algorithm is used to divide customers into groups. Note that the max-min scaling is needed before the clustering, avoiding an incorrect clustering result caused by different scales of load usage. Not only can the accuracy be improved through the clustering algorithm, but the amount of model usage can also be reduced. The clustering algorithm groups the customers, and hence, the model usage can be reduced from m_{n} to m_{k}. The one-hot encoding creates extra variables to identify information of customers in the same group.

Transformer Decoder is used to generate load predictions for the next time step. Informer [32], a variant of transformer [29], developed a generative-style decoder. Rather than dynamic-style decoder in vanilla designation, the decoder here outputs the prediction results at once. The following is definition:

$$X_{\mathrm{sos}} = \{x_i \,|\, x_i \in X_{\mathrm{enc}}, \frac{\mathrm{len}(X_{\mathrm{enc}})}{2} \leq i \leq \mathrm{len}(X_{\mathrm{enc}})\}, \tag{3}$$

$$X_{\mathrm{dec}} = \mathrm{concat}(X_{\mathrm{sos}}, X_0), \tag{4}$$

where X_{dec} is the decoder's input. X_{sos} denotes the start of sequence, composed of the last half of encoder inputs X_{enc}. A zero vector X_0 is concatenated at the end of X_{sos} to form the decoder input X_{dec}. Compared with the original decoder used in the study [29], the generative-style decoder allows it to have a more extended sequence input. It provides the decoder with more information than using only the last step from the encoder inputs. The position of zero vector plays a role in the query vector to investigate the importance of the sequence.

Meta-learner is used to determine the portion of each model at this level. Unlike just averaging the predictions of three submodels, the deep neural network is used to adjust the main component of submodels. As shown in Figure 6, besides

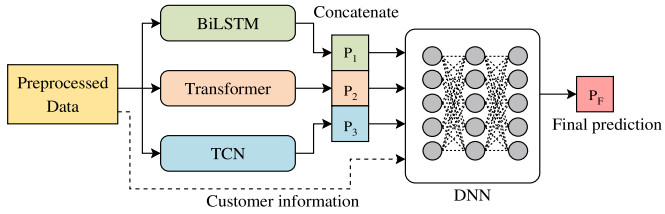


Fig. 6. Meta-learner of residential level.

the vanilla architecture of the stacking method, an additional path is added to inform the meta-learner which load profile is being predicted. The reason why the extra path is needed for the meta-learner is that each model handles load profiles from residents. With the help of customer information, meta-learner can adjust the weight of submodel effectively and hence increase the prediction accuracy.

4 Experimental Results

4.1 Experimental Environment, Settings, and Datasets

All simulations are conducted on a PC with an AMD-Ryzen 7 5700X processor, 128 GB for main memory, and NVIDIA GeForce RTX 3080 Ti. Deep learning models are written in PyTorch 2.0.1 and Python with the version of 3.8.16 on Ubuntu 20.04.6 LTS. For the comparisons of forecasting models, the number of epochs is set equal to 100 with the mechanism of early stopping, the learning rate is fixed at 0.001, and the batch size initially set to 48 and increased to 480 along with the epochs progress. To evaluate the proposed ensemble architecture, two datasets with different size of variables are used to compare deep learning models. The following is the introduction of datasets:

1. Customer Behavior Trials (ISET) [20], provided by the Commission for Energy Regulation in Ireland. It aims to investigate the habits of customers. The dataset includes 6,435 smart meters collected during July 1, 2009 to December 31, 2010. In this paper, we select 920 residents in the dataset as parameter settings of study [26]. The data is split into 1,440 data points for testing, 2,880 for the validation set, and the remaining will be training set.
2. Ausgrid Resident [22], collected from the biggest power company in the east of Australia. The dataset includes 300 smart meters gathered since July 1, 2010 to June 30, 2013. According to different deployment of appliance. the number of variables range from 2 to 3 in the last year of dataset. Zero imputation is used for missing variables to ensure the deep learning models function normally. The way to split dataset is same as previous one.

The selected users in above dataset may contain missing values. In order to let the data can be send into models, the imputation of missing values is followed.

$$f(x_t^i) = \begin{cases} \bar{X}^i, & \text{if } x_t^i \text{ is NaN}, \\ x_t^i, & \text{otherwise}, \end{cases} \tag{5}$$

where x_t^i represents the load value of i user recorded at time t. If the value is missing, which will be replaced as the average values of the user.

4.2 Results of aggregated Level

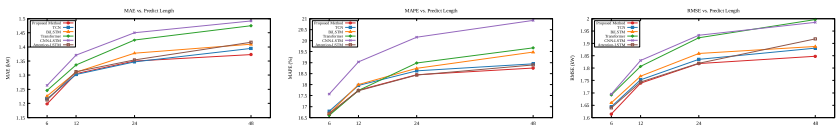
The mean absolute percentage error (MAPE) is introduced in this level forumated as: $\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$, where y_i represents the actual load value and $\hat{y}_i$ is the prediction value. The calculation makes the metric become a dimensionless quantity. The smaller the values of the metrics, the more accurate the prediction is. The benchmark models include CNN-LSTM, Attenion-LSTM, Transformer, Informer, TCN, and ASEA. Each model will take the past 96 timesteps of load data as model input to predict the next 48 timesteps. As shown in Table 1, Attention-LSTM demonstrates superior performance compared

to CNN-LSTM in both datasets, indicating the benefit of combining attention mechanisms and recurrent networks in two datasets. Also, the TCN and BiLSTM have their advantages. The TCN model yields great prediction results in the Ausgrid dataset. Instead, the BiLSTM model is good at predicting results in ISET dataset. This indicates that the TCN can handle multivariable datasets much better than the BiLSTM. The performance of the dynamic decoding style transformer may not be as strong as the Informer. Informer model performs well in MAPE metric, indicating the designation can minimize the relative errors in the single-variable dataset. Rather than using a single model, ASEA is an ensemble model proposed in the study [19]. Here, modifying the output layer makes it suitable for handling this problem. Due to the effective designation of meta-learner in the proposed method, the output of submodels can be integrated to be the ultimate prediction, achieving the most accuracy in most scenarios.

Table 1. Comparisons of aggregated level.

Model	ISET			Ausgrid		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
CNN-LSTM	2.2947	3.1263	19.4411	1.4921	1.9844	20.9205
Attention-LSTM	2.2812	3.1642	18.9331	1.4165	1.9179	18.8902
Transformer [29]	2.3063	3.1913	18.8909	1.4751	1.9959	19.6663
Informer [32]	2.2995	3.1515	**18.5639**	1.4403	1.9245	19.5485
BiLSTM [24]	2.2790	3.1084	19.2193	1.4112	1.8926	19.4583
TCN [4]	2.2807	3.1289	19.2797	1.3942	1.8806	18.9465
ASEA [19]	2.2662	3.0927	19.2601	1.3846	1.8568	19.0441
Proposed method	**2.2544**	**3.0670**	19.2313	**1.3732**	**1.8475**	**18.7485**

As shown in Figure 7, it is made by running models on the Ausgrid dataset. The



(a)MAE (b)MAPE (c)RMSE

Fig. 7. The comparison of prediction horizon.

figure shows the ability of models to predict different time horizons from 6 to 48. We can see that the attention-LSTM model has a remarkable ability in handling the scenarios of 12 and 24. In most cases, our proposed method can reach the best accuracy. Along with the prediction length going longer, the advantages of our proposed method gradually exhibit.

4.3 Results of Residential Level

In this level, the equation introduced in the previous level cannot be used because the load usage is often close to zero, leading to a divide-by-zero problem. Therefore,

the normalized root mean squared error (NRMSE), as used in the study [26], is introduced. NRMSE is a dimensionless metric like MAPE and it is defined as $\left(\sqrt{\left(\sum_{i=0}^{N}(y_i-\hat{y}_i)^2\right)/N}\right)/(y_{\max}-y_{\min})$, where $y_{\max}$ means the maximum and $y_{\min}$ is the minimum in the load profile. As Table 2 shows, transformer

Table 2. Comparisons of residential level.

Model	ISET			Ausgrid		
	MAE	RMSE	NRMSE	MAE	RMSE	NRMSE
CNN-LSTM	0.2899	0.5175	0.1075	0.2448	0.3804	0.1517
Attention-LSTM	0.3016	0.5243	0.1073	0.2292	0.3598	0.1436
Informer [32]	0.2723	0.4816	0.0998	0.1756	0.2800	0.1143
PDRNN [26]	0.2614	0.4571	0.0954	0.1496	0.2487	0.1014
Transformer [29]	0.2547	0.4519	0.0927	**0.1421**	0.2455	0.0991
BiLSTM [24]	0.2597	0.4492	0.0967	0.1451	0.2460	0.0994
TCN [4]	0.2560	0.4504	**0.0919**	0.1466	0.2474	0.0999
ASEA [19]	0.2621	0.4686	0.0971	0.1517	0.2493	0.1018
Proposed method	**0.2522**	**0.4476**	0.0933	**0.1421**	**0.2441**	**0.0986**

shows great prediction results in the MAE metric for both datasets, reflecting its strength of minimizing average absolute errors but not RMSE and NRMSE in both datasets. TCN also has a good performance in NRMSE on the ISET dataset but not MAE and RMSE in both datasets. On the other hand, the proposed method performs well across both datasets, achieving the lowest values in most cases. The stacking architecture effectively combines the outputs of different models, resulting in significantly reducing the forecasting error. This indicates that the proposed method provides the most accurate predictions among all the models.

5 Conclusion

To investigate the short-term load forecasting problem in detail, this paper conducts a comprehensive analysis at both the residential and aggregated levels. Due to the varying behavior of load profiles between customers, a clustering algorithm and aggregation method are applied to the residential and aggregated levels, respectively. The residential forecasting horizon is set to a single timestep, providing a detailed load usage of the next timestep for each customer within the grid. The forecasting horizon is set to multiple timesteps, providing a relatively long-term load usage at aggregated level. The proposed method consists of three deep learning models, including BiLSTM, TCN, and transformer, with different designations of meta-learner to each level. Compared with other deep learning models, the experiment results show the effectiveness of proposed method on two load datasets from Australia and Ireland. In future work, we will attempt to use a pruning algorithm to further reduce the model size and computations of our proposed method so that it can be installed on IoT devices with limited storage capacity.

Acknowledgment

This work was supported in part by the National Science and Technology Council of Taiwan, R.O.C., under Contract NSTC111-2222-E-110-006-MY3, NSTC112-2628-E-110-001-MY3, and NSTC113-2634-F-110-001-MBK.

Bibliography

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3), 175–185 (1992)
2. Amral, N., Ozveren, C., King, D.: Short term load forecasting using multiple linear regression. In: Proceedings of International Universities Power Engineering Conference, pp. 1192–1198 (2007)
3. Aouad, M., Hajj, H., Shaban, K., Jabr, R.A., El-Hajj, W.: A CNN-sequence-to-sequence network with attention for residential short-term load forecasting. Electric Power Systems Research **211**, 108152 (2022)
4. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
5. Ceperic, E., Ceperic, V., Baric, A.: A strategy for short-term load forecasting by support vector regression machines. IEEE Transactions on Power Systems **28**(4), 4356–4364 (2013)
6. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**, 273–297 (1995)
7. Dudek, G.: Pattern-based local linear regression models for short-term load forecasting. Electric Power Systems Research **130**, 139–147 (2016)
8. Fan, G.F., Guo, Y.H., Zheng, J.M., Hong, W.C.: Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. Energies **12**(5), 916 (2019)
9. Hambali, A., Akinyemi, M., JYusuf, N.: Electric power load forecast using decision tree algorithms. Computing, Information Systems, Development Informatics & Allied Research Journal **7**(4), 29–42 (2016)
10. Hippert, H.S., Pedreira, C.E., Souza, R.C.: Neural networks for short-term load forecasting: A review and evaluation. IEEE Transactions on Power Systems **16**(1), 44–55 (2001)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
12. Hossen, T., Plathottam, S.J., Angamuthu, R.K., Ranganathan, P., Salehfar, H.: Short-term load forecasting using deep neural networks. In: Proceedings of North American Power Symposium, pp. 1–6 (2017)
13. Humeau, S., Wijaya, T.K., Vasirani, M., Aberer, K.: Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In: Proceedings of Sustainable Internet and ICT for Sustainability, pp. 1–6 (2013)
14. Huy, P.C., Minh, N.Q., Tien, N.D., Anh, T.T.Q.: Short-term electricity load forecasting based on temporal fusion transformer model. IEEE Access **10**, 106296–106304 (2022)
15. Kazeminejad, M., Dehghan, M., Motamadinejad, M., Rastegar, H.: A new short term load forecasting using multilayer perceptron. In: Proceedings of International Conference on Information and Automation, pp. 284–288 (2006)
16. Kim, T.Y., Cho, S.B.: Predicting residential energy consumption using CNN-LSTM neural networks. Energy **182**, 72–81 (2019)

17. Kyriakides, E., Polycarpou, M.: Short term electric load forecasting: A tutorial. Trends in Neural Computation pp. 391–418 (2007)
18. Lang, C., Steinborn, F., Steffens, O., Lang, E.W.: Applying a 1D-CNN network to electricity load forecasting. In: Proceedings of Theory and Applications of Time Series Analysis, pp. 205–218 (2020)
19. Lu, C.T., Tsai, C.W.: An effective adaptive stacking ensemble algorithm for electricity theft detection. In: Proceedings of International Conference on Intelligent Computing and its Emerging Applications, pp. 22–27 (2021)
20. Martin, G.: Electricity smart metering customer behaviour trials findings report, technical report. CER Commission for Energy Regulation pp. 1–146 (2011)
21. Ran, P., Dong, K., Liu, X., Wang, J.: Short-term load forecasting based on ceemdan and transformer. Electric Power Systems Research **214**, 108885 (2023)
22. Ratnam, E.L., Weller, S.R., Kellett, C.M., Murray, A.T.: Residential load and rooftop PV generation: An Australian distribution network dataset. International Journal of Sustainable Energy **36**(8), 787–806 (2017)
23. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
24. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
25. Sevlian, R., Rajagopal, R.: Short term electricity load forecasting on varying levels of aggregation. arXiv preprint arXiv:1404.0058 (2014)
26. Shi, H., Xu, M., Li, R.: Deep learning for household load forecasting—a novel pooling deep rnn. IEEE Transactions on Smart Grid **9**(5), 5271–5280 (2017)
27. Stephen, B., Tang, X., Harvey, P.R., Galloway, S., Jennett, K.I.: Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting. IEEE Transactions on Smart Grid **8**(4), 1591–1598 (2015)
28. Tarmanini, C., Sarma, N., Gezegin, C., Ozgonenel, O.: Short term load forecasting based on ARIMA and ANN approaches. Energy Reports **9**, 550–557 (2023)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
30. Wang, H., Zhao, Y., Tan, S.: Short-term load forecasting of power system based on time convolutional network. In: Proceedings of International Symposium on next Generation Electronics, pp. 1–3 (2019)
31. Zhao, Z., Xia, C., Chi, L., Chang, X., Li, W., Yang, T., Zomaya, A.Y.: Short-term load forecasting based on the transformer model. Information **12**(12), 516 (2021)
32. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)

PUF-Based Authentication and Authorization Protocol for IoT

Yu-Tse Shih[1], Shih-Ming Huang[2][0000−0001−9305−921X], and Chun-I Fan[1,3,4,⋆][0000−0002−7512−1291]

[1] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
steven8450@gmail.com, cifan@mail.cse.nsysu.edu.tw
[2] Industrial Technology Research Institute, Hsinchu 310401, Taiwan
VincentSMH@itri.org.tw
[3] Information Security Research Center, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
[4] Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung 804201, Taiwan

Abstract. With the rapid development of technology, smart devices have become indispensable tools for people, and IoT devices are ubiquitous, playing an important role in our daily lives. When using IoT device applications and services, users must go through complex authentication processes, which are time-consuming and pose various security issues. Not only must users be authenticated, but the devices must also be identified to ensure security. Physically Unclonable Functions (PUF) can be considered as hardware fingerprints due to their randomness and unpredictability, used to identify the uniqueness of devices. Given the limited computational power of IoT devices, the need for lightweight mechanisms is even more pronounced. Therefore, this protocol proposes a three-party mutual authentication using a smartphone involving the user, the server, and the IoT device. The PUF within the device generates a session key to authorize the IoT device to use and access the service. Finally, the contributions of this mechanism are highlighted through security analysis and comparison with related works.

Keywords: Physically Unclonable Functions · Authentication · Authorization · Key Exchange · Internet of Things

1 Introduction

In recent years, Internet of things (IoT) environments [1] have grown dramatically due to the rapid development of the 5th generation mobile network (5G) [2]. There are lots of IoT devices designed to assist our daily lives. More and more IoT devices can be used through people's smart devices. Not only can those IoT devices help with daily tasks but provide different services on their applications

⋆ Corresponding auther

as well. There are two scenarios of network environments. One is a sensor network environment, and the other is a machine-to-machine environment.
In the IoT environments, the user tends to access the IoT devices. Still, they must be authenticated or authorized to protect the user's privacy or guarantee the security of transferred information [3]. With the rapid development of technologies, there have been various types of IoT devices recently. We may access IoT devices in different situations and locations as we need. IoT devices provide many applications or services in public places that we only have to access temporarily. In this situation, we need an authorization framework [8] and acquire the right to access those IoT devices [5].
After the specific times, the IoT devices can provide their services to others. We take the rental car as an example. If a customer needs to rent a car for a week, then she/he needs to get the authority to access the specific smart car and access the services during the authorization time. Besides, the smartphone has gradually become an indispensable piece of equipment to humans recently. Hence, the smartphone is capable of playing an important role as an authentication factor to assist the authentication of IoT devices [6]. Customers can get authorization through authenticated smartphones. Under the circumstances, authenticating the specific IoT devices by Physically Unclonable Function (PUF) may become a proper method to deal with it. C. Herder*et al.* [7] mentioned that due to the slight differences between manufacturing chips, they generate different challenge-response pairs (CRPs). PUF generates different challenge-response pairs (CRPs). In 2017, M. Beltrán *et al.* [8] proposed an authentication and authorization protocol combining PUF and tokens. They adopted PUF and tokens to authenticate and authorize federated system-to-services through HTTP and CoAP.
The protocol uses pseudo-random function (PRF) and XOR operation instead of the traditional hash function and symmetric encryption algorithm, which ensures security while reducing the overhead. Moreover, the proposed protocol does not require the server to store a large number of challenge-response pairs (CRPs), which reduces the storage overhead on the server while avoiding the risk of leakage of CRPs. In summary, we will propose a novel protocol by applying PUF to enhance security during the authorization period with various services for IoT devices. Meanwhile, we simplify and accelerate the authorization process through the authenticated smartphone to achieve the authentication and authorization of the proposed protocol.

Contributions We introduce a PUF-Based authentication and authorization protocol for IoT environment on the smartphone in this research. The contributions of the research are as follows:

1. The proposed protocol provides an approach to accelerate and simplify the authentication and authorization phase.
2. The proposed protocol applies PUF to guarantee the security of parameters and user's information and privacy.

3. The proposed protocol can provide more comprehensive security features than most related works.

2 Preliminaries

We will introduce all the definitions and knowledge of mechanisms used in the proposed scheme in this section.

Internet of Things Recently, the Internet of Things has become more and more popular and common in our daily lives. In the Internet of Things environment, a combination of many different networks, such as machine-to-machine network (M2M) [9], device-to-device network (D2D) [10], wireless volumetric network (WBAN) [11] and wireless sensor network (WSN) [12]. The goal of IoT is to allow devices or sensors to connect and exchange data.

In the IoT environment, security issues [13] are the most important concern because the computing and storage capabilities of devices are uneven. In addition, group authentication and key agreement are important issues because devices and sensors will share data with many participants.

Physically Unclonable Function This part introduces physically unclonable functions (PUF) in key generation applications and some low-cost authentication. First, it inspired PUFs and traditional secure non-volatile memory and defined two main types of PUFs: strong PUFs and weak PUFs. The strong PUF is usually used in low-cost authentication. Next, the implementation of weak PUF and its use in key generation applications are covered here. It covers error correction schemes such as pattern matching and index-based coding. Each PUF will output a unique value when it gets a challenge because of its integrated circuit (IC). So, each different PUF can generate its unique challenge-response pair. That's why someone called this characteristic like a fingerprint of this PUF. Another integrated circuit cannot produce it. Therefore, PUF can become a secret value to represent the specific device when we adopt it. Then, we briefly introduce the definition of PUF below:

Each PUF can be considered as a generator of the unique challenge-response pairs (CRPs). It can be represented as $R = PUF(C)$, where C is a challenge and R is the return. A PUF returns a 1-bit response in every execution. We denote an integrated PUF with d PUFs by d-bit PUF, where $d > 1$. An ideal d-bit PUF should have the following properties:

- For any two different d-bit PUFs PUF_1, PUF_2, and any challenge C_1, $HD(PUF_1(C_1), PUF_2(C_1)) \approx d/2$, where HD denotes the hamming distance.
- For any two different d-bit PUFs PUF_1, PUF_2, and any two different challenges C_1, C_2, $HD(PUF_1(C_1), PUF_2(C_2)) \approx d/2$.

- For any d-bit PUF PUF_1, and any challenge C_1, $HD(PUF_1(C_1), PUF_1(C_1)^{'}) = 0$. (But in reality, it usually has an error within $d/10$ bits since PUF may be affected by the environment.)
- For any d-bit PUF PUF_1, and any two different challenges C_1, C_2, $HD(PUF_1(C_1), PUF_1(C_2)) \approx d/2$.

3 Related Works

In this section, we introduce some related works and the survey paper from [24]. We compare some functionalities and security issues of these related works with the proposed scheme. There are many kinds of authentication methods in the IoT environments along with many existent challenges. We will briefly introduce and explain these related works and discuss their advantages and disadvantages.

3.1 Beltrán *et al.*'s Authorization Scheme with PUFs

In 2017, Beltrán *et al.* proposed an authorization scheme with PUFs [8]. The protocol usually guarantees the confidentiality and integrity of basic data and uses encryption to protect communication data, but there are still critical issues related to authorization and authentication. They provided a new system to service authorization and authentication mechanism based on a combination of PUF and two tokens, which can rely on HTTP or COAP on the joint plan to adapt to the particular requirements of this environment. This mechanism is validated, and they also use real healthcare case studies to evaluate its efficiency and safety.

3.2 Beltrán's Authentication Scheme in IoT

In 2018, Marta Beltrán proposed an IoT authentication and authorization scheme called identifying, authenticating, and authorizing smart objects and end-users to cloud services in the IoT [18]. Their scheme allows IoT services deployed in the cloud or locally to authenticate and authorize smart objects using COAP and HTTP. The end-users can be identified, authorized, and authenticated through their scheme through these smart objects and/or re-inquired if possible. In addition, they validated their scheme, using real healthcare case studies to evaluate its usability, efficiency, and safety.

3.3 Jin *et al.*'s Multi-Factor Authenticated Key Exchange with PUF

In 2019, Jin *et al.* proposed an authenticated key Exchange with PUF named an efficient multi-factor authenticated key exchange with a physically unclonable function [**?**]. In this part, they proposed an efficient and secure physical non-transparent function based on multi-factor authentication key exchange (made

by PUF). The main difficulty comes from the fact that it should establish public keys from multi-factor authenticators and PUF-embedded devices. Their architecture is a secure PUF production protocol, requiring only two communication phases.

3.4 Siddiqui *et al.*'s Lightweight PUF–PKI Digital Certificate Authentication Scheme for the IoTs

In 2023, Siddiqui *et al.* propose a highly secure and robust authentication protocol[19] based on a public key infrastructure (PKI) digital certificate based on two certificate authorities (CAs) for cloud IoT systems. The proposed authentication method is verified and validated using the Tamarin prover and supported with a detailed security and performance analysis discussion. The scheme's security and privacy attributes are compared with other IoT authentication schemes. The analysis has proved that the proposed authentication scheme is more secure and highly reliable as compared to the Prosanta and Biplab authentication schemes.

4 The Proposed Protocol

In this section, we propose a secure and lightweight PUF-based authentication and authorization protocol for IoT devices. There are three roles in the proposed protocol. The first one is user equipment (UE), the second role is an IoT device, and the last role is the server which is the provider of the service that the user tends to use. Each part in this protocol needs to carry a PUF chip and use PUF to generate parameters. In the registration phase, both the user and IoT device need to register with the server in a secure channel. Afterward, in the login phase, the user and IoT device are close to each other. They share the information generated by the server with each other through NFC communication. Next, the server verifies user equipment and IoT devices, respectively. If the user and IoT device pass the verification, the server generates and sends a session key to the IoT device. The proposed protocol contains three phases: the registration phase, the login phase, and the long-term key update phase.

4.1 Setup

In the beginning, S sets the necessary parameters and functions used in the proposed protocol. S first randomly picks a number s as a secret key for symmetric encryption. We know that there is a very small difference between two responses in the same PUF function even if we use the same challenge. However, we can use a fuzzy extractor to recover the original output. Therefore, we assume that the perfect PUF function is adopted in the proposed protocol. It means that the PUF function will generate the same response if we issue the same challenge. Finally, S needs to keep a registration table to verify the user and IoT device.

Table 1. Notations

Notation	Description
S	the server
ID_i	the identity of user i
D_j	the identity of IoT device j
C_{ID_i}	a random number chosen by user i as PUF function seed
C_{S,ID_i}	a random number chosen by S as PUF function seed
R^m_{S,ID_i}	the m-th response generated from PUF function stored in S
PUF_{ID_i}	the 128-bit PUF in user i's equipment
$Cert_T$	the certificate with a time period T
C'_{S,ID_i}	new random number chosen by S for user i as PUF function seed
s	a secret key kept by S
$E_s(.)$	a symmetric encryption with key s
$D_s(.)$	a symmetric decryption with key s
$\oplus$	the bitwise exclusive-or (XOR) operator

4.2 Registration

In the registration phase, we divide the proposed protocol into two parts, where one is user equipment and the other is an IoT device. Note that the registration phase is executed in a secure channel and user equipment needs to download and execute the procedure under the application. First, user equipment i randomly chooses a number C_{ID_i} and then user equipment i inputs C_{ID_i} as a challenge to the PUF function n times and generates the response $R^n_{S,\ ID_i}$. Next, user equipment i sends ID_i and $R^n_{S,\ ID_i}$ to S. After S receives the parameter from user i, S also picks a random number $C_{S,\ ID_i}$ and takes this random number as a challenge into PUF function and executes m_1 times to generate $R^{m_1}_{S,ID_i}$. And then, S sends $R^{m_1}_{S,\ ID_i}$ back to user i and keeps ID_i, $C_{S,\ ID_i}$ and $R^n_{ID_i}$ in the registration table. At last, user equipment i stores $R^{m_1}_{S,\ ID_i}$ and C_{ID_i} in the equipment. The procedure of the IoT device part in the registration phase is mostly the same as the user equipment part. The only difference is that S executes PUF function m_2 times instead of m_1.

4.3 Login and Session Key Establishment

In the scenario of the proposed protocol, the user needs to be close to the IoT device; therefore, the user can operate or use the IoT device. We assume that the user transfers and receives the parameters from S in the registration phase under NFC communication. It means that there exists a secure channel between the user equipment and the IoT device. In the login phase, the application that works in the user equipment asks user i to pass the biometric recognition and then the user equipment randomly chooses a number r_{ID_i} and computes $P_{ID_i} = R^{m_1}_{S,\ ID_i} \oplus r_{ID_i}$. Next, the user equipment sends P_{ID_i} to the IoT device. Note that the transfer channel between IoT devices and user equipment is

based on NFC communication. Therefore, it can guarantee the security of parameter transmission. Meanwhile, the IoT device randomly picks a number r_{D_j} and computes $P_{D_j} = R^{m_2}_{S,\ D_j} \oplus r_{D_j}$ and then sends it to user equipment. At last, user equipment computes $X = R^{m_2}_{S,\ D_j} \oplus r_{D_j} \oplus r_{ID_i}$ and IoT device computes $Y = R^{m_1}_{S,\ ID_i} \oplus r_{ID_i} \oplus r_{D_j}$ which will be stored in the device.

In the login and session key establishment phase, user i inputs C_{ID_i} to PUF function as a challenge and executes $n-1$ times to generate $R^{n-1}_{ID_i}$. Then, user i computes $P_{ID_i,\ D_j} = E_{R^n_{ID_j}}\left(ID_i,\ D_j, X, R^{n-1}_{ID_i}\right)$ and sends P_{ID_i,D_j} to S. At the same time, the IoT device computes $R^{n-1}_{D_j} = PUF^{n-1}_{D_j}\left(C_{D_j}\right)$ by the PUF function stored in the device. Next, the IoT device encrypts $ID_i, D_j, Y, R^{n-1}_{D_j}$ with symmetric key R^n_{S,D_j} and then generates and sends P_{D_j,ID_i} to S.

After receiving P_{ID_i,D_j} and P_{D_j,ID_i} from user i and D_j, S decrypts P_{ID_i,D_j} with symmetric key $R^n_{ID_i}$ and decrypts P_{D_j,ID_i} with symmetric key $R^n_{D_j}$ to get parameters X and Y. Then, S verifies if $X \oplus \mathrm{Y}$ is equal to $R^{m_1}_{S,ID_i} \oplus R^{m_2}_{S,D_j}$. Next, S randomly chooses $r_{i,j}$ and calculates $k_{S,D_j} = \left(r_{i,j} \oplus R^{n-1}_{D_j}\right)$. Meanwhile, S executes PUF function PUF_{S,ID_i} $m_1 - 1$ times and PUF_{S,D_j} $m_2 - 1$ times to obtain $R^{m_1-1}_{S,ID_i}$ and $R^{m_2-1}_{S,D_j}$. At last, S computes $P_{S,ID_i,D_j} = E_X\left(ID_i,\ D_j,\ Cert_T,\ R^{m_1-1}_{S,\ ID_i}\right)$ and $P_{S,\ D_j,\ ID_i} = E_Y(D_j,\ ID_i,\ Cert_T, R^{m_2-1}_{S,\ D_j})$ and then sends them to user i and D_j, respectively.
Finally, user i receives $R^{m_1-1}_{S,\ ID_i}$ and stores it in the device. In the IoT device side, D_j also receives $P_{S,\ D_j,\ ID_i}$ and gets session key $k_{S,\ D_j}$ from decrypting $P_{S,\ D_j,\ ID_i}$. And then, D_j stores $R^{m_2-1}_{S,\ ID_i}$ for next login. D_j then uses session key $k_{S,\ D_j}$ to communicate with S. Note that S will check $Cert_T$ to verify if this transmission is a replay attack or not. The session key establishment by S of login phase is shown in Fig.1.

4.4 Long Term Key Update Phase

In the long-term key update phase, we also divide this phase into two parts. The first one is the user equipment part, and the other is the IoT device part. The long-term key is limited when m_1 in user equipment or m_2 in the IoT device equals to 1. User equipment and IoT devices need to request S to update the long-term key.

In the user equipment part, when $m_1 = 1$ in $R^{m_1}_{S,\ ID_i}$, UE needs to update the long-term key. User equipment first chooses a random number u_{ID_i} and encrypts ID_i, u_{ID_i} with $R^1_{S,\ ID_i}$ to generate Req_{ID_i}. And then, user i sends Req_{ID_i} to S. S decrypts Req_{ID_i} and randomly chooses a number $C'_{S,\ ID_i}$. Next, S uses $C'_{S,\ ID_i}$ as new challenge and executes PUF function m_1 times to get $R'^{m_1}_{S,\ ID_i}$. At last, S updates the information in registration table of user i and

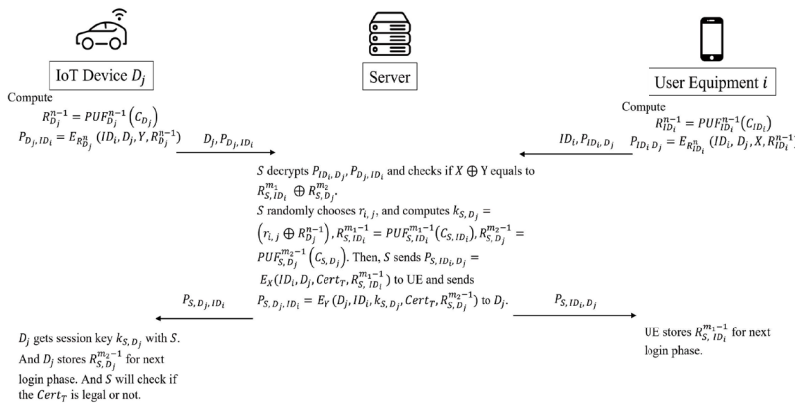


Fig. 1. Login and Session Key Establishment

sends $Res_{ID_i} = E_{u_{ID_i}}\left(ID_i,\ R'^{m_1}_{S,\ ID_i}\right)$ to user equipment. Finally, user i decrypts Res_{ID_i} and updates long term key as $R'^{m_1}_{S,\ ID_i}$.

In the IoT device part, when $m_2 = 1$ in $R'^{m_2}_{S,D_j}$, D_j needs to update the long term key. D_j first chooses a random number u_{D_j} and encrypts D_j, u_{D_j} with R^1_{S,D_j} to generate Req_{D_j}. And then, D_j sends Req_{D_j} to S. S decrypts Req_{D_j} and randomly chooses a number C_{S,D_j}. Next, S uses C'_{S,D_j} as new challenge and executes PUF function m_2 times to get $R'^{m_2}_{S,D_j}$. At last, S updates the information in registration table of IoT device D_j and sends $Res_{D_j} = E_{u_{D_j}}\left(D_j, R'^{m_2}_{S,D_j}\right)$ to IoT device. Finally, D_j decrypts Res_{D_j} and updates long term key as $R^{m_2}_{S,\ D_j}$.

5 Security Analysis

In this part, we provide the security analysis corresponding to every phase of the proposed protocol. We protect the security of the parameters in the proposed protocol mainly based on the unique output from PUF chips. Each of the user equipment, the IoT device, and the server in the proposed protocol contains a PUF chip. Furthermore, only the same PUF chip can get the same response from the original challenge. Therefore, we generate the long-term key calculate some parameters in the registration phase, and store them in user equipment, IoT devices, and the server. Moreover, in the login phase, we update the session key every time we log in. Furthermore, we list some security properties we achieve in the proposed protocol as follows.

– Mutual Authentication
  We discuss mutual authentication between S and the user side. In the pro-

posed protocol, we discuss the mutual authentication between S and IoT devices as well. In the Login and Session Key Establishment of the proposed protocol, the user equipment and the IoT device take X and Y as challenges in the challenge response pair. Since the parameter X is computed from $R^{m_2}_{S,D_j}$, r_{D_j} and r_{ID_i}, and Y is computed from $R^{m_1}_{S,\ ID_i}$, r_{D_j} and r_{D_j}, r_{ID_i} and r_{D_j}, only S can calculate $R^{m_2}_{S,\ D_j}$ and $R^{m_1}_{S,\ ID_i}$ through the corresponding PUF function. Therefore, S can verify the legality of the user and IoT device by X and Y. Finally, if S can decrypt and compute the $P_{S,\ ID_i,\ D_j}$ and $P_{S,\ D_j,\ ID_i}$ correctly, user and IoT device can verify S.

- Resistance to the Replay Attack
  The adversary can replay $P_{ID_i,\ D_j}$ or $P_{D_j,\ ID_i}$ to S. However, the adversary can not affect the authentication procedure or cause any damage to the proposed protocol.

- Forward Secrecy
  Since S keeps the PUFs in S's database, the adversary can not produce the session key $R_{S,\ ID_i}$ or $R_{S,\ D_j}$ even if the adversary gets the long-term key $C_{S,\ ID_i}$ or $C_{S,\ D_j}$. Furthermore, S will update the session key with the user equipment and IoT device. Even if the adversary obtains the session key, the adversary cannot still decrypt previous messages.

- Resisting Stolen User Equipment Attacks
  The operation of the proposed protocol is based on the applications stored in the smartphone. Furthermore, the PUF function also relies on the working device. If the device is off, then the PUF function can not produce any response. We adopt biometric authentication in the application to resist stolen user equipment attacks.

6 Performance Comparison and Results

6.1 Properties Comparison

In this section, we evaluate the functionalities and security properties with some related works. Because the proposed protocol is an authentication and authorization protocol by smartphone combined with PUF chips, we can not find precisely related work with the proposed protocol. However, we still compare some functions and security properties close to the proposed protocol. We compare six security properties in this section. We also list the security properties below in Table **??** that we achieve.

6.2 Computation Cost

In this paragraph, we evaluate the efficiency of the proposed protocol. We divide the proposed protocol into three phases. However, the setup will be only

Table 2. The Comparisons of Functionalities and Security Properties

	Beltrán *et al.* [8]	Siddiqui *et al.*[19]	Beltrán [18]	Zhao *et al.* [20]	Yilmaz *et al.* [21]	Ours
S1	Yes	No	Yes	Yes	Yes	Yes
S2	No	Yes	No	Yes	No	Yes
S3	No	Yes	No	Yes	No	Yes
S4	No	No	No	No	No	Yes
S5	No	No	No	Yes	No	Yes

S1: Mutual Authentication; S2: Session Key Security; S3: Resistance to the Replay Attack; S4: Forward Secrecy; S5: Resisting Stolen User Equipment (SIM Card) Attack;

executed once. Hence, we only calculate three phases. The first one is the registration phase; every user and IoT device needs to register initially. And then, the second one is the login phase. Each registered user who tends to access the IoT Device needs to implement this phase. Finally, the IoT device needs to set a session key with S in the third part of the login phase. We evaluate the computation cost of the registration, login, and long-term key update phases. The time of each operation is measured with Ubuntu 16.04, i7-6700 3.40 GHz, 16.0 GB DDR4 2133MHz RAM, which have been shown in [20][22]. We define T_e and T_{PUF} to be the time cost for executing symmetric encryption and decryption for a 128-bit block with AES-EAX (0.05ms) and PUF run time (0.12ms), respectively.

In this part, we use some code names to represent the cost on which side and phase. We define U as the user side, I as the IoT device side, and SV as the server side. Moreover, we use the number 1 for the registration phase, number 2 for the login phase, and 3 for the long-term key update phase. In the proposed protocol, the registration phase on the user side and IoT device side costs $\mathrm{U1} = nT_{PUF}$, and $\mathrm{SV1} = (m_1 + m_2)T_{PUF}$ in the server side. In the login phase, the cost in the user side is $\mathrm{U2} = (n-1)T_{PUF} + 2T_e$, $\mathrm{I2} = (n-1)T_{PUF} + 2T_e$ in the IoT device side, and $\mathrm{SV2} = (m_1 + m_2 - 2)T_{PUF} + 2T_e$ in the server side. Finally, in the long-term key update phase, the cost on the user side is $\mathrm{U3} = 2T_e$, $\mathrm{I3} = 2T_e$ in the IoT device side, and $\mathrm{SV3} = m_1 T_{PUF} + 2T_e$ in the server side. The detailed information is shown in Table 3.

7 Conclusion

In the proposed protocol, we not only combine the advantages of PUF and smartphones but also achieve fast authentication and authorization for IoT devices. We also provide security against replay attacks, perfect forward secrecy, and formal security proof to ensure that the proposed protocol is secure. Although only one service can be authorized each time in the proposed protocol, we hope to improve and achieve batch authorization to make the proposed protocol more

Table 3. The Comparisons of Functionalities and Security Properties

	User Side (**U**)	IoT Device Side (**I**)	Server Side (**SV**)
Registration Phase (**1**)	**U1** nT_{PUF}=0.12n (ms)	**I1** nT_{PUF}=0.12n (ms)	**SV1** $(m_1+m_2)T_{PUF}$ =0.12(m_1+m_2) (ms)
Login Phase (**2**)	**U2** $(n-1)T_{PUF}+2T_e$ =0.12 n −0.02(ms)	**I2** $(n-1)T_{PUF}+2T_e$ = 0.12 − 0.02(ms)	**SV2** $(m_1+m_2-2)T_{PUF}+2T_e$ = 0.12(m_1+m_2) − 0.14(ms)
Long Term Key Update Phase (**3**)	**U3** $2T_e$ = 0.1(ms)	**I3** $2T_e$ = 0.1(ms)	**SV3** $m_1(m_2)T_{PUF}+2T_e$ = 0.12$m_1(m_2)$ + 0.1(ms)

convenient. Besides, we use certification with limited time to achieve the passive revocation in the proposed protocol. We consider increasing the properties of active revocation to conform to the reality usage. We believe that the proposed protocol can provide substantial assistance to improve the daily lives of human beings in the future.

Acknowledgments. This work was supported in part by the National Science and Technology Council of Taiwan under Grants NSTC 113-2634-F-110-001-MBK, 113-2221-E-110-082, in part by the Information Security Research Center at National Sun Yat-sen University, and in part by the Intelligent Electronic Commerce Research Center from the Featured Areas Research Center Program through the Framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

References

1. M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of things (iot): Research, simulators, and testbeds," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1637–1647, 2017.
2. 3GPP, "Ts 33.501 security architecture and procedures for 5g system," tech. rep., 3GPP, 2018.
3. M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.
4. L. Seitz, G. Selander, and C. Gehrmann, "Authorization framework for the internet-of-things," *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–6, 2013.
5. H. Kim and E. A. Lee, "Authentication and authorization for the internet of things," *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.
6. M. A. Ferrag, L. Maglaras, and A. Derhab, "Authentication and authorization for mobile iot devices using biofeatures: Recent advances and future trends," *Security and Communication Networks*, vol. 2019, 2019.
7. C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

8. M. Beltrán, M. Calvo, and S. González, "Federated system-to-service authentication and authorization combining pufs and tokens," *2017 12th International Symposium on Re-configurable Communication-centric Systems-on-Chip (ReCoSoC)*, pp. 1–8, 2017.
9. C. Buratti, A. Stajkic, G. Gardasevic, S. Milardo, M. D. Abrignani, S. Mijovic, G. Morabito, and R. Verdone, "Testing protocols for the internet of things on the euwin platform," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 124–133, 2015.
10. K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to lte-advanced networks," *IEEE communications magazine*, vol. 47, no. 12, pp. 42–49, 2009.
11. B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless networks*, vol. 17, no. 1, pp. 1–18, 2011.
12. J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
13. J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys adn Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.
14. A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for puf-enabled rfids," *International Conference on Financial Cryptography and Data Security*, pp. 374–389, Springer, 2012.
15. Keromytis, Angelos D. *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2012.
16. T. Halonen, "Authentication and authorization in mobile environment," *Tik-110.501 Seminar on Network Security*, Citeseer, 2000.
17. C. I. Fan, Y. C. Chan, and Z. K. Zhang, "Robust remote authentication scheme with smart cards," *Computers and Security*, vol. 24, no. 8, pp. 619 – 628, 2005.
18. M. Beltrán, "Identifying, authenticating and authorizing smart objects and end users to cloud services in internet of things," *Computers and Security*, vol. 77, pp. 595–611, 2018.
19. Z. Siddiqui, J. Gao and M. Khurram Khan, "An Improved Lightweight PUF–PKI Digital Certificate Authentication Scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19744-19756, 15 Oct.15, 2022, doi: 10.1109/JIOT.2022.3168726.
20. J. Zhao, W. Bian, D. Xu, B. Jie, X. Ding, W. Zhou, and H. Zhang, "A secure biometrics and pufs-based authentication scheme with key agreement for multi-server environments," *IEEE Access*, vol. 8, pp. 45292–45303, 2020.
21. Y. Yilmaz, S. R. Gunn, and B. Halak, "Lightweight puf-based authentication protocol for iot devices," *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*, pp. 38–43, IEEE, 2018.
22. S. Kumari, A. K. Das, X. Li, F. Wu, M. K. Khan, Q. Jiang, and S. K. Hafizul Islam, "A provably secure biometrics-based authenticated key agreement scheme for multi-server environments," *Multimedia Tools Appl.*, vol. 77, p. 2359–2389, Jan. 2018.
23. An, Yang, et al. "A lightweight and practical anonymous authentication protocol based on bit-self-test PUF." *Electronics* 11.5 (2022): 772.
24. P. Mall, R. Amin, A. K. Das, M. T. Leung and K. -K. R. Choo, "PUF-Based Authentication and Key Agreement Protocols for IoT, WSNs, and Smart Grids: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205-8228, 1 June1, 2022, doi: 10.1109/JIOT.2022.3142084.

Field Testing and Detection of Camera Interference for Autonomous Driving

Ki Beom Park and Huy Kang Kim

School of Cybersecurity, Korea University, Republic of Korea
{vkdnj0413,cenda}@korea.ac.kr

Abstract. In recent advancements in connected and autonomous vehicles (CAVs), automotive ethernet has emerged as a critical technology for in-vehicle networks (IVNs), superseding traditional protocols like the CAN due to its superior bandwidth and data transmission capabilities. This study explores the detection of camera interference attacks (CIA) within an automotive ethernet-driven environment using a novel GRU-based IDS. Leveraging a sliding-window data preprocessing technique, our IDS effectively analyzes packet length sequences to differentiate between normal and anomalous data transmissions. Experimental evaluations conducted on a commercial car equipped with H.264 encoding and fragmentation unit-A (FU-A) demonstrated high detection accuracy, achieving an AUC of 0.9982 and a true positive rate of 0.99 with a window size of 255.

Keywords: Automotive ethernet and Camera Interference · Intrusion Detection.

1 Introduction

Automotive ethernet [7] in recent connected and autonomous vehicles (CAVs) stands for in-vehicle networks (IVNs) between electric control units (ECUs). controller area network (CAN) traditionally has dominated the IVNs market share. However, due to the limited bandwidth of CAN, automotive ethernet is becoming a successor to enable high-definition applications such as video streaming for autonomous driving and infotainment systems.

CAN supports a data transfer rate of up to 1 Mbps, but its limited bandwidth renders it incapable of handling simultaneous communications with multiple devices necessary for autonomous driving.

Conversely, automotive ethernet facilitates data transfer speeds of up to 1 Gbps, enabling real-time transmission of data from cameras, LiDAR and Radar. Its extensive bandwidth supports seamless communication with multiple devices. Furthermore, it utilizes protocols such as the Audio/Video Transport Protocol (AVTP), Scalable Service-Oriented Middleware over IP (SOME/IP) and Diagnostics over Internet Protocol (DoIP) to provide various functions essential for autonomous driving.

Autonomous driving technologies leverage an array of sensor equipment, including cameras, LiDAR and Radar, to detect pedestrians, interpret traffic signals, gauge distances from other vehicles, and recognize lane markings to maintain the lane. Recently, Tesla introduced a vehicle capable of autonomous driving using only cameras [17]. This camera-based autonomous driving system, known as Tesla Vision, operates without ultrasonic sensors or LiDAR. While automotive ethernet-based IVNs offer significant advantages from a cybersecurity perspective, some security risks and threats persist. For example, although a switched medium prevents the occupation of the entire bandwidth by a specific node, as seen in CAN buses, it still transmits packets from legitimate ECUs as well as potential attack nodes.

A compromised ECU can enable an attacker to manipulate the vehicle in desired ways, potentially causing significant harm to the driver and the surrounding environment. For instance, an attacker who has taken control of an ECU connected to the vehicle's data transmission switch can disrupt the data being sent to the autonomous driving module. Among various attacks in automotive ethernet environment, we focused on camera interference attack (CIA) that can disrupt autonomous driving. The overall workflow for the CIA detection is as shown in Fig. 1. We assumed that an attacker compromise the IVNs by exploiting vulnerabilities of OTA, supply chain management or using physical access. Then, the attacker can send the CIA to the target car. Based on the attack scenarios, we built the datasets which includes the normal and attack cases. Then we developed a model for the CIA detection and estimated the overall performance of the model.

IDS is considered as a primary countermeasure to monitor in-vehicle traffic and detect anomalies. Several studies (see §3) have recently proposed IDS to secure automotive ethernet-based IVNs. Although these proposed methods have exhibited satisfactory performance, a common limitation in the literature is that experimental results are based on datasets captured on a testbed rather than on commercial vehicles.

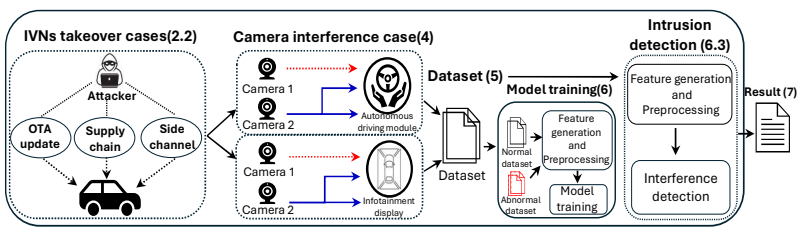


Fig. 1: Overview of the proposed methodology for intrusion detection workflow.

1.1 Contribution

In this paper, we first provide a proof-of-concept of CIA on an automotive ethernet-driven vehicle, specifically, Hyundai Genesis G80 model, based on a ma-

licious packet injection attack. To the best of our knowledge, this is the first study on a commercial automotive ethernet-driven vehicle. Second, we have compiled a network intrusion dataset containing benign traffic (from legitimate around-view cameras) and abnormal traffic (from the attacker). Finally, we propose CIA-IDS, an IDS that effectively detects the CIA related hacking attempts. We leverage a novel feature to detect network intrusion on automotive ethernet.

2 Background

2.1 Ethernet-Based IVNs

Automotive ethernet has been introduced in modern vehicles to overcome the limitations in transmission speed and bandwidth of traditional IVNs such as CAN. By leveraging the high transmission speed and bandwidth of automotive ethernet, it is possible to transmit larger volumes of data in real-time for applications such as autonomous driving and infotainment systems.

Automotive ethernet includes SOME/IP, where one ECU requests events from other ECUs to exchange information; DoIP, which communicates with remote locations over the automotive ethernet to check the online status of an ECU; Additionally, AVTP is employed to handle time-sensitive data, ensuring synchronization of video playback through time synchronization mechanisms [5].

2.2 Attack Vector

We explain how the attacker accesses IVNs using our proposed method.

- **OTA Update Attack** OTA updates allow car manufacturers to update and enhance vehicular software remotely. exploiting OTA capability present a potential avenue for attackers to compromise vehicular control. Malefactors could upload a malicious update to the car manufacturer's OTA server, subsequently distributing this compromised update to the vehicles.
- **Supply Chain Attack** A supply chain attack involves adversaries infiltrating the automotive manufacturing process via specific components or software supply chains to embed malicious code or vulnerabilities. This subsequently allows for potential control over the car's internal network or control systems.
- **Physical Access** Physical approaches to gaining access method is also exist. Side-channel analysis is an attack technique used to extract information from a device that is not physically accessible. In cars that use smart keys, side-channel analysis allows attackers to detect and analyze the wireless communication signals of the smart key, enabling them to unlock the vehicle [18].

3 Related Work

Autonomous vehicles employ various methods to recognize the road conditions ahead and their surrounding environments, such as LiDAR sensors and V2X

communication technology. However, cameras are also one of the frequently used technologies [15]. Consequently, recent studies have highlighted the potential risks of vehicle camera attacks and proposed detection methods.

Jeong *et al.* [12] presents a method to detect replay attacks by manipulating the AVTP image data of vehicles in a testbed environment utilizing AVB Listener and AVB Talker. They released the 'Automotive ethernet Intrusion Dataset' [11]. They utilized a CNN model as the detection approach for this dataset, achieving an accuracy of 0.9955. Their dataset contains payloads of the MPEG codec.

Natasha *et al.* [2] introduced a real-time IDS using deep learning and machine learning techniques, utilizing the automotive ethernet Intrusion Dataset [11]. They performed anomaly detection on the dataset based on an unsupervised deep learning model, the autoencoder. Utilizing these models for unsupervised learning on the [11], they demonstrated performance metrics such as an F1-score of 0.98 for the LSTM autoencoder.

Han *et al.* [6] introduced a methodology for detecting abnormalities in automotive ethernet based on the Wavelet transform. This approach proves to be considerably more effective than methods utilizing ResNet or EfficientNet. However, it's worth nothing that the data used in this study was not extracted from vehicles in actual driving conditions.

Shibly *et al.* [16] generate attack data using a generative adversarial network (GAN) based on the dataset from [11]. They use this dataset to conduct detection using a feature-aware semi-supervised approach.

Jeong *et al.* [10] proposes an IDS named automotive ethernet Real-Time Observer (AERO), which analyzes traffic from various protocols in IVNs. This is an unsupervised IDS that learns only from normal condition data and detects anomalies.

Song *et al.* [13] explain the network attacks and defense methods that can occur in CAN networks and automotive ethernet.

Choi *et al.* [3] and Han *et al.* [14] propose models that can identify drivers based on patterns such as acceleration and braking, which are indicative of driving habits, using deep learning models.

4 Proof of Concept

An attacker can exploit previously introduced methods (§2.2) to compromise an ECU connected to an ethernet switch responsible for data transmission within IVNs utilizing automotive ethernet.

By taking control of a compromised ECU, the attacker can manipulate data transmitted to the autonomous driving module. Fig. 2 illustrates this process. Step ① represents the attacker's intrusion into the IVNs, step ② indicates the attacker's control over the ECU connected to the ethernet switch within the IVNs, and finally, step ③ shows the attacker executing an attack on the infotainment display and autonomous driving module.

The ethernet switch is responsible for transmitting data sent by ECUs or on-board cameras to their appropriate destinations. When necessary, the switch

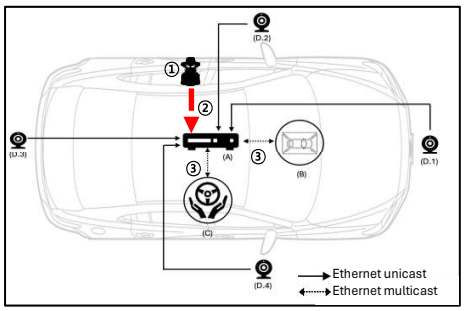
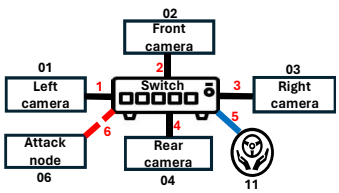


Fig. 2: Example for attack autonomous driving module. (A) is ethernet switch, (B) is infotainment display, (C) is autonomous driving module, D.1, D.2, D.3, D.4 are on-board camera.



(a) Example of ethernet switch.

MAC address	Port
:01	1
:02	2
:03	3
:04	4
:11	5

(b) MAC table before cache poisoning attack.

MAC address	Port
:01	1
:02	2
:03	3
:04	4
:06	6
:11	6

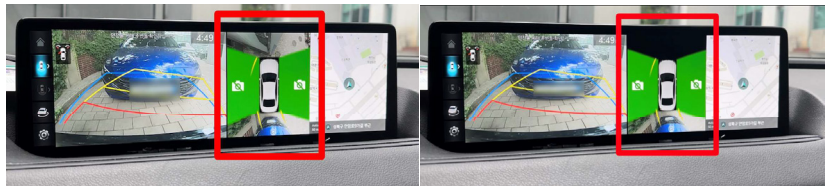
(c) MAC table after cache poisoning attack.

Fig. 3: Cache poisoning attack in vehicle.

uses multicast transmission to efficiently send data to multiple recipients within a specific group. This is particularly effective for modules such as the autonomous driving module or infotainment system, which require large amounts of data.

In this network environment, we execute an attack to disrupt the transmission of camera image data to the autonomous driving module or the infotainment system. Specifically, we targeted the infotainment display to demonstrate an attack where camera image data is transmitted abnormally. This abnormal data transmission attack can be explained by a cache poisoning attack [1]. Similarly, the autonomous driving module receives image data using multicast transmission, just like the infotainment display. Therefore, by employing the same attack method on the autonomous driving module, we can disrupt the normal transmission of camera image data to the module.

Fig. 3 illustrates the switch architecture and changes in the MAC address table that enable abnormal data transmission to the autonomous driving module using a cache poisoning attack. An attacker who has compromised an ECU connected to the ethernet switch manipulates data destined for the MAC ad-



(a) Under normal status (b) Under CIA status

Fig. 4: Demonstration of the camera inference attack. The infotainment system becomes blind when it goes under the attack.

dress at port 11, redirecting it through port 5 to themselves. After intercepting the data, the attacker mixes it with other data before transmitting it to the autonomous driving module. As shown in the "**Camera interference case**" part of Fig. 1, we transmit the data from one camera to an incorrect destination. To carry out this attack, we use the Rad-Galaxy equipment, which is capable of sniffing the vehicle's data. This equipment allows us to intercept and analyze the data traffic within the vehicle network, facilitating the execution of the attack. The Rad-Galaxy is a multipurpose ethernet tap and media converter tailored for automotive ethernet applications [9]. We configure the direction of the left camera data towards the front display using the Gateway Builder [1] tool provided by this device. Fig. 4 shows the results of executing a CIA on an actual vehicle. Fig. 4(a) depicts the normal operation of the cameras, with both the front and left cameras working correctly. Fig. 4(b) shows the outcome after attempting an attack using the Rad-Galaxy equipment.

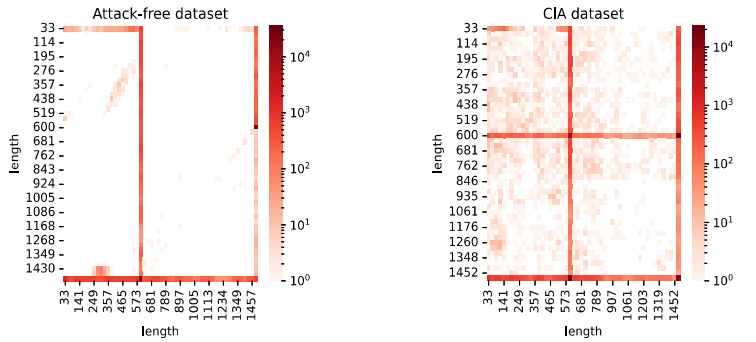
5 Dataset

5.1 Dataset Description

In this section, we describe the differences between the normal and anomalous datasets. We extract the data by using the rad-galaxy tool. The normal dataset comprises packets transmitting front camera data to the front display. The anomalous dataset includes packets where data from both the left-side and front cameras are simultaneously transmitted to the front display.

The normal dataset, collected over 30 minutes of driving, is approximately 5.86GB, containing 6,346,876 packets. The anomalous dataset, also gathered during a 30-minute drive on campus, is approximately 4.97GB, containing 5,450,620 packets. The images captured by the front, left, right and rear cameras are transmitted to their respective displays. Normal dataset consisting solely of front-camera image data are labeled as 0, while attack dataset containing both front and left-side image data are labeled as 1. As the goal is to detect deviations

[1] Rad-Galaxy program that changes the data transmission path of IVNs.



(a) Heatmap for normal dataset length sequence.

(b) Heatmap for abnormal dataset length sequence.

Fig. 5: Heatmap for length sequence 2-gram feature.

from this consistency, labeling all attack dataset, which include both frontal and left-side mixed attacks, as 1 can yield favorable results.

The image data captured by the camera is too large to be transmitted in a single packet, so it is fragmented before transmission. In our experiment, we use the FU-A fragmentation method [8]. Fragmentation ensures that packets do not exceed the MTU size. In a normal dataset, the fragmented packets create continuity. However, due to our attack, extraneous data are injected into the normal dataset, disrupting this continuity. As a result, the continuity observed in a normal dataset is altered. To demonstrate the difference in length sequence between normal and abnormal datasets, we utilize a heatmap for visualization.

Fig. 5 illustrates the differences in the length sequences between the normal and abnormal dataset. Each figure represents the 2-gram of the length sequences from the datasets. For example, for the 576, 1474, 160, 1474 sequence, 2-grams of $P = \{[576, 1474], [1474, 160], [160, 1474]\}$ can be expressed, and $P_i[0]$ is the X-axis. We can determine the frequency by specifying $P_i[1]$ as the Y-axis. Darker colors indicate a higher frequency of the corresponding 2-gram. It is evident that there are differences in the 2-gram patterns between the normal and abnormal datasets.

We employ data length as a feature that represents these characteristics and describe the method used for anomaly detection utilizing this feature. To effectively train the model on the continuity of packets, we need to determine a window size. The packet's sequence number has a maximum value of 255. Under normal status, the packet sequence increases sequentially from 0 to 255. We chose a window size corresponding to the packet's sequence number, ranging from 0 to 255. When an attack occurs, this sequence within the range of 0 to 255 displays irregular continuity.

6 Methodology for CIA Detection

We described potential attacks on vehicle cameras using automotive ethernet and the differences between datasets in normal and attack scenarios. In this section, we introduce the detection system for CIA. We propose a methodology for the model to learn the sequence of length values described earlier and detect sequences that differ from the learned ones.

6.1 Gated Recurrent Unit (GRU)

Utilizing the time-series data preprocessing, we employ a GRU deep learning model to detect camera interference attack in vehicle cameras [4].

The GRU model is specialized for processing time-series and sequential data. It operates using two key mechanisms. The update gate is used to determine how much past information to retain, whereas the reset gate decides how much past information to forget. Leveraging these characteristics, the model learns information about the sequence of payload data lengths, determining at which point to start forgetting, and conducts anomaly detection based on the learned content. The following equations govern the behavior of a GRU:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + br) \tag{1}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{2}$$

$$\widetilde{h_t} = tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \tag{3}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t} \tag{4}$$

The reset gate's weight W_r is operated with the previous state information h_{t-1} and the current state information x_t to determine how important h_{t-1} and x_t are. This operation involves an activation function. r_t represents the reset gate, which determines how much of the past information to forget in the computation process. This formulation allows the determination of how much of the past information to forget. The update gate operates on the previous state information and the current state information using the update gate's weights, followed by the application of an activation function. Using the reset and update gates, it is decided how much of the previous information to retain or forget. To achieve this, weights are computed for the current state, and a hyperbolic tangent activation function is used to express the weights within the range of -1 to 1. Finally, the update gate z_t is employed to determine how much information to blend between the previous state information h_{t-1} and the current state information h_t.

6.2 Data Preprocessing

For the model to be properly trained and accurately distinguish anomalies, preprocessing of the dataset is necessary. In section IV, we describe the dataset we extracted for our experiments. This dataset comprises camera data continuously extracted over time. Such data is referred to as time-series data, and while

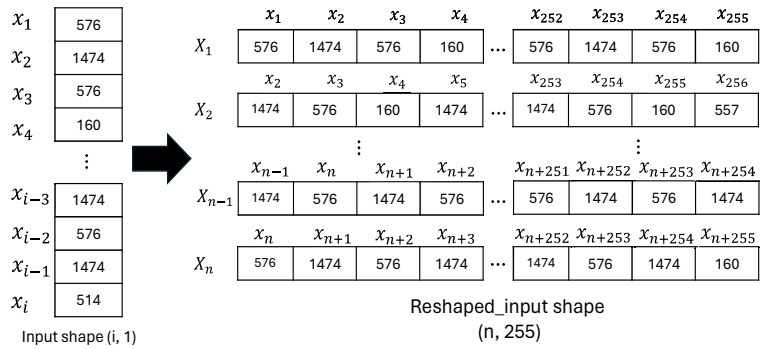


Fig. 6: Dataset reshaping for model input.

a single data point can be meaningful, analyzing a sequence of data can yield more significant information. As mentioned in §5.1, our dataset's data length encompasses the time-series information arising from fragmenting continuously captured image data.

In this section, we describe the preprocessing process for the dataset to ensure that the model accurately learns this consistency. We denote each packet's data length as x_i. A single window, segmented by the window size, is represented as X_n. Since a window comprises 255 rows, a single window can be expressed as $X_n = x_n ... x_{n+254}$. However, training the model in this form would only teach it to learn from a single payload data length, not a sequence. To train the model on a window containing 255 packets, we reshape the 255 rows into a single row composed of 255 columns. Fig. 6 illustrates the process of reshaping through a diagram. First, input shape for i in (i, 1) means the number of total packets. We reshape this input shape to (n, 255). Here, n represents the number of windows.

Preprocessing is also required for the label indicating whether the dataset is normal or anomalous. Like the data length, labels are also divided according to a window size of 255. We can represent the label for an individual packet as y_1. A set of labels divided by the window size, Y_i, has 255 labels and is composed as $Y_i = y_1 ... y_{255}$. To represent Y_i as a single label for the 255 labels, we use the average value of these 255 labels.

6.3 GRU-Based Camera Interference Attack IDS

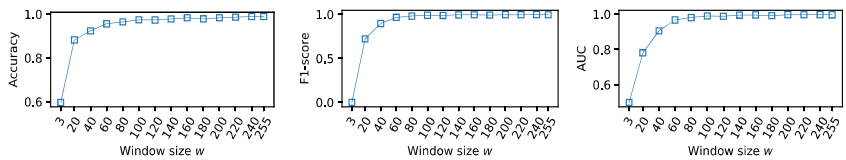
We describe a model capable of detecting attack using the previously introduced feature selection, data preprocessing and GRU model.

Input Layer: As explained in Fig. 6, the shape of the dataset for input is (255, 1). Each X_i contains values from x_i to x_{i+255}, resulting in 255 columns. This input shape allows the GRU model to analyze the continuity of the 255 values of x_i.

GRU: GRU learns the continuity of the input X_i. The update gate determines how much information about the continuity of the current input data length should be retained. In contrast, the reset gate decides how much of the previously learned information about data length continuity should be forgotten.

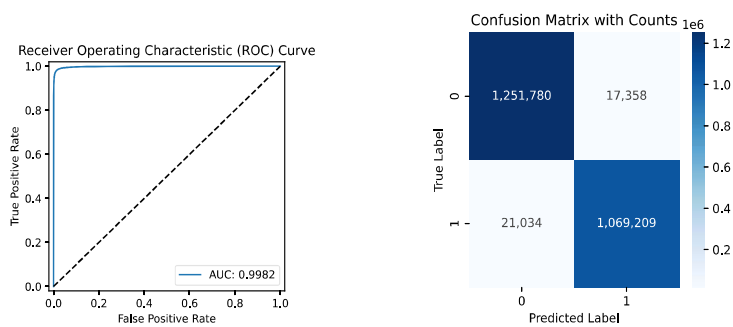
Output: Based on the GRU model's learning results, the output layer produces predictions about the input. By comparing the predicted results with Y_i, the model can determine whether it has correctly detected normal sequences as normal and abnormal data as anomalous.

7 Experiment Result



(a) Detection accuracy (b) F1-score (c) AUC

Fig. 7: Performance of CIA-IDS by window size



(a) ROC curve (b) Confusion matrix

Fig. 8: ROC curve and confusion matrix

In this study, we elucidate the detection results of abnormal camera data transmission scenarios using a GRU-based IDS facilitated by the previously described sliding-window data preprocessing technique. Our experimental setup encompasses an Intel i7-10700k CPU@3.80GHz, 64GB of memory and an NVIDIA RTX 3070 GPU, which are utilized for both training and evaluation. For the dataset generated through our preprocessing steps, we allocate 80% of the data

for training and 20% for testing purposes. We explore various window sizes to assess detection performance alongside the impact of window size on model complexity. Fig. 7 illustrates the model's accuracy, F1-score and AUC for window sizes ranging from 3 to 255, in increments of 20.

In our experiments, we observe that despite the satisfactory performance at a smaller window size, a comparison of the inference times at window sizes of 140 and 255, yielding $0.002\text{ms}/X$ and $0.009\text{ms}/X$ respectively, shows negligible difference. Therefore, we have selected a window size of 255 for our IDS. Fig. 8(a) presents the ROC curve for a window size of 255, exhibiting an AUC value of 0.9982.

Furthermore, Fig. 8(b) demonstrates the detection outcomes of the IDS on the test dataset, where the X-axis represents the model's predicted results, whether the window is classified as normal or containing an attack, and the Y-axis indicates the actual labels of the windows, whether they are truly normal or anomalous, forming a confusion matrix. The true positive rate, which achieves a value of 0.99, indicates outstanding performance. Table 1 shows the performance metrics of the IDS.

Table 1: Performance Metrics of the CIA-IDS

Accuracy	AUC	F1-score	FPR	FNR	TPR	TNR
0.98848	0.99823	0.99824	0.0137	0.0193	0.9807	0.9863

8 Conclusion

In this study, we successfully demonstrated a proof-of-concept for detecting CIA on a Hyundai Genesis G80, leveraging a GRU-based IDS designed for automotive ethernet. Our approach utilized a novel sliding-window data based on packets's length value sequence. The experimental results showed that our IDS, with a window size of 255, achieved high detection accuracy, an impressive AUC of 0.9982 and a true positive rate of 0.99, indicating its exceptional ability to differentiate between normal and anomalous data. It has the advantage of carrying out attacks targeting actual vehicles and using the corresponding data, but has the disadvantage of being able to apply IDS only in a network structure fragmented by the FU-A method and using H.264 encoding. In upcoming works, limitations can be overcome if the actual image is recovered using the payload of the data set and detection is performed based on the image.

Acknowledgement. This work was supported by the 2021 Autonomous Driving Development Innovation Project of the Ministry of Science and ICT, Development of technology for security and ultrahigh-speed integrity of the next-generation internal network of autonomous vehicles under Grant 2021-0-01348.

References

1. Abad, C.L., Bonilla, R.I.: An analysis on the schemes for detecting and preventing arp cache poisoning attacks. In: 27th international conference on distributed computing systems workshops (ICDCSW'07). pp. 60–60. IEEE (2007)

2. Alkhatib, N., Mushtaq, M., Ghauch, H., Danger, J.L.: Unsupervised network intrusion detection system for avtp in automotive ethernet networks. In: 2022 IEEE intelligent vehicles symposium (IV). pp. 1731–1738. IEEE (2022)
3. Choi, Y.A., Park, K.H., Park, E., Kim, H.K.: Unsupervised driver behavior profiling leveraging recurrent neural networks. In: Information Security Applications: 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11–13, 2021, Revised Selected Papers 22. pp. 28–38. Springer (2021)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
5. working group, I.P.: Ieee standard for a transport protocol for time-sensitive applications in bridged local area networks. https://standards.ieee.org/standard/1722-2016.html (2016)
6. Han, M., Kwak, B., Kim, H.: Tow-ids: Intrusion detection system based on three overlapped wavelets for automotive ethernet. IEEE Transactions on Information Forensics and Security **18**, 411–422 (2023)
7. Hank, P., Müller, S., Vermesan, O., Van Den Keybus, J.: Automotive ethernet: in-vehicle networking and smart mobility. In: Proceedings of the Conference on Design, Automation and Test in Europe. p. 1735–1739. DATE '13, EDA Consortium, San Jose, CA, USA (2013)
8. IETF: Fragmentation units (fus). https://datatracker.ietf.org/doc/html/rfc6184 (2011), accessed: 2024-08-07
9. Intrepidcs.com: Intrepid, rad-galaxy. https://www.intrepidcs.com/rad-galaxy/, accessed: 2024-08-07
10. Jeong, S., Kim, H., Han, M., Kwak, B.: Aero: Automotive ethernet real-time observer for anomaly detection in in-vehicle networks. IEEE Transactions on Industrial Informatics pp. 1–12 (2023)
11. Jeong, S., Jeon, B., Chung, B., Kim, H.K.: Automotive ethernet intrusion dataset (2021). https://doi.org/10.21227/1yr3-q009, https://dx.doi.org/10.21227/1yr3-q009
12. Jeong, S., Jeon, B., Chung, B., Kim, H.K.: Convolutional neural network-based intrusion detection system for avtp streams in automotive ethernet-based networks. Vehicular Communications **29**, 100338 (2021)
13. Kim, K., Kim, J., Jeong, S., Park, J., Kim, H.: Cybersecurity for autonomous vehicles: Review of attacks and defense. Computers & Security **103**, 102150 (2021)
14. Kwak, B., Han, M., Kim, H.: Driver identification based on wavelet transform using driving patterns. IEEE Trans. Ind. Informatics **17**(4), 2400–2410 (2021)
15. Ondruš, J., Kolla, E., Vertaľ, P., Željko Šarić: How do autonomous cars work? Transportation Research Procedia **44**, 226–233 (2020). https://doi.org/https://doi.org/10.1016/j.trpro.2020.02.049, https://www.sciencedirect.com/science/article/pii/S2352146520300995, lOGI 2019 - Horizons of Autonomous Mobility in Europe
16. Shibly, K., Hossain, M., Inoue, H., Taenaka, Y., Kadobayashi, Y.: A feature-aware semi-supervised learning approach for automotive ethernet. In: Proc. IEEE CSR. pp. 426–431. IEEE (2023)
17. Tesla, I.: Transitioning to tesla vision. https://www.tesla.com/support/transitioning-tesla-vision, accessed: 2024-08-07
18. Wouters, L., Van den Herrewegen, J., D Garcia, F., Oswald, D., Gierlichs, B., Preneel, B.: Dismantling dst80-based immobiliser systems. IACR Transactions on Cryptographic Hardware and Embedded Systems **2020**(2), 99–127 (2020)

One-class Classification-based Position Falsification Detection System in C-ITS Communication Network

Ilhwan JI[1[0009-0003-9027-1305]], ChangJin Choi[1[0009-0004-9136-763X]], Cheolho Lee[2[0009-0005-1539-0528]], Seungho Jeon[1[0000-0002-7116-6062]], and Jung Taek Seo[1*[0000-0003-0971-8548]]

[1] Gachon University, Seongnam-daero 1342, Seongnam-si, Republic of Korea
{ilhwan1013, ethobis, shjeon90, seojt}@gachon.ac.kr
[2] ENKI WhiteHat167, Songpa-daero, Songpa-gu, Seoul, Republic of Korea
{chlee}@enki.co.kr

Abstract. Cooperative-Intelligent Transport Systems (C-ITS) is a system for ensuring road safety and user safety, such as traffic congestion and traffic accidents, by sharing traffic risk information in advance through real-time two-way data transmission between road traffic components. The main services provided by C-ITS, such as traffic jam and traffic accident prevention, are performed based on information derived by collecting and analyzing vehicle location information at the C-ITS Center. For this reason, if an attacker falsifies position data on the communication channel from the vehicle to the C-ITS Center with malicious purposes, most of the C-ITS services cannot operate normally, causing financial losses and casualties. This research proposes an effective position falsification attack detection system in C-ITS based on the environmental analysis of C-ITS. For the purpose of developing cyberattack detection systems, it is very difficult to collect training datasets by conducting real-world cyberattacks in most IT infrastructure environments, including C-ITS environments. For this reason, the C-ITS target position falsification attack detection system proposed in this research utilizes the method of training only normal data and detecting position falsification based on that information. For the position falsification attacks detection system proposed in this research, high performance of precision 99.97%, recall 100%, and F1 Score 99.98% was obtained using VeReMi and BurST-ADMA datasets, which are datasets containing normal data and position falsification data collected from ITS and VANET environment simulators.

Keywords: C-ITS, AI, Anomaly Detection.

1 Introduction

Cooperative-Intelligent Transport Systems (C-ITS) collects road and user information such as real-time traffic information, road surface weather information, and vehicle collision prevention in real-time for the purpose of promoting road safety and user safety. C-ITS is a system to ensure road safety and user safety by generating and transmitting information to predict and respond to traffic congestion and traffic accidents based on the collected information [1]. Due to the development of information and communication technology and the increase in vehicle productivity, the vehicle road transportation system is becoming more complex, and the number of vehicles is increasing, resulting in

traffic congestion and traffic accidents. For this reason, the development and application of C-ITS to ensure road safety and user safety based on two-way communication of data between road traffic components is actively being developed and applied [2].

The main services provided by C-ITS, such as traffic jam and traffic accident prevention, are performed based on information derived by collecting and analyzing vehicle location information at the C-ITS Center. If an attacker falsifies position information data on the communication channel from the vehicle to the C-ITS Center with malicious purposes, most of the C-ITS services may not operate normally, causing financial damage and casualties. For this reason, cybersecurity technology research is being conducted to proactively detect and respond to position information manipulation attacks in C-ITS. Mahshid et al. [3], Hawlader et al. [4], Ilango et al. [5] proposed a detection system for position falsification attacks in C-ITS and VANET environments based on machine learning (ML) algorithms. As a result of the experiment of each research, high detection performance against position falsification attacks was derived. However, most research, including Mahshid et al. [3], Hawlader et al. [4], Ilango et al. [5] simply aim to improve classification performance and how data is classified into normal and attack without considering C-ITS environmental features. C-ITS includes a wide variety of communication sections, and the number of communication sections is also large.

For this reason, it is necessary to derive C-ITS environment and network features such as data communication section, communication method, communication data standard, and communication data type through environmental analysis of C-ITS, and to develop and apply a position falsification attack detection system that effectively collects position information data and detects attacks based on this.

In this research, we derive C-ITS environment features such as communication section, communication protocol, and data type through C-ITS environment analysis, and propose a position falsification attack detection system to effectively detect position falsification attacks occurring in C-ITS based on these features. The position falsification attack detection system presented in this research is located between the C-ITS Center and the Roadside Station to effectively collect vehicle location data and detect attacks on the location data before it is delivered to the C-ITS Center. The system consists of a vehicle position data parser, a feature preprocessor, and a one class classification-based cyberattack detector. The vehicle position data parser extracts location-related data for cyberattack detector training from network packets containing vehicle location information sent from the vehicle to the C-TIS Center and converts it into a consistent format. The feature preprocessor performs missing value removal, data cleaning, normalization, and label-encoding to improve the training and attack detection efficiency of the position falsification attack detector. The one class classification-based cyberattack detector detects position falsification attacks based on input data. To evaluate the detection performance of a position falsification attack detection system with the above configuration, we used the VeReMi and BurST-ADMA datasets, which are datasets containing normal data and position falsification data collected from Intelligent Transport Systems (ITS) and Vehicular ad-hoc network (VANET) environment simulators. The contributions of this research are as follows:

- We analyze the C-ITS environment and present the security threats that can arise in the C-ITS environment due to position falsification.
- We propose a One-class Classification based Position Falsification Attacks Detection System for the detection of position falsification attacks in C-ITS Center data collection section.
- For the One-class Classification based Position Falsification Attacks Detection System presented in this research, a high performance of precision 99.97%,

recall 100%, and F1 Score 99.98% is obtained using VeReMi and BurST-ADMA datasets, which are datasets containing normal data and position falsification data collected from ITS and VANET environment simulators.

The remainder of this paper is organized as follows. In Section 2, we analyze the related work on detecting position falsification attacks analyze. In Section 3, we C-ITS. Section 4 presents a position falsification attack detection system in C-ITS based on one class classification. In Section 5, we evaluate the detection performance of our system for detecting position falsification in C-ITS and conduct a comparative analysis with existing works. In section 6, the conclusions and future research directions are presented.

2 Related works

Various research has been conducted to detect location data forgery attacks on C-ITS and vehicular ad hoc networks (VANETs). In particular, cyberattack detection and classification using artificial intelligence (AI) is being actively researched for fast and accurate detection of position falsification attacks on large amounts of position data.

Sharma et al. [6] proposed a machine learning-based detection scheme for position spoofing attack detection in vehicular ad-hoc network (VANET). In this research, they generated augmented data based on two BSM information instead of a single BSM information and performed position spoofing attack detection based on the augmented data. In this research, KNN, RF, Naïve Bayes, and DT algorithms were used to verify the performance of the proposed machine learning-based position spoofing attack detection method using the VeReMi dataset[7], which is generated based on BSM information occurring in VANETs. Among the four algorithms, the KNN algorithm performed the best in detecting position spoofing attacks, with a precision of 99.3% and a recall of 98.38%.

Le et al. [8] proposed a detection scheme for position falsification based on n consecutive position information of vehicles derived by applying the n-sequence technique to position information generated in VANETs. In this research, cyberattacks are detected based on KNN and SVM algorithms for data containing n consecutive position information extracted. VeReMi dataset is used to detect the position falsification attack proposed in this paper. The experimental results show that the classification performance of position falsification attacks is 99.7% precision and 99% recall.

Anyanwu et al. [9] proposed a Randomized Search Optimization Ensemble-based Falsification Detection Scheme (RSO-FDS) for detecting Basic Safety Message (BSM) falsification attacks that contain vehicle position information. The RSO-FDS proposed in this paper utilizes the information in the BSM to detect falsification attacks based on ensemble-based RF. To evaluate the detection performance of RSO-FDS, we utilized VeReMi dataset, BurST-ADMA dataset [10], and V2X-Fabrication dataset, and obtained detection performance of 99.50%, 99.89%, and 99.99%.

Arif et al. [2] proposed an ML-based cyberattack detection approach for detecting Content Poisoning Attacks (CPA) on vehicular networks. In this paper, LR, DT, KNN, RF, and GNB algorithms are used to detect CPA. The BurST-ADMA dataset is utilized to validate the proposed CPA detection algorithm. In the experiment, the best detection performance was obtained in RF, with 100% accuracy, 100% recall, and 100% precision.

There is a wide range of research on cyberattack detection to detect position falsification attacks on C-ITS and automotive networks. However, most of these research do not consider the application of cyberattack detection systems to C-ITS environments, but simply consider how to detect and classify cyberattacks. C-ITS includes a wide variety of communication sections and the number of communication sections is also large, so it is difficult to apply a position falsification detection method to all

communication sections. For this reason, it is necessary to derive C-ITS environmental features through a systematic analysis of the C-ITS environment and to research the development and application of a position falsification attack detection system in C-ITS with effectiveness and efficiency that reflects these features.

3 Analysis of Cooperative-Intelligent Transport Systems

ITS refers to a transportation system that scientizes and automates the operation and management of transportation systems and improves the efficiency and safety of transportation by developing and utilizing advanced transportation technologies and traffic information such as electronic control and communication for transportation vehicles and transportation facilities. The ITS collects and processes the collected information at the control center and delivers the information one-way to the targets that need the information, such as vehicles, bus stops, and traffic flow information indicators. However, ITS cannot generate and deliver information that reflects real-time field conditions due to one-way information transmission. From the past to the present, the vehicle road transportation system is becoming more complex, and as the number of vehicles increases, the incidence of traffic jams and traffic accidents is increasing. For this reason, there is a growing demand for real-time traffic information, road surface weather information, vehicle collision prevention, and other real-time collected information based on field conditions [11]. Therefore, C-ITS is being developed and applied to collect and analyze field information based on two-way communication of data between road traffic components and transmit the derived information to road traffic components to prevent traffic congestion and traffic accidents.

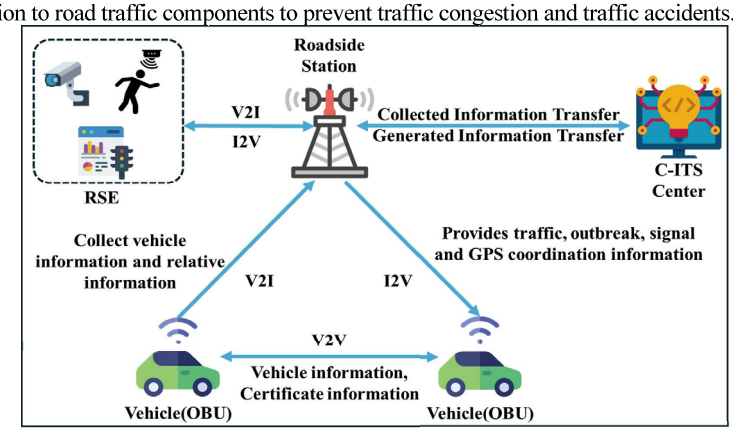


Fig. 1. Structure of C-ITS

C-ITS is a system that enables real-time two-way data transmission between road traffic components to share traffic risk information and respond in advance. Fig 1. shows the structure of the C-ITS. C-ITS consists of the vehicle's on-board unit (OBU), various sensors and road side equipment (RSE), the C-ITS Center that manages and distributes information, and the Roadside Station that performs data transmission between OBU, RSE, and C-ITS Center OBUs transmit vehicle location information and vehicle basic information to other OBUs and RSEs. The RSE provides the vehicle with data related to road conditions, traffic signals, accident warnings, and various services. The C-ITS center

receives the data collected from OBUs and RSEs through the Roadside Station. The C-ITS Center analyzes the received data to produce information for traffic congestion and traffic accident response and transmits the information to OBUs and RSEs through the Roadside Station.

C-ITS has three communication segments: Vehicle-to-Vehicle (V2V), which is vehicle-to-vehicle communication; Vehicle-to-Infrastructure (V2I), which is vehicle-to-infrastructure communication; and Vehicle-to-Everything (V2X), which is vehicle-to-everything communication with other vehicles, roads, RSE, and other components. The V2X communication section including V2V and V2I within C-ITS performs WAVE, a wireless communication method using the SAE J2735 message standard, and Cellu-lar-V2X communication using LTE and 5G communication based on the IEEE 1609.3 message standard. The communication section between Roadside Station and C-ITS Center communicates by optical communication method based on SAE J2735 or IEEE 1609.3 message standard because it needs to transmit a large amount of data quickly.

The main services of C-ITS, including traffic congestion and traffic accident response, are performed based on the analysis of vehicle position data collected by the C-ITS Center. For this reason, if an attacker with malicious intent falsifies the position data transmitted between OBU, RSE, and C-ITS Center, it is difficult to perform the main services of C-ITS. By falsifying the position information of a vehicle, an attacker can manipulate traffic information on a particular stretch of road to cause traffic jams. For this reason, the integrity of the position data collected by C-ITS Center is very important. As the C-ITS Center is a system operated for the safety and efficiency of the road traffic environment, a cyberattack on the C-ITS Center can cause not only financial damage but also casualties[2, 12]. For this reason, cybersecurity technologies are needed to proactively detect and respond to position falsification attacks on the information collected by the C-ITS Center.

4 One-class Classification based Position Falsification Detection System

In this section, we describe a position falsification attack detection system in C-ITS based on one class classification. The proposed method detects position data that has been falsified for cyberattacks based on training information of vehicle position data that occurs in the normal operation of C-ITS. Section 3.1 provides an overview of position falsification attack detection in C-ITS. Section 3.2 describes the position data parser. Section 3.3 describes the feature preprocessor extracted from encrypted traffic to improve cyberattack detection efficiency and cyberattack detection performance. Section 3.4 describes a position falsification attack detector based on training information about vehicle position data generated in the normal operation of C-ITS.

4.1 Overview

For most IT infrastructure environments, including C-ITS environments, it is very difficult to collect datasets for training by conducting actual cyberattacks[13]. For this reason, it is necessary to apply a method to detect position falsification attacks based only on training information from data generated in the normal operation of C-ITS.

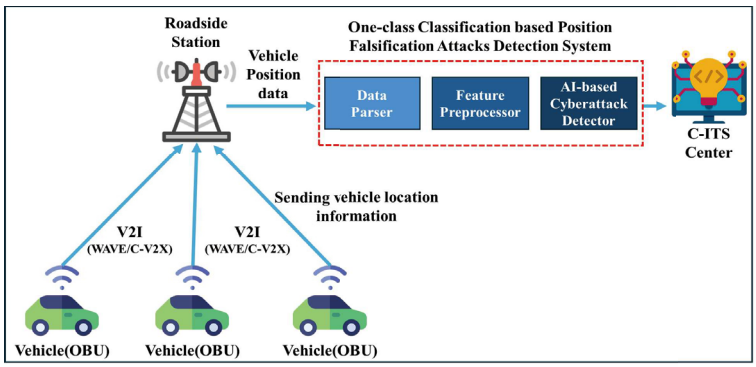


Fig 2. Position falsification attack detection system application diagram in C-ITS

Fig. 2. shows an overview of the position falsification attack detection system for position data collected by C-ITS Center using V2I communication based on the one-class classification proposed in this paper. The position falsification attack detection system in C-ITS based on one class classification is located between the Roadside Station and the C-ITS Center, and detects position falsification attacks based on the position information data collected through V2I communication and delivered to the C-ITS Center through the Roadside Station. Since the data collected from OBU and RSE are all transmitted to C-ITS Center through Roadside Station, it is appropriate to apply the position falsification attack detection system for that section. The position falsification attack detection system consists of a vehicle position data parser, feature preprocessor, and one class classification-based cyberattack detector, and processes the data in the following order for the collected data.

1. **Vehicle position data parser**: Extract and convert data collected by position falsification attack detection systems into a consistent format to be used for training from network packets and detecting cyberattacks.
2. **Feature Preprocessor**: Preprocess the input features to improve the accuracy and efficiency of training and anomaly detection of cyberattack classification models.
3. **One Class Classification-based Cyberattack Detector**: It receives preprocessed data as input, learns, and detects cyberattacks. When detecting a cyberattack, if the difference between the input data value and the model's predicted value is greater than the threshold, it is detected as a cyberattack.

4.2 Vehicle position data parser

Vehicle position data parser extracts and converts data for cyberattack detector training into a consistent format from network packets containing vehicle position information transmitted via optical communications based on SAE J2735 or IEEE 1609.3 message standards from the Roadside Station to the C-TIS Center. The Vehicle position data parser parses information related to vehicle location data, such as GPS coordinates, velocity, acceleration, and noise values, and derives features from that information.

4.3 Feature Preprocessor

The Feature Preprocessor performs data feature preprocessing to improve the training and attack detection efficiency of the position falsification detector. The feature preprocessing step is very important because noise in the data can degrade performance [14]. The input value for training and anomaly detection of One Class Classification-based Cyberattack Detector should be numerical data. Therefore, string data is converted into integer category values through label-encoding or one-hot encoding for features that are not numerical values extracted from the vehicle position data parser. In addition, since many missing values have a negative impact on model training, we replace the missing values with specific values to remove missing values in the data. Then, Feature Selection is used to remove uncorrelated features or features that have direct information about the time series to avoid overfitting and improve the performance of the model. Finally, normalization is performed to unify the data range of each feature to prevent the learning and anomaly detection efficiency from decreasing due to the difference in the range of values between features.

4.4 One Class Classification-based Cyberattack Detector

One class classification-based cyberattack detector trains only the vehicle position information data that occurs in normal operation among the data collected by the C-ITS Center. The trained cyberattack detector detects position data information that has been falsified by cyberattacks based on normal position data information. Supervised learning-based anomaly detection systems have the advantage of relatively high anomaly detection rates and the ability to classify attack types within attacks. However, for the development of cyberattack systems, it is very difficult to collect datasets for training by conducting actual cyberattacks on most IT infrastructure environments, including C-ITS environments [13]. For this reason, we only train our model on normal data collected in a C-ITS environment. The cyberattack detector is trained to produce the smallest possible error on normal data. If such cyberattack data is input to the cyberattack detector, a relatively large error value is derived by cyberattack data that is different from normal data, and if the derived error value is larger than the set threshold, the data is detected as cyberattack data.

5 Evaluation

In this section, we evaluate the one class classification-based position falsification attacks detection system proposed in Section 3 using a dataset containing vehicle position data from a C-ITS environment. Section 4.1 describes the dataset containing vehicle position data from the C-ITS environment used in our experiments. Section 4.2 describes the experimental environment, model structure, and preprocessing. Section 4.3 describes various metrics to evaluate the performance of the proposed attack detection system. In Section 4.4, we verify the cyberattack detection performance of our one-class classification-based position falsification attacks detection system.

5.1 Dataset Description

5.1.1 VeReMi dataset

The VeReMi dataset [7] is a dataset developed to evaluate the effectiveness of falsification attack detection methods in Vehicle Ad-HocNetworks (VANETs). The VeReMi dataset contains five types of position falsification attacks and normal data: Constant attack (attack by retransmitting a fixed location), Constant Offset at-tack (attack by changing the fixed location by a certain value), Random attack (attack by transmitting a random location), Random Offset attack (attack by transmitting the location of neighboring vehicles from a random location), and Eventual Stop attack (attack by repeating the same location over a period of time). The dataset has 21 features including vehicle ID, position information, speed, acceleration, etc. 382,952 training data (normal data: 382,952) and 198,576 test data (normal data: 108,530 / attack data: 90,046) were used in the training and validation experiments of the position falsification attack detection system proposed in this research.

5.1.2 BurST-ADMA dataset

The BurST-ADMA dataset [10] was developed to evaluate the performance of position falsification attack detection in C-ITS. The BurST-ADMA dataset contains seven types of position falsification attacks and normal data, including Constant Random Position, Positive Position Offset, Negative Position Offset, Constant Random Speed, Positive Speed Offset, Negative Speed Offset, and Reversed Heading. The dataset has 8 features including vehicle ID, location information, speed, acceleration, etc. 101,501 training data (normal data: 101,501) and 99,999 test data (normal data: 77,627 / attack data: 22,372) are used in the training and validation experiments of the position falsification attack detection system proposed in this paper.

5.2 Experimental Setup

Computational environment. The experiments were conducted using Ubuntu 20.04.6 LTS operating system, Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64GB RAM, Tesla V100S PCIe 32GB GPU and python3 development environment.

Preprocessing. In this experiment, data preprocessing was performed to improve the training of cyberattack detection models and the efficiency of attack detection. To remove missing values in VeReMi dataset and BurST-ADMA dataset used in this experiment, we replaced all missing values with zero. For cyberattack detection experiments, we removed the labels representing the five attack types in the VeReMi dataset and replaced all the labels (1-7) of the seven types of data falsification attack types in the BurST-ADMA dataset with 1. Among the features in the BurST-ADMA dataset, id consists of a combination of strings and integers. We separated the Id feature into two features, string and number, so that the cyberattack detection model can train it. Then, label-encoding was performed on the string features to convert them into integers. In addition, to prevent the learning and anomaly detection efficiency from decreasing due to the difference in the range of values between the features of VeReMi dataset and BurST-ADMA dataset, we normalized the data and applied the MaxAbs scaling method, which changes the minimum value of each data to -1 and the maximum value to 1.

Hyper-parameter settings. In this experiment, LSTM based AutoEncoder (AE-LSTM) algorithm is used to apply the one class classification-based position falsification attack detection system to detect cyberattacks based on only normal data presented in section 4. The AE-LSTM model used for training and validation on VeReMi dataset and BurST-ADMA dataset is designed as an encoder and decoder consisting of LSTM layers. Table 1 describes the detailed architecture of the model. We used tanh as the activation function for each layer in both models. The optimizer for all models is Adam, and the learning rate is 0.001.

Table 1. Detailed model architecture trained on each dataset.

		VeReMi		BurST-ADMA	
	Layer Type	**Output Shape**	**Activation**	**Output Shape**	**Activation**
Encoder	LSTM	(1, 7)	Tanh	(1, 19)	Tanh
	LSTM	(1, 64)	Tanh	(1, 64)	Tanh
	LSTM	(1, 32)	Tanh	(1, 32)	Tanh
	RepeatVector	(1, 32)	-	(1, 32)	-
Decoder	LSTM	(1, 32)	Tanh	(1, 32)	Tanh
	LSTM	(1, 64)	Tanh	(1, 64)	Tanh
	LSTM	(1, 7)	Tanh	(1, 19)	Tanh
	TimeDistributed	(1, 7)	Linear	(1, 19)	Linear

5.3 Performance Indicators

The performance metrics of the anomaly detection model in this experiment include Accuracy, which is an evaluation of how well the detection system correctly classifies normal data as normal data and anomaly data as anomaly data in the collected data by defining the ratio of the number of correctly classified data to the total number of data; Recall, which is an evaluation of how well the system finds the data to detect; and Precision, a metric that evaluates the percentage of data predicted to be anomaly by the system that is actually anomaly; and F-1 Score, a metric used to measure the overall Accuracy of the model. We also used the receiver operating characteristic area under the curve (ROC-AUC), an evaluation metric that indicates the rate of change in the true positive rate (TPR) and false positive rate (FPR).

5.4 Performance Evaluation

In this section, we evaluate the detection performance of each of the location data forgery attack detection algorithms presented in Section 4.2. The AE-LSTM-based data forgery attack detection model is trained only on position information data extracted from normal operation, and its detection performance is verified using the validation dataset. Fig. 3. shows the training and validation losses of VeReMi dataset and BurST-ADMA dataset for the AE-LSTM-based position falsification attack detection model.

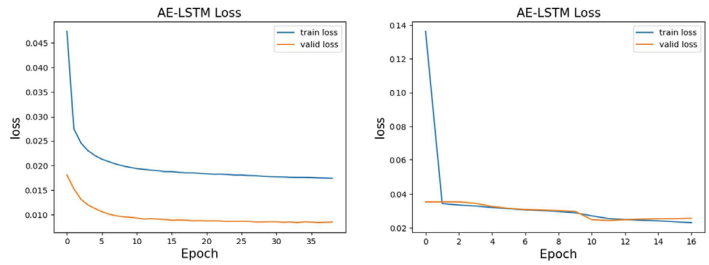


Fig 3. Learning and validating loss of AE-LSTM-based position falsification attack detection model for (A)VeReMi dataset, (B)BurST-ADMA dataset

In Fig. 3, the AE-LSTM-based data tampering attack detection model was trained on both datasets in the direction of decreasing validation loss. We obtained a validation loss of 0.0085 for the VeReMi dataset and 0.0257 for the BurST-ADMA dataset.

Fig. 4. and Fig. 5. show the detection results of AE-LSTM-based position falsification for VeReMi dataset and BurST-ADMA dataset, respectively.

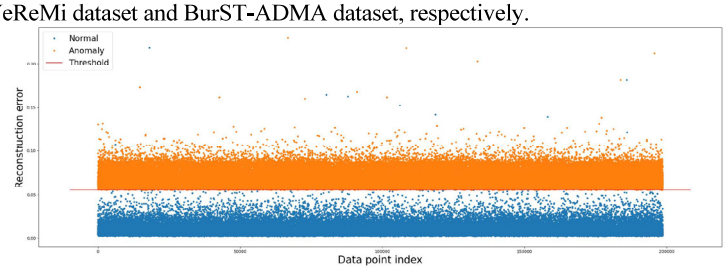


Fig 4. Results of Position Falsification Attacks detection on VeReMi dataset

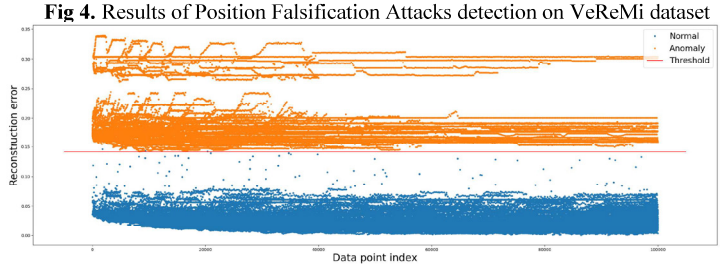


Fig 5. Results of Position Falsification Attacks detection on BurST-ADMA dataset

The blue dots in Fig. 4. and Fig. 5. represent normal data, while the orange dots represent position falsification attack data. The red straight line indicates the threshold for anomaly detection. The threshold value is set to the value when the precision and recall of the anomaly detection model are equal. High precision means fewer false positives, which detect normal data as anomalous, and high recall means fewer false negatives, which miss anomaly data [15]. Since these two performance metrics typically have a trade-off relationship, this study derived the thresholds in the same way as above to ensure an unbiased consideration of both [16]. The position falsification attack detection system

detects the data as cyberattack data if the error between the input and output data is higher than the threshold of each model. In each figure, you can see that the difference between normal data (blue dots) and cyberattack data (orange dots) is identifiable.

Table 2. shows the performance comparison of the one-class classification-based position falsification detection system proposed in this research with existing research on position falsification detection for C-ITS and VANETs.

Table 2. Comparison of existing models, including ours.

	VeReMi			BurST-ADMA		
reference	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
KNN [3]	0.993	0.9838	0.951	None	None	None
SVM [5]	0.997	0.99	0.990	None	None	None
Ensemble based RF [9]	0.9966	0.9916	0.9916	**1**	0.995	0.998
RF [2]	**-**	**-**	**-**	**1**	**1**	**1**
Our model	**0.9988**	**0.9988**	**0.9988**	0.9997	**1**	0.9998

As shown in Table 2., the one-class classification-based position falsification detection system proposed in this research achieved a high performance of 99.88% on all detection performance metrics for the VeReMi dataset. In addition, the proposed model achieved a high performance of 99.97% precision, 100% recall, and 99.98% F1 score for BurST-ADMA. We evaluated the detection performance on both datasets and found that it is comparable to or better than the performance of existing research based on supervised learning algorithms.

6 Conclusion and Future works

C-ITS is a system that enables real-time two-way data transmission between road traffic components to share and respond to traffic risk information in advance. The main services provided by C-ITS, such as traffic congestion and traffic accident prevention, are performed based on information derived by collecting and analyzing vehicle position information at the C-ITS Center. For this reason, in the event of a position falsification attack on the data collection section of the C-ITS Center, most of the C-ITS services may not operate normally, causing financial and human casualties In this research, we proposed a one-class classification-based position falsification attack detection system to position falsification based on the position data extracted from the normal operation status of C-ITS. To validate the proposed attack detection system, we used the VeReMi dataset and BurST-ADMA dataset, which contain position falsification attack data for ITS and VANET environments. The detection result of VeReMi dataset is 99.88% in all detection performance indicators, and the detection result of BurST-ADMA dataset is 99.97% in precision, 100% in recall, and 99.98% in F1 Score. Based on the experimental results, we conclude that it is possible to detect most of the attacks when applying the position falsification attack detection system presented in this research to the C-ITS environment.

The one-class classification-based position falsification attack detection system proposed in this paper is designed to detect only position falsification attack on vehicle location information collected through V2I communication sections, and does not consider position falsification attacks on V2V communication sections in C-ITS.

In future works, we will research the application of a position falsification attack detection system for all communication sections in the actual C-ITS environment and considerations for application.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2024-00331950, Technology Development for Vulnerability Assessment and Cybersecurity Attack&Defense Training on the Next-generation Intelligent Transport System(C-ITS))

References

1. Lokaj, Z., Šrotýř, M., Vaniš, M., Mlada, M.: Methodology of functional and technical evaluation of cooperative intelligent transport systems and its practical application. Applied Sciences 11, 9700 (2021)
2. Magsi, A.H., Ghulam, A., Memon, S., Javeed, K., Alhussein, M., Rida, I.: A machine learning-based attack detection and prevention system in vehicular named data networking. Comput. Mater. Contin 77, 1445-1465 (2023)
3. Behravan, M., Zhang, N., Jaekel, A., Kneppers, M.: Intrusion Detection Systems Based on Stacking Ensemble Learning in VANET. In: 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 1-7. IEEE, (2022)
4. Hawlader, F., Boualouache, A., Faye, S., Engel, T.: Intelligent misbehavior detection system for detecting false position attacks in vehicular networks. In: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE, (2021)
5. Ilango, H.S., Ma, M., Su, R.: Novel Position Falsification Attacks Detection in the Internet of Vehicles using Machine Learning. In: 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 955-960. IEEE, (2022)
6. Saudagar, S., Ranawat, R.: An Amalgamated Novel IDS Model for Misbehaviour Detection using VeReMiNet. Computer Standards & Interfaces 88, 103783 (2024)
7. Van Der Heijden, R.W., Lukaseder, T., Kargl, F.: Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In: Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part I, pp. 318-337. Springer, (2018)
8. Le, A., Maple, C.: Shadows don't lie: n-sequence trajectory inspection for misbehaviour detection and classification in vanets. In: 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pp. 1-6. IEEE, (2019)
9. Anyanwu, G.O., Nwakanma, C.I., Lee, J.-M., Kim, D.-S.: Falsification Detection System for IoV Using Randomized Search Optimization Ensemble Algorithm. IEEE Transactions on Intelligent Transportation Systems 24, 4158-4172 (2023)
10. Amanullah, M.A., Chhetri, M.B., Loke, S.W., Doss, R.: BurST-ADMA: Towards an Australian dataset for misbehaviour detection in the internet of vehicles. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 624-629. IEEE, (2022)
11. Choi, J., Song, M.G., Lee, H., Sagong, C., Park, S., Lee, J., Yoo, J., Kim, H.K.: C-ITS Environment Modeling and Attack Modeling. arXiv preprint arXiv:2311.14327 (2023)
12. Seo, D.: A Studyon C-ITS Security Technology Analysis. Journal of the Korean Society for Software Evaluation 18, 103-113 (2022)
13. He, K., Kim, D.D., Asghar, M.R.: Adversarial machine learning for network intrusion detection systems: A comprehensive survey. IEEE Communications Surveys & Tutorials 25, 538-566 (2023)
14. Hnamte, V., Najar, A.A., Nhung-Nguyen, H., Hussain, J., Sugali, M.N.: DDoS attack detection and mitigation using deep neural network in SDN environment. Computers & Security 138, 103661 (2024)
15. Al Razib, M., Javeed, D., Khan, M.T., Alkanhel, R., Muthanna, M.S.A.: Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework. IEEE Access 10, 53015-53026 (2022)
16. Sun, X., Wang, H.: Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 686-693. (2022)

Reverse Engineering-Guided Fuzzing for CAN Bus Vulnerability Detection

Manu Jo Varghese[1], Frank Jiang[1], Abdur Rakib[2], Robin Doss[1], and Adnan Anwar[1]

[1] Deakin Cyber Research and Innovation Centre (Deakin Cyber), Deakin University, Geelong, Australia.
mvarghese@deakin.edu.au, frank.jiang@deakin.edu.au, robin.doss@deakin.edu.au, adnan.anwar@deakin.edu.au
[2] Centre for Future Transport and Cities
Coventry University, Coventry CV1 5FB, UK.
ad9812@coventry.ac.uk

Abstract. Vehicular systems are becoming interconnected digital ecosystems, reliant on complex networks for operations and safety. Cybersecurity vulnerabilities in these networks can jeopardize road safety. This study introduces a fuzzing framework to identify vulnerabilities in the Controller Area Network (CAN) bus. Using *Automated Reverse Engineering-Guided Fuzzing*, modified data packets are injected into the CAN framework, and Electronic Control Units (ECUs) reactions are monitored to uncover vulnerabilities. This approach identifies CAN bus network weaknesses and enhances understanding of their operational characteristics, setting a new standard for automotive cybersecurity. +

Keywords: CAN bus· Vulnerability detection · CAN test-bed · CAN security.

1 Introduction

Automotive advancements have transformed vehicles into digital ecosystems, reliant on interconnected Electronic Control Units (ECUs) for functions like engine dynamics and navigation [1]. These ECUs communicate via the Controller Area Network (CAN) bus, originally designed without prioritizing security [2]. Increased complexity and connectivity have raised concerns about CAN bus vulnerabilities [3]. This study introduces a fuzzing framework for the CAN bus. The Automated Reverse Engineering-Guided Fuzzing (ARE-GF) framework targets ECUs to identify vulnerabilities, using specialized security metrics to prioritize testing. The framework's efficacy is validated through simulations, uncovering potential vulnerabilities and enhancing security resilience. With vehicles increasingly becoming repositories of critical data concerning location, engine status, and beyond, securing these digital systems assumes paramount importance [4]. This research endeavours to fabricate a simulated CAN bus environment using low-cost, real-time apparatus. This simulation aims to generate traffic for sensitive data acquisition, laying a foundation for comprehensive security threat

analysis. The objective is to assess the system's robustness and efficiency, underscoring the escalating importance of security in CAN-facilitated systems. The expanding imperative for security within automotive software development presents a multifaceted challenge, necessitating a holistic approach [5]. This encompasses secure coding practices, rigorous testing and validation protocols, continuous monitoring, and an anticipatory posture towards emerging threats and technologies. As connected and autonomous vehicles become more ubiquitous, instituting stringent cybersecurity measures becomes increasingly vital to safeguard the safety, privacy, and trust of all road users. This research makes several key contributions to the field of automotive cybersecurity. It introduces the ARE-GF framework, a novel approach to identifying vulnerabilities within the CAN bus system. Additionally, it provides a methodology for prioritizing ECU modules based on susceptibility metrics, ensuring that the most critical areas are tested first. The research demonstrates the practical application of the framework within a simulated environment, highlighting its potential to enhance the security of real-world automotive systems. Finally, it underscores the importance of comprehensive security measures in the development of modern vehicles, contributing to ongoing efforts to protect critical automotive infrastructure from emerging cyber threats.

The rest of this paper is organized as follows. Section 2 provides the background on the CAN bus architecture and the importance of automotive cybersecurity. Section 3 reviews related work in the fields of fuzzing methodologies and automotive security. Section 4 presents the proposed fuzzing framework, detailing its components and methodologies. Section 5 evaluates the framework through empirical studies, demonstrating its effectiveness and comparing it with existing solutions. Finally, Section 6 concludes the paper and outlines future research directions.

2 Background

The Controller Area Network (CAN) protocol, developed by Bosch, facilitates seamless data exchange among vehicle components without direct addressing. Its decentralized model enhances system flexibility and scalability [35]. Central to the CAN bus system is a distinctive data frame structure, pivotal for communication among Electronic Control Units (ECUs). The CAN ID, consisting of 11 bits, serves as a unique message identifier within the CAN network, distinguishing it from socket-based protocols such as Ethernet by operating on a message-based broadcast system. The Data Length Code (DLC) is a 4-bit field that indicates the payload size within the message, which can range from 0 to 8 bytes. In reverse engineering endeavours, the DLC value is instrumental in ascertaining the data volume conveyed in each message. The data field itself, which can be up to 64 bits, constitutes the actual payload. Its size varies according to the DLC, and it transmits sensor data or control commands. Payload analysis is paramount in reverse engineering for elucidating the specifics of ECU communications.

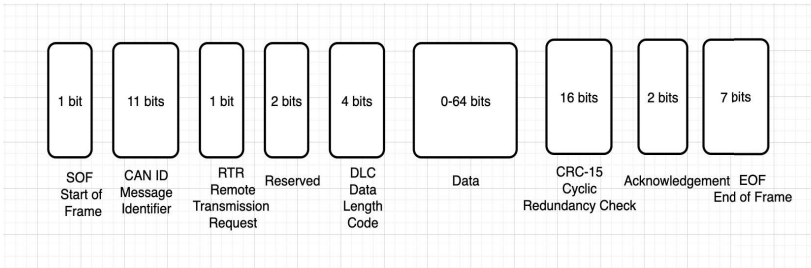


Fig. 1. CAN Data Frame Structure

Despite CAN's standardized protocol, manufacturers often use proprietary encoding for the data payloads, making the information difficult to interpret without additional context. This obscurity necessitates techniques like reverse engineering to decipher the proprietary encoding and understand the data's significance. Reverse engineering is vital in automotive cybersecurity [34], helping to identify vulnerabilities within ECUs and their communication protocols. It helps to make sense of the data transmitted over vehicular networks and identify potential security weaknesses [8]. Fuzzing is a critical technique in the Automotive domain, involving the injection of invalid, unexpected, or random data into systems to identify software bugs and vulnerabilities that cyber threats could exploit. This method is invaluable in the automotive sector, especially for vehicles equipped with Advanced Driver-Assistance Systems (ADAS), which consist of complex software ecosystems. By simulating a wide range of inputs, fuzzing helps to thoroughly test these ecosystems, uncover vulnerabilities, and ensure that communication systems meet stringent safety and security standards [9,10]. ADAS, which relies heavily on real-time data from sensors, radars, and cameras, presents significant challenges in terms of testing and validation. The dynamic nature of the data and the wide range of potential operating conditions make comprehensive testing difficult. This underscores the importance of robust testing methodologies to ensure the reliable functionality of ADAS systems throughout the vehicle's operational lifespan. The integration of ADAS also raises cybersecurity concerns, particularly when development is outsourced. This necessitates vigilant oversight of third-party components to maintain the security and integrity of the vehicle's software ecosystem [13,14,15].

3 Related Work

Extensive investigations in fuzzing methodologies and automotive cybersecurity have highlighted gaps necessitating further exploration. This section reviews the evolution of fuzzing approaches and their applicability to the automotive sector.
Automotive Cybersecurity: Research in automotive cybersecurity has intensified in response to the increasing complexity of vehicular systems. Security analyses have focused on in-vehicle infotainment systems due to their external connectivity and interactivity. Li et al. [1] devised the SP-E framework for a nuanced security assessment of infotainment systems, emphasizing the need for

specialized evaluative methodologies. Comprehensive security strategies for in-vehicle networks have also been explored. Luo et al. [2] advocated for holistic security strategies, while Wang et al. [3] examined the security mechanisms relevant to OTA updates, stressing the importance of secure updates in maintaining system integrity. Li et al. [4] introduced a security evaluation framework for connected vehicles based on attack chains, and Shirvani et al. [5] proposed a framework for assessing security risks in electric vehicles.

Fuzzing Techniques: Foundational expositions on fuzzing, including its operational principles and broad applications, are presented in [17]. An exhaustive survey of fuzzing techniques and a taxonomy of prevalent fuzzers are detailed in [18], providing a valuable resource for selecting appropriate methods in new projects. Specialized tools for network protocol fuzzing have been developed. The SNOOZE Fuzzer [32] automates fuzzing based on protocol states and messages, while AutoFuzz [33] constructs FSMs from recorded network traffic to guide fuzzing.

Automotive Security Research: Pioneering work by Miller and Valasek [19] unveiled significant vulnerabilities within a 2014 Jeep Cherokee, demonstrating the feasibility of remote system control. Subsequent research scrutinized the AUTOSAR standard [20] and applied threat modeling techniques to the CAN protocol, uncovering vulnerabilities such as ECU spoofing via CAN packets. Challenges unique to fuzzing automotive control networks, as contrasted with conventional software fuzzing, are documented by Smith in the Car Hacker's Handbook [11] and further studies [22]. These works highlighted the unique responses of vehicles and ECU vulnerabilities when fuzzing automotive networks, either with or without CAN bus feedback. Proposals to use DBC files for generating automatic fuzzing test cases aim to enhance fuzzing precision and efficacy.

Need for a Novel Automobile-Specific Fuzzing Framework: Current automotive fuzzing frameworks struggle with the intricate interplay between ECUs in modern vehicles. The proposed fuzzing framework aims to address this by combining automated reverse engineering of CAN signals, real-time sensor feedback, and targeted fuzzing guided by a vulnerability prediction model. This specialized approach promises to improve the efficiency and thoroughness of uncovering security weaknesses in today's complex vehicular systems.

4 Proposed Fuzzing Framework

The Automated Reverse Engineering-Guided Fuzzing (ARE-GF) framework incorporates Automated Reverse Engineering to target ECUs. This methodology deploys tailored fuzzing strategies to enhance the efficiency and efficacy of vulnerability detection within vehicular CAN systems.

4.1 Segmentation Process

In the complex ecosystem of vehicle CAN networks, deciphering the signal encoding within data frames presents a formidable challenge, compounded by the

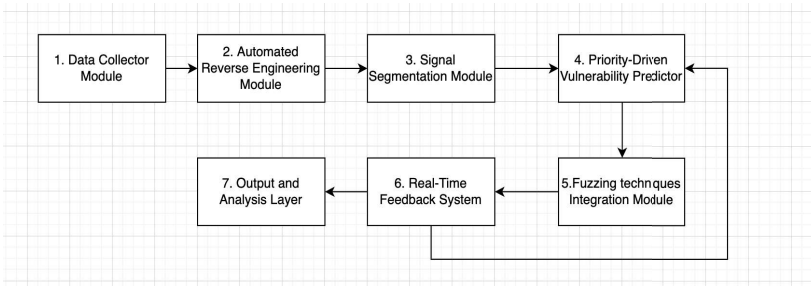


Fig. 2. Proposed Fuzzing Framework

initial absence of encoding knowledge. To surmount this obstacle, our research introduces a novel methodological approach for signal grouping and interpretation. Employing a 'greedy' algorithm, this technique endeavours to reconstruct signals from the constituent bits of a data frame, subsequently subjecting these signals to a rigorous 'signal validation' process. The segmentation process starts

Algorithm 1 Signal Segmentation

```
procedure SignalSegmentation(candump)
    for each frame in candump do
        startBit ← 0
        while startBit < number of bits in frame do
            for length ← 1 to (number of bits in frame − startBit) do
                signal ← extractSignal(frame, startBit, length)
                if V(signal) ≠ V(extractSignal(frame, startBit, length + 1)) then
                    commitSignal(signal)
                    startBit ← startBit + length
                    break
                end if
            end for
        end while
    end for
end procedure
```

with the algorithm at the first bit (bit 0) of the frame. It gradually combines more bits to form a possible complete signal. After the initial signal is assembled, each possible signal is carefully checked to see if it is valid. The decision-making in this process involves two key steps: choosing whether to add another bit to the possible signal for re-evaluation or to consider the signal complete and move on to the next group of bits. This iterative process continues until every bit within the frame is evaluated and categorized, ensuring a thorough examination and interpretation of signals.

Once the Frame series is obtained group based on the origin of ECU we could analyse the boundaries in this data and find common types of data. Figure 3

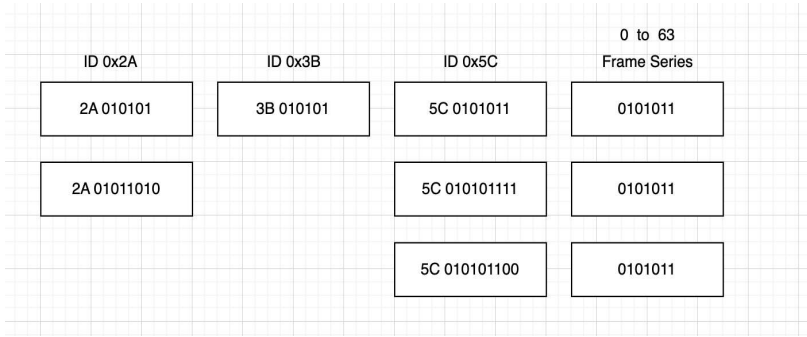


Fig. 3. CAN Data Segmentation

showcases the segmentation of CAN data, demonstrating how data frames are dissected to extract meaningful signals.

Algorithm 2 Determine Bit Significance Order

```
procedure BitSignificanceOrder(observedPayloads)
    bitChangeFrequency ← [0] * 8
    previousPayload ← observedPayloads[0]
    for i ← 1 to length(observedPayloads) do
        changes ← observedPayloads[i] ⊕ previousPayload
        for bit ← 0 to 7 do
            if changes & (1 ≪ bit) then
                bitChangeFrequency[bit] ← bitChangeFrequency[bit] + 1
            end if
        end for
        previousPayload ← observedPayloads[i]
    end for
    sortedIndices ← sort indices of bitChangeFrequency by value
    return sortedIndices
end procedure
```

4.2 Efficiency of Algorithms

Signal Segmentation Algorithm: The `SignalSegmentation` algorithm has a time complexity of $O(n \times m^2)$, where n is the number of frames and m is the number of bits in a frame. The outer loop iterates over each frame ($O(n)$), and within each frame, the algorithm evaluates each bit and possible signal length ($O(m) \times O(m)$). This quadratic complexity in terms of bits per frame ensures a thorough segmentation process.

Bit Significance Order Algorithm: The `BitSignificanceOrder` algorithm operates with a linear time complexity of $O(n)$, where n is the number of observed payloads. It efficiently determines the significance of each bit by tracking changes across payloads. This involves an $O(1)$ operation for each bit in the payload, resulting in a highly efficient process for large datasets.

4.3 Enhancing CAN Bus Security through Priority-Driven Vulnerability Prediction

The Priority-Driven Vulnerability Predictor (PVP) module constitutes a pivotal enhancement within the Automated Reverse Engineering-Guided Fuzzing (ARE-GF) framework. This module utilizes specialized security metrics to prioritize fuzz test cases, focusing on areas with the highest potential for unearthing significant vulnerabilities within the Controller Area Network (CAN) bus. By systematically identifying and prioritizing fuzzing efforts, the PVP module enhances the efficiency and effectiveness of vulnerability detection processes within the framework. The specialized security metrics used by the PVP module are summarized in Table 1. These metrics guide the PVP module in systematically identifying and prioritizing fuzz test cases, thereby significantly enhancing the efficiency and effectiveness of vulnerability detection processes within the framework.

Table 1. Specialized Security Metrics for Priority-Driven Vulnerability Prediction

Metric	**Description**
Vulnerability Score	Scores based on known vulnerabilities and history.
Exposure Level	Degree of exposure to external inputs/communications.
Critical Functionality	Priority for ECUs controlling critical functions.
Communication Frequency	Frequency of communication on the CAN bus.
Interconnectedness	Impact on other vehicle systems.
Anomaly Detection	Ability to detect anomalies in behavior.

Refinement of Fuzzing Techniques within the CAN Bus Arbitration Process: Our methodology initiates with a foundational approach to fuzzing, leveraging the random selection of arbitration IDs and message payloads to scrutinize the system. To elevate the efficacy of this process and augment the identification of vulnerabilities within the Electronic Control Units (ECUs), we incorporate a triad of sophisticated fuzzing strategies: brute-force fuzzing, mutation fuzzing, and state change identification fuzzing. Brute-force fuzzing employs a comprehensive examination of all possible permutations of certain message components, specifically targeting those known to influence ECU responses. This approach proves invaluable when targeting ECUs with a known sensitivity to a specific range of IDs. Mutation fuzzing introduces incremental modifications to the messages, manipulating bits or altering message configurations to gauge the impact of minor variations on ECU behavior. State change identification fuzzing relies on the discernment that only specific payload bits significantly affect ECU state changes. By adjusting these pivotal bits, this strategy seeks to unveil conditions leading to state alterations, often necessitating a sequence of messages to effectuate a change.

4.4 Automated ECU Analysis via Data Acquisition System (DAS)

Integrating these enhanced fuzzing strategies, the framework transitions into an automated exploration mode, guided by immediate feedback from a dedicated

sensor harness. This harness, affixed directly to the target ECU, enables the precise assessment of the fuzzing impact, facilitating dynamic adjustments to our strategies [23]. Each node within the CAN network initiates by establishing a unique identifier (ID), which is essential for network coordination. Upon detecting the Start of Frame (SOF) signal, nodes prepare for message transmission, initiating with the highest-order bit of their ID. During this phase, nodes concurrently transmit and monitor bits, engaging in an arbitration process to resolve priority conflicts. Dominance is established through the transmission of a 'dominant' bit (0), with the node asserting priority unless a 'recessive' bit (1) is observed, signalling the halting of transmission in deference to a higher-priority message. Following arbitration, the prevailing node distributes its data, with receiving nodes acknowledging pertinent messages. Error detection mechanisms, including CRC checks and acknowledgement slots, ensure message integrity, with error frames signalling detected faults. Post-transmission or error resolution, the network synchronizes in preparation for subsequent communications, reinstating an idle state.

Automated ECU Analysis via Data Acquisition System (DAS): Merging these enhanced fuzzing strategies culminates in an automated exploration mode, orchestrated by real-time feedback from a sensor harness. This direct attachment to the target ECU facilitates a granular assessment of the fuzzing impacts, enabling dynamic strategy adjustments.

CAN Bus Testbed Components: The construction of a CAN Bus testbed involves the assembly of several key components, each fulfilling a pivotal role within the system: **STM Nucleo − F103RB (MCU)** acts as the central controller within the CAN bus system, selected for its inherent support for CAN communication and cost-effectiveness. Integration with the CAN network is achieved via dedicated CAN interface pins, with programming facilitated through the Mbed framework in C language to manage CAN communication protocols. **MCP 2551 (Transceiver)** enables the interface between the CAN controller (MCU) and the physical bus, crucial for the transmission and reception of CAN signals. **Dupont Lines** serve to establish physical connections between hardware pins and the breadboard, enabling the linkage of transceiver pins to the corresponding MCU pins and CAN bus lines. **CAN Bus Logger (CL2000)** is responsible for logging and storing traffic data from the CAN bus, with connectivity to a computer via USB for data offloading and analysis. **PicoScope 2204A** functions as both an oscilloscope and protocol decoder for the CAN bus, capturing and decoding communication signals with software tools aiding in signal analysis and decoding.

Design and Connection Considerations: Adherence to ISO 11898 [37] standards for CAN network cabling and termination is imperative. Additional design considerations include the implementation of LEDs on the development board for visual indication of transmission activities, and the establishment of a power supply by connecting the 5V and Ground pins from the STM Nucleo board to appropriate power sources. Testing and validation of the CAN bus system are

facilitated through the utilization of tools like the CL2000 and PicoScope 2204A, ensuring the system's functional integrity.

5 Evaluation and Findings

The evaluation of the fuzzing framework highlights its efficacy and resource efficiency in automotive security testing. Conducted on a robust hardware setup (Apple M1 chip, 16GB memory, 512 GB SSD, macOS), this section compares the framework's performance against established metrics and alternative frameworks.

5.1 Performance Metrics of the Cybersecurity Testing Framework

Table 2 presents the performance of the novel fuzzing framework, illustrating its competencies across various operational metrics. These metrics elucidate the framework's responsiveness, efficiency, and resource utilization, evidencing its superiority in handling cybersecurity testing within automotive systems.

Table 2. Performance Results of the Proposed Cybersecurity Testing Framework Compared to Industry Benchmarks [36].

Metric	Definition	**Results**	Benchmark
Response Time	Number of services per unit time	**<500 ms**	600-800 ms
Query to View Time	Total time by the Query service	**<50ms**	< 50 ms
Query per Second	Number of queries per unit time	**>600**	300-600
Failure Ratio	Failed service/total service	**<0.1%**	< 0.1%
CPU Usage	CPU utilization (mean/peak)	**12%/80%**	15-20% / 75-85%
RAM Usage	Utilization rate of RAM resources	**1 GB/8 GB**	2 GB / 8 GB
Disk Throughput	Amount of data r/w from a disk	**400 MB/s**	350-450 MB/s
Network Throughput	The amount of network data	**9 MB/s**	8-10 MB/s

The detailed analysis highlights the framework's capacity to process a significant volume of queries and transactions swiftly, maintaining operational integrity with minimal resource consumption.

5.2 Comparative Analysis

A comparative examination reveals the distinct advantages of the proposed fuzzing framework over its counterparts, as depicted in Table 3. This analysis accentuates the comprehensive process incorporation, extensive test case coverage, and the adaptability of the framework to various testing scenarios [25,28].

5.3 Addressing Identified Vulnerabilities

The prowess of the proposed framework in identifying and mitigating vulnerabilities is further substantiated through a comparative pass rate analysis, as

Table 3. Comparison between the Novel Fuzzing Framework and other frameworks.

Framework	Process	Test Case	Framework	Scenarios
Q. Li [1].	×	3	×	Risk Assessment
Shirvani [2].	×	10	×	Risk Assessment
Schönhärl [3].	×	3	✓	Education
S. Li [4].	✓	4	×	Penetration Testing
F. Luo [5].	✓	1	✓	Penetration Testing
K. He [6].	×	2	×	Penetration Testing
Proposed Framework	✓	10	✓	Penetration Testing

summarized in Table 4. This evaluation underscores the framework's efficacy in addressing critical vulnerabilities, outperforming existing solutions [29].

Table 4. Comparison of frameworks in addressing identified vulnerabilities and their pass rate.

Framework	Extended CAN IDs	DoS Vulnerability	Pass Rate
Q. Li [1]	×	×	70%
Shirvani [2]	Unknown	×	80%
Schönhärl[3]	×	Unknown	75%
S. Li [4]	Unknown	✓	67%
F. Luo [5]	✓	✓	72%
K. He [6]	×	×	76%
Proposed Framework	✓	✓	**94**%

The assessments elucidate the proposed framework's capabilities in enhancing automotive cybersecurity, particularly through its nuanced approach to fuzzing and vulnerability analysis. The empirical data gleaned from these evaluations advocates for its adoption and further exploration in mitigating automotive system vulnerabilities.

6 Conclusion and Future Work

This study introduces a fuzzing framework tailored for the CAN bus, targeting ECUs to identify vulnerabilities. Using specialized security metrics, the framework prioritizes testing of susceptible ECU modules. Simulations in a CAN bus environment demonstrate its effectiveness in uncovering vulnerabilities and enhancing security resilience. This research highlights the importance of securing vehicular digital ecosystems and provides a practical solution to mitigate automotive cybersecurity risks. In future work, we would like to refine the reverse engineering process through the application of machine learning techniques. We aim to extend the scope by incorporating various vehicle models and ECU ranges, thereby enhancing the performance of the model and ensuring its adaptability to real-world scenarios.

References

1. S. Li, X. Zhang, Y. Zhou, and M. Zhang, "SP-E: Security Evaluation Framework of In-vehicle Infotainment System based on Threat Analyses and Penetration Tests," J. Phys. Conf. Ser., vol. 2517, p. 012012, 2023.
2. F. Luo, X. Zhang, and S. Hou, "Research on Cybersecurity Testing for In-vehicle Network," In Proceedings of the 2021 ICITES, Chengdu, China, September 2022.
3. K. He, C. Wang, Y. Han, and X. Fang, "Research on cyber security Technology and Test Method of OTA for Intelligent Connected Vehicle," In Proceedings of the 2020 ICBAIE, Virtual Conference, China, July 2022.
4. Q. Li, J. Zuo, R. Cao, J. Chen, Q. Liu, and J. Wang, "A Security Evaluation Framework for Intelligent Connected Vehicles Based on Attack Chains," IEEE Network, p. 1, 2023.
5. S. Shirvani, Y. Baseri, and A. Ghorbani, "Evaluation Framework for Electric Vehicle Security Risk Assessment," IEEE Trans. Intell. Transp. Syst., pp. 1–24, 2023.
6. B. Arkin, S. Stender, and G. McGraw, "Software penetration testing," IEEE Secur. Priv., vol. 3, pp. 84–87, 2005.
7. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, ... & T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in Proceedings of the 20th USENIX conference on Security (SEC'11), USENIX Association, 2011.
8. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, & S. Savage, "Experimental security analysis of a modern automobile," in 2010 IEEE Symposium on Security and Privacy, IEEE, pp. 447-462, 2010.
9. C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Black Hat USA, 2015.
10. D. K. Nilsson, U. E. Larson, and F. Picasso, "Simulating attacks on CAN buses: A look at the AUTOSAR and FlexRay protocols," in 2008 5th Workshop on Embedded Systems Security (WESS '08), ACM, 2008.
11. A. Pawlowski, B. Stelte, and C. Müller, "CAN-Fuzz - A Practical Framework for Fuzzing CAN Applications," in NordSec 2017: Secure IT Systems, Springer, Cham, pp. 45-61, 2017.
12. W. Lawrenz, "CAN System Engineering: From Theory to Practical Applications," Springer, 2016.
13. A. Takanen, J. DeMott, and C. Miller, "Fuzzing for Software Security Testing and Quality Assurance," Artech House, 2018.
14. M. Wolf and M. Scheibel, "Automotive Cybersecurity: Securing the Modern Vehicle," Wiley, 2020.
15. B. Groza and P. S. Murvay, "Security for Automotive Electrical and Electronic Systems," in Automotive Cybersecurity, Springer, Cham, pp. 1-33, 2020.
16. G. Loukas, "Cyber-Physical Attacks and Defenses in the Smart Grid and Connected Vehicles," CRC Press, 2019.
17. V. J. M. Manès, H. Han, C. Han, S. K. Cha, M. Egele, S. Schwab, and M. Woo, "The art, science, and engineering of fuzzing: A survey," IEEE Transactions on Software Engineering, vol. 46, no. 11, pp. 1216-1242, 2019.
18. G. Banks, M. Cova, V. Felmetsger, K. Almeroth, R. Kemmerer, and G. Vigna, "SNOOZE: Toward a Stateful Network Protocol Fuzzer," in Information Security, Springer, Berlin, Heidelberg, pp. 343-358, 2006.
19. C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," Black Hat USA, vol. 2014, p. 94, 2014.

20. O. Avatefipour and W. H. Hassan, "Comprehensive overview of cyber-physical system security in connected and autonomous vehicles," IET Cyber-Physical Systems: Theory & Applications, vol. 5, no. 3, pp. 211-221, 2020.
21. K. T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection," in 25th USENIX Security Symposium (USENIX Security 16), pp. 911-927, 2016.
22. Q. Zeng, J. Liu, Q. Chen, J. Zhao, and S. Liu, "Fuzz testing based on attack injection for in-vehicle network security evaluation," IEEE Access, vol. 7, pp. 35973-35983, 2019.
23. T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks – Practical examples and selected short-term countermeasures," Reliability Engineering & System Safety, vol. 96, no. 1, pp. 11-25, 2011.
24. S. N. Narayanan, S. Mittal, A. Joshi, and K. Joshi, "Machine Learning Techniques for Intrusion Detection Systems: A Review," Journal of Network and Computer Applications, vol. 167, p. 102739, 2020.
25. Y. Chen, C. Su, M. F. Yeh, and C. Y. Huang, "ECU Fuzzing: Discovering Vulnerabilities in Automotive Systems via Fuzz Testing," in Proceedings of the ACM Symposium on Applied Computing, 2020.
26. J. Van Bulck, M. Münch, N. Weichbrodt, R. Kapitza, F. Piessens, and R. Strackx, "SGX-based Secure CAN Communication," in 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), IEEE, pp. 357-366, 2018.
27. B. Groza and P. S. Murvay, "Evaluating the Security of Connected Vehicle Architectures Through Penetration Testing and Fuzzing," in Automotive Cybersecurity, Springer, Cham, pp. 3-29, 2021.
28. T. Wang, T. Wei, G. Gu, and W. Zou, "TaintScope: A checksum-aware directed fuzzing tool for automatic software vulnerability detection," in 2010 IEEE Symposium on Security and Privacy, IEEE, pp. 497-512, 2010.
29. M. Boehme and M. d'Amorim, "Fuzzing Frameworks: A Survey and Taxonomy," Technical Report, TR-2013-004, Department of Computer Science, University of Pernambuco, 2013.
30. Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," IEEE Transactions on Software Engineering, vol. 33, no. 4, pp. 225-237, 2007.
31. M. Zhang, Y. Duan, Q. Liu, and H. Yin, "V-Fuzz: Vulnerability-oriented Evolutionary Fuzzing," arXiv preprint arXiv:1901.01142, 2019.
32. G. Banks, M. Cova, V. Felmetsger, K. Almeroth, R. Kemmerer, and G. Vigna, "Snooze: Toward a stateful network protocol fuzzer," in Lecture Notes in Computer Science, pp. 343–358, 2006.
33. S. Gorbunov and A. Rosenbloom, "AutoFuzz: Automated Network Protocol Fuzzing Framework," International Journal of Computer Science and Network Security, vol. 10, pp. 239–245, August 2010.
34. T. Huybrechts, Y. Vanommeslaeghe, D. Blontrock, G. Van Barel, and P. Hellinckx, "Automatic Reverse Engineering of CAN Bus Data Using Machine Learning Techniques," Advances on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 751–761, 2017.
35. Bosch GmbH, "CAN Specification Version 2.0," Robert Bosch GmbH, 1991.
36. Sensors 2022, 22, 9211. "A Comprehensive Survey on Fuzz Testing for Automotive Cybersecurity."
37. International Organization for Standardization, "ISO/SAE 21434:2021 Road vehicles — Cybersecurity engineering," [Online]. Available: `https://www.iso.org/standard/86384.html`

Air-Fuzz: Feasibility analysis of fuzzing-based side-channel information leakage attack in air-gapped networks

Yeon-Jin Kim[1][0009−0006−5538−7187], Na-Eun Park[1][0009−0008−0939−5762], and Il-Gu Lee[1][0000−0002−5777−4029]

Sungshin Women's University, 2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul 02844, Republic of Korea
220246046@sungshin.ac.kr
nepark.cse@gmail.com
iglee@sungshin.ac.kr

Abstract. As the number of cyberattacks via networks increases, air-gapped networks that are physically isolated from the Internet are recommended to protect systems and industrial facilities. However, previous research has highlighted the possibility of data exfiltration attacks in air-gapped environments by exploiting computer sys-tems or peripheral devices. Consequently, research to identify potential attack types and vectors in air-gapped environments is required. This study proposes an attack model that uses fuzzing on Internet of Things connectivity devices to exfil-trate data within an air-gapped environment through side-channel signal patterns. We conducted an experiment to verify the feasibility of the attack model in a noisy channel and compared its performance with that of conventional attack models affected by interference and noise. By injecting Gaussian noise into wire-less signals encoded with data, we compared the Bit Error Rate (BER) of the proposed attack model with that of conventional models. The BER of the pro-posed attack model was improved by approximately 59.43% compared with that of conventional attack models.

Keywords: Air-Gap Network · Side Channel · Internet of Things.

1 Introduction

With the widespread high-speed information and communication technologies, na-tional infrastructure, industrial facilities, and individual homes and objects have be-come connected to the Internet [1]. However, as important information assets have become connected to the Internet, the frequency of cyberattacks has increased rap-idly. Large-scale Distributed Denial of Service (DDoS) attacks that reduce system availability and ransomware attacks that take important documents and systems hostage and demand ransoms are expanding beyond industrial companies to target national infrastructure [2, 3]. Efficient defense technologies have been investigated to protect systems from various cyberattacks [4]. In

addition, as the number of net-work-based cyberattacks has increased recently, the air-gap concept has been proposed to protect systems from networks.

An air-gapped network is a closed environment in which business and Internet networks are physically separated [5]. To prevent cyberattacks through networks, air-gapped networks protect industrial systems from external threats and ensure their safe operation. However, recent research has demonstrated the possibility of data exfiltration in air-gapped environments by encoding binary data into optical, acous-tic, and electromagnetic signals generated while operating computer systems or pe-ripheral devices and transmitting them to an external network [6]. Stuxnet, which was used to attack Iran's Natanz nuclear facility in 2010, used intermediary devices such as USB flash drives to gain access within an air-gapped environment. Additionally, four zero-day vulnerabilities were used to access the operating system and disrupt the normal operation of industrial control processes by changing the frequency of the current power of the centrifuge [7]. The Stuxnet incident confirmed that internal infil-tration and data leakage were possible even in an air-gapped environment. Therefore, research to identify attack vectors where attacks can occur, and to detect and re-spond to them to protect systems and data within air-gapped networks is needed.

This study shows that a data leak attack is feasible by utilizing the side channel of Internet of Things (IoT) connectivity devices in an air-gapped network, and by ana-lyzing the attack vector.
The main contributions of this paper are as follows.

- We proposed Air-Fuzz, an attack model that generates wireless signals by finding signal patterns in the side channel through IoT firmware fuzzing.
- The concept of the proposed attack model was verified experimentally, and the feasibility of an air-gapped attack using wireless signals was demonstrated.

The remainder of this paper is organized as follows. Section 2 analyzes the previ-ous research on data leakage attacks in air-gapped networks. Section 3 explains the proposed data leakage attack model, and Section 4 analyzes the experimental results to verify the effectiveness of the attack model. Open issues are discussed in Section 5, and the conclusions are presented in Section 6.

2 Related Work

Guri et al. [8] proposed a data leakage attack scenario in an air-gapped environment by generating a signal in the 2.4 GHz Wi-Fi frequency band by repeating the reading and writing memory on the DDR SDRAM memory bus. The experiment demonstrat-ed that the attack was effective up to a distance of 1.8 m. This paper demonstrated through experiments that data can be leaked using a DDR SDRAM memory bus in an air-gap environment. However, physical proximity is required between the attack-target computer and the receiving device. One limitation is that the reliability of the transmitted data is low because the surrounding environment greatly influences it.

Lee et al. [9] analyzed attack surfaces that could be used within a closed network, designed an attack system that leaked internal data using optical media, and demon-strated its effectiveness. The data leaked from devices inside the air-gapped network were collected and encoded into binary data. The encoded data are externally trans-mitted through controllable LEDs. They conducted experiments on various environ-mental variables that could increase the success rate and efficiency of data leakage attacks in the proposed attack scenario. The study derived commands for manipulat-ing LEDs and proposed an air-gapped attack scenario. However, the data leakage method using LED has the disadvantage that the receiver must be located in a straight line of sight, and abnormal LED operation patterns can be easily observed.

Li et al. [10] proposed SpiralSpy, which leaks data by manipulating the speed of a cooling fan in a conventional air-gapped environment attack scenario. The experi-ments proved that data could be transmitted at a speed of 6 bit/s, even at a distance of 8 m. However, the fan must be controlled precisely, considering the its response delay and the time required to adjust the speed. Additionally, because the fan may be activated owing to intervention by the operating system while data is being trans-mitted through SpiralSpy, there is a limitation in that it is necessary to distinguish who is operating the cooling fan.

Previous research has proposed an attack scenario that leaks data using internal PC systems or peripheral devices in an air-gapped network environment. Although it suggests an attack vector that has not been previously considered, it has common limitations such as limitations on devices that can be used for attacks or many envi-ronmental variables that must be considered. Therefore, we propose an Attack mod-el, Air-Fuzz, which transmits data by exploring usable frequency bands through fuzz-ing to address the limitations of conventional methods, such as physical accessibility and the effects of channel interference and noise.

3 Attack Model: Air-Fuzz

The structural diagram of Air-Fuzz is shown in Fig. 1.

Air-Fuzz first assumes that IoT devices within an air-gapped network are infected with malware through social engineering techniques or supply chain attacks. PCs on the same network as the IoT devices are also infected with malware, and attackers who break into air-gapped networks collect data to be leaked externally, such as accounts and system passwords [11]. The collected data are converted into binary data and encoded into a wireless signal through On-Off Keying (OOK) encoding, and the transmitter transmits the signal to an external receiver. The transmitter used in this process is a firmware that controls the IoT connectivity devices. By fuzzing the firm-ware, a wireless signal is generated by determining the pattern of the signal exiting the side channel. The receiver outside the air-gapped network receives the wireless signal. The received signal was analyzed to extract the OOK-encoded signal, process it, and decode the binary data.

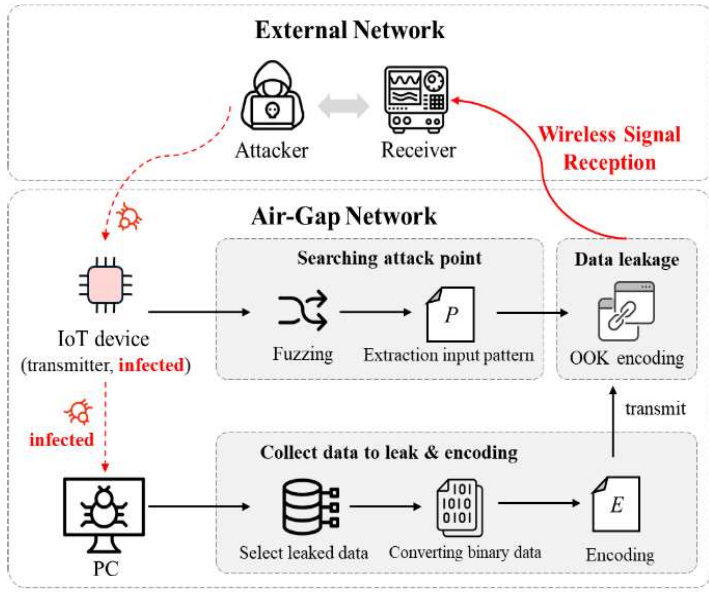


Fig. 1. Structural diagram of the Air-Fuzz.

Air-Fuzz proceeds in two ways: a transmitter within the air-gapped network and a receiver outside the air-gapped network. The transmitter performs fuzzing on the internal program to collect signal patterns that can be triggered. When the OOK en-coded data is "1," a wireless signal is transmitted through the identified signal pattern; when it is "0," it sleeps. The receiver calculates a threshold that can distinguish be-tween "1" and "0" in the collected wireless signals, divides the signal by the sample rate, and compares it to the threshold to distinguish between "1" and "0."

3.1 Transmitter

The pseudocode that receives binary data from an attacker and generates a wireless signal to leak data is the same as that in Algorithm 1.

To run the program, the frequency band for transmitting the wireless signal (frequency), binary data pattern (bit_pattern), time to continue the frequency transmission (time_on), and offset time between each bit (time_off) of the data must be in-put. Environmental variables, excluding bit patterns, must be agreed upon by the wireless signal sender and receiver before attempting data leakage. The data trans-mission process continues until the receiver requests the end of the data transmission. Each bit entered as a bit pattern is read individually, and different actions are taken to identify "1" and "0."

Algorithm 1 Data Generation Transmitter Algorithm

```
1: procedure FrequencyControl(frequency, bit_pattern, time_on, time_off)
2:     pid ← 0
3:     trap "Script terminated by user" SIGINT, SIGTERM
4:
5:     while true do
6:         for each bit in bit_pattern do
7:             if bit ≡ 1 then
8:                 StartTune(frequency, time_on, time_off)
9:             else
10:                Sleep(time_on)
11:            end if
12:            Sleep(time_off)
13:        end for
14:        Sleep(10)                          ▷ Sleep for 10 seconds before repeating
15:    end while
16: end procedure
17:
18: procedure StartTune(frequency, time_on, time_off.)
19:    pid ← start tune process with frequency
20:    Sleep(time_on)
21:    tune_pid ← pid + 2
22:    kill tune_pid
23: end procedure
```

When the bit to be transmitted is "1," the StartTune procedure is executed. The StartTune procedure uses the variables frequency, time_on, and time_off as inputs, and transmits a wireless signal at a specified frequency for the duration of the time_on variable. After the signal has been transmitted for the set duration, the kill command terminates the transmission. If the bit is "0," the system indicates this by sleeping for the duration of time_on without transmitting any signal. After a bit has been transmitted, the system sleeps for the duration of timeoff to provide an offset interval between two consecutive bits.

This process is repeated for the length of the bit pattern, and when all transmis-sions are completed, it sleeps for 10 s to notify the receiver that the bit transmission has been completed. After Sleep, the transmission resumes from the first bit for the receiver to receive the bit again from the beginning.

3.2 Receiver

The receiver receives and analyzes the wireless signals transmitted by the transmitter from outside the air-gapped network. The pseudocode for decoding the binary data from a signal received through a wireless signal receiver is the same as that in Algorithm 2.

Algorithm 2 Data Receiver Algorithm

```
function LoadHackRFSamples(filename)
    raw_data ← Load file data as 8-bit integers
    iq_data ← raw_data[0::2] + 1j × raw_data[1::2]
    return iq_data
end function
function AverageSignalStrength(samples, sample_rate)
    samples_per_second ← int(sample_rate)
    num_seconds ← len(samples) // samples_per_second
    average_signal_strength ← EmptyList
    for i ← 0 to num_seconds do
        start_idx ← i× samples_per_second
        end_idx ← start_idx + samples_per_second
        one_second_samples ← samples[start_idx:end_idx]
        abs_samples ← Abs(one_second_samples)
        avg_strength ← Mean(abs_samples)
        Append(average_signal_strength, avg_strength)
    end for
    return average_signal_strength
end function
function DecodeSignal(samples)
    binary_data ← EmptyList
    abs_samples ← Abs(samples)
    threshold ← Mean(abs_samples)
    for sample in samples do
        if Abs(sample) > threshold then
            Append(binary_data, 1)
        else
            Append(binary_data, 0)
        end if
    end for
    return binary_data
end function
output_file ← UserInput('Enter the output file name:')
center_freq ← UserInput('Enter the center frequency (in MHz):') MHz
sample_rate ← UserInput('Enter the sample rate (in MSPS):') MSPS
samples ← LoadHackRFSamples(output_file)
average_signal_strength ← AverageSignalStrength(samples, sample_rate)
binary_data ← DecodeSignal(average_signal_strength)
Print(binary_data)
```

The LoadHackRFSamples function reads sample data from a binary file. Data from a binary file are read as an array of 8-bit integers, and real (I) and imaginary (Q) numbers are added to return a complex number array.

The AverageSignalStrength function calculates the number of samples per second. Divide the Samples by the sample rate of the receiver and store the time in seconds of Sampled data in the binary file in the num_seconds variable. The for statement is repeated for the number of seconds stored in the num_seconds variable, and the average signal intensity per second is calculated, stored in an array, and returned.

The DecodeSignal function receives one signal array per second and extracts the binary bits. To distinguish between 0 and 1, the average of the absolute values of all the samples is calculated and used as a threshold. It traverses the Samples array in which the average signal intensity per second received as a parameter is calculated, and distinguishes between 1 and 0 by checking whether the average intensity exceeds the threshold.

To execute the pseudocode of Algorithm 2, the receiver must enter the file of the bin extension where the received radio signal is stored, center frequency at which the signal is collected, and sample rate data of the receiver. Subsequently, the average signal strength per second is calculated by converting the bin file contents into com-plex number data using the LoadHackRFSamples function. The binary data trans-mitted by the sender is extracted using the array calculated through the AverageSig-nal Strength function as the input value of the DecodeSignal function.

4 Evaluation

An experiment was conducted to verify the concept of the proposed attack model. One hundred inputs were randomly generated through fuzzing. Fig 2 shows a graph that runs the target program with the generated input and collects the signal pattern generated during this process.

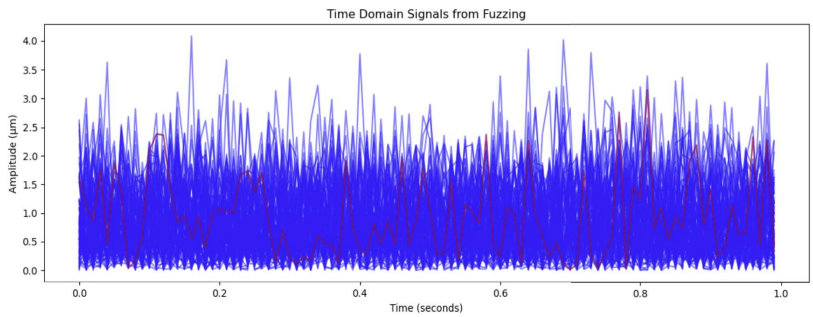


Fig. 2. Graph of fuzzing response signal.

The X-axis represents time and the Y-axis represents the amplitude of the signal. Responses with large amplitudes were obtained from the collected signals based on the center frequency, and binary data were encoded. Fig. 2 The signal in red in the graph is the signal with the largest amplitude and is used for OOK encoding. If the binary data is "1," the input that generated the largest amplitude is re-executed, and if it is "0," sleep is performed for the receiver to distinguish the data. The OOK-encoded signal is illustrated graphically in Fig. 3.

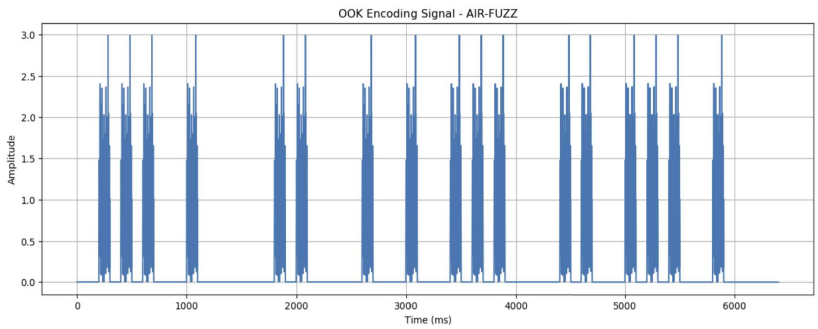


Fig. 3. Graph of the OOK encoding signal.

The extent to which Air-Fuzz is affected by noise when transmitting a wireless sig-nal using encoded data is determined. The results were compared with those of the conventional attack model, and the results of calculating the Bit Error Rate (BER) according to the noise are summarized in Table 1.

Table 1. Comparison of Bit Error Rates (BERs) for Air-Fuzz and conventional attack models under differ-ent noise level and iterations.

Noise Insertion Method	Air-Fuzz BER(%)	Conventional Attack Model BER(%)
Random Noise Insertion (100 iterations)	2.84	7
Incremental Noise Insertion (0.1 to 1.3, 100 iterations)	10.59	19.28
Incremental Noise Insertion (0.1 to 1.3, 300 iterations)	9.97	18.72

Because the conventional attack model uses the same bands as Wi-Fi and Blue-tooth signals, it is affected by channel noise when transmitting signals to receivers outside the air-gapped. In the experiment, when the noise signal was randomly insert-ed by fixing the maximum noise level at 0.5, a low BER of

approximately 7% was obtained. However, when the total average was calculated by increasing the maxi-mum noise level to more than 1.0, the BER increased by 19%. In contrast, Air-Fuzz showed a BER of 10% or lower regardless of the noise level. Based on random noise, Air-Fuzz improved BER by approximately 59.43% compared with the conventional attack model. It is analyzed that Air-Fuzz is more resistant to noise than the conven-tional attack model, which encodes data in general signals. The data were encoded using inputs that triggered wireless signals through fuzzing.

An experiment was conducted to verify the effectiveness of the encoding and leaking data into wireless signals. This experiment extracts encoded binary signals by transmitting wireless signals from a Raspberry Pi 3b+ and receiving them through HackRF.

The transmission signal for data leakage was generated using an open source, RPiTX [12]. RPiTX can convert a Raspberry Pi into an RF transmitter and generate RF signals using GPIO 4. It can generate signals from 434 MHz to 1.5 GHz through sigma-delta modulation and high-speed timing control technologies. Raspberry Pi uses an operating frequency of over 700 MHz, and the RPiTX synthesizes the fre-quency based on the clock speed of the Raspberry Pi. The RPiTX adjusts the switch-ing speed of GPIO 4 to generate the required frequency and emits a signal through an antenna physically connected to GPIO 4.

The transmission frequency was set to the default value of 434 MHz and the Sample_rate of HackRF was set to 10 MHz to receive the signal. The encoded binary signal "10101010" was transmitted for 2 s, and an offset value of 1 s was given to utilize the IDLE time between each bit. This allowed the receiver to decode the signal by distinguishing between bits. The received signal was saved as a binary file using the hackrf_transfer command in Ubuntu 20.04 and then loaded into Python for analysis. The spectrogram of the received signal is shown in Fig. 4.

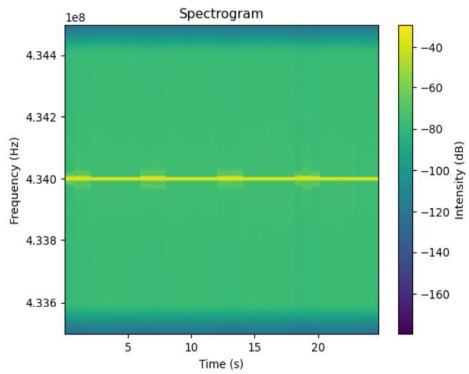


Fig. 4. Received frequency signal spectrogram.

Spectrogram analysis indicates that the received bit pattern is "10101010." After reading the signal stored in the binary file and downsampling it to improve the pro-cessing speed and reduce the data size, the processed data were output as a spectro-gram to identify changes in the frequency components. The down-sampling factor was set to 10. The X-axis of the spectrogram represents time and the Y-axis repre-sents frequency. The color bar indicates the signal strength. When "1" is transmitted, the signal strength at the center frequency of 434 MHz increases; the signal strength decreases when "0" is transmitted, which can be confirmed through the color change of the spectrogram.

5 Discussion

Unlike previous studies, this study proposes a new attack vector, proving that wireless signals enable data leakage. Because it uses a non-visible medium, it is more difficult for system users to notice a leak attack than optical or acoustic media. Additionally, rather than requiring additional media to generate a wireless signal, we discovered an input pattern that can trigger frequencies using the fuzzing technique. Fuzzing can induce abnormal behaviors in the system, which can help determine its response to an air-gapped environment.

Research on attack vectors that can be used in air-gapped networks is actively un-derway. However, research on their detection is lacking. Therefore, in future studies, we plan to develop a detection model for data leakage attacks in air-gapped net-works. We plan to conduct follow-up research to collect datasets for model training by collecting the frequency signals that are commonly transmitted in air-gapped situ-ations. We will then train the model through data processing to prove that abnormal-behavior detection is possible.

6 Conclusion

In this study, we demonstrated that data leakage attacks using side channels are feasible by fuzzing IoT connectivity devices in an air-gapped network and conducted research to analyze attack vectors. We demonstrated the feasibility of the proposed attack model by comparing the BER with that of the conventional model. The exper-iment showed that when the signal was transmitted 100 times with random noise inserted, the BER was improved by 59.43% compared with the conventional model. In addition, an experiment was conducted to transmit the leaked data by generating a wireless signal using Raspberry Pi, which is widely used in industrial settings, and to extract the leaked data by receiving the signal with a wireless signal receiver. The proposed Air-Fuzz utilizes fuzzing for attacks, making it easy to understand the re-sponse of the system in an air-gapped environment. Additionally, it transmits and receives data using wireless signals, which are an invisible medium that makes it difficult to detect attacks.

We plan to conduct follow-up research to demonstrate whether the attack model proposed in this study can be applied to commercial IoT devices under realistic industrial environmental conditions.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (IITP-2022-RS-2022-00156310) and Information Security Core Technology Development (RS-2024-00437252) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Quy, V. K., Chehri, A., Quy, N. M., Han, N. D., Ban, N. T., Innovative Trends in the 6G Era: A Comprehensive Survey of Architecture, Applications, Technologies, and Challenges. IEEE Access 11, 39824-39844 (2023).https://doi.org/10.1109/ACCESS.2023.3269297
2. Kaur, J., Ramkumar, K. R., The recent trends in cyber security: A review. Journal of King Saud University. Computer and Information Sciences 34(8), 5766-5781 (2022). https://doi.org/10.1016/j.jksuci.2021.01.018
3. Lee, S., Shim, H., Lee, Y., Park, T., Park, S., Lee, I., Study on Systematic Ransomware De-tection Techniques, 2022 24th International Conference on Advanced Communication Tech-nology (ICACT), 297-301 (2022). https://10.23919/ICACT53585.2022.9728909
4. Jeon, S., Lee, S., Lee, E., Lee, Y., Ryu, J. et al. (2023). An effective threat detection frame-work for advanced persistent cyberattacks. Computers, Materials & Continua, 75(2), 4231-4253 (2023). https://doi.org/10.32604/cmc.2023.034287
5. Miketic, I., Dhananjay, K., Salman, E., Covert Channel Communication as an Emerging Se-curity Threat in 2.5D/3D Integrated Systems. Sensors 23(2081), 2081 (2023). https://doi.org/10.3390/s23042081
6. Park, J., Yoo, J., Yu, J., Lee, J., Song, J., A Survey on Air-Gap Attacks: Fundamentals, Transport Means, Attack Scenarios and Challenges. Sensors 23(3215), 3215 (2023). https://doi.org/10.3390/s23063215
7. Miller, T., Staves, A., Maesschalck, S., Sturdee, M., Green, B., Looking backto look for-ward: Lessons learnt from cyber-attacks on Industrial Control Systems, International Journal of Critical Infrastructure Protection 35, 100464 (2021). https://doi.org/10.1016/j.ijcip.2021.100464
8. Guri, M., AIR-FI: Leaking Data From Air-Gapped Computers Using Wi-Fi Frequencies. IEEE Transactions on Dependable and Secure Computing 20(3), 25472564 (2023). https://doi.org/10.1109/TDSC.2022.3186627
9. Lee, J., Yoo, J., Lee, J., Choi, Y., Yoo, S. K., Song, J., Optical Air-Gap Attacks: Analysis and IoT Threat Implications, IEEE Network (2024). https://doi.org/10.1109/MNET.2024.3382969
10. Li, Z., Chen, B., Chen, X., et al., SpiralSpy: Exploring a Stealthy and Practical Covert Chan-nel to Attack Air-gapped Computing Devices via mmWave Sensing, Network and Distribut-ed Systems Security (NDSS) Symposium (2022). https://doi.org/10.14722/ndss.2022.23023

11. Chen, D., Chowdhury, M. M., Latif, S., Data Breaches in Corporate Setting, 2021 Interna-tional Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 01-06 (2021). https://doi.org/10.1109/ICECCME52200.2021.9590974
12. F5OEO: RPiTX: Turning a Raspberry Pi into a RF Transmitter, GitHub repository, https://github.com/F5OEO/rpitx

Fuzzing JavaScript Engines with Diversified Mutation Strategies

Cheng-Han Shie[1,2][0000−0002−9247−251X], Pin-Huang Fang[1], and Chun-I Fan[1,2,3,⋆][0000−0002−7512−1291]

[1] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
[2] Information Security Research Center, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
[3] Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung 804201, Taiwan

Abstract. Current fuzzers for JavaScript (JS) engines often produce test cases that are syntactically correct but semantically incorrect, hindering efficient crash discovery. To address this issue, we propose three mutation methods: simple mutation, state mutation, and Just-In-Time (JIT) compilation optimization triggering. The combination of these methods significantly increases the variety of seed mutations, distinguishing our approach from other fuzzers. Additionally, we analyze each JS engine to identify conditions that trigger JIT optimization, incorporating these conditions into the mutation methods to enhance path coverage. This approach achieves over 20% branch coverage, at least 5% higher than other fuzzers, with a semantic correctness rate above 80%, reducing time wasted on semantic errors. Finally, our method nearly doubles the crash discovery rate compared to other fuzzers.

Keywords: Fuzzing, Javascript engine, Abstract syntax tree (AST), Just-in-time (JIT) compilation

1 Introduction

A web browser [12] is a tool used to connect to the internet and retrieve information for interactive purposes. The browser consists primarily of a browser engine, user interface, network objects, user interface backend, data storage, and JavaScript engine. The JavaScript engine is responsible for interpreting and executing JavaScript scripts. When a webpage loads, the JS engine parses the code and creates an Abstract Syntax Tree (AST) representation. The engine then compiles the AST into machine code for the computer to execute. JS engines handle various language features, including garbage collection, optimization, and runtime analysis. They also provide APIs that allow JS to interact with the host environment, such as the Document Object Model (DOM) in web browsers.

⋆ Corresponding author

To attract more users, web browsers often offer extensions that users can freely access, such as tools for storing passwords [2], automatic browsing history classification, generating new rendering scripts based on user preferences [9], and grammar correctors. However, these tools may be developed using JavaScript code that contains zero-day vulnerabilities.

The JavaScript (JS) engine converts JavaScript code into machine language. Although each engine may have slight variations in its implementation, the overall process generally involves parsing the code, optimizing it, and then executing it on the hardware.

Common JS engines include V8 (used in Google Chrome and Node.js), SpiderMonkey (used in Firefox), and JavaScriptCore (used in Safari). Each browser's engine is implemented differently, so executing the same code may produce different results, especially with Just-In-Time (JIT) compilation. This can lead to type confusion errors exploitable for remote attacks. Identifying and addressing these vulnerabilities proactively can prevent such attacks. Fuzz testing is necessary to generate a large amount of random data, uncovering potential risks in JavaScript engines.

The approach utilized in this work is based on the architecture of a variant Fuzzer to detect and identify suitable portions for mutation in the seed, reducing the compilation errors caused by mutation testing. Its main contributions are as follows:

1. The proposed method increases the variability range by studying the attributes of the AST.
2. The proposed methods can trigger JIT optimization, leading to an increase in code coverage.
3. The proposed outperforms existing fuzzers by discovering more crashes simultaneously and achieving a lower seed compilation error rate.

The structure of this paper is as follows. Section 2 provides an introduction to the relevant background knowledge related to this paper. The proposed method is proposed in Section 3. The evaluation results, comparisons, and conclusions are presented in Section 4 and 5.

2 Preliminaries

In this section, we will introduce the background knowledge related to the proposed methods.

2.1 Just-In-Time (JIT) Compilation

JIT is a dynamic compilation technique [4] where code is compiled into machine code during runtime, rather than being compiled into machine code during the compilation phase. This allows JIT compilation to optimize the code more flexibly, as it can adapt the optimization based on the actual runtime conditions.

Furthermore, JIT compilation can also optimize for different hardware and platforms to achieve better performance. Although the optimization features may be similar across different implementations, the implementation methods can vary. As a result, each browser's JavaScript engine has its own vulnerabilities and issues.

2.2 Instrumentation

The technique involves inserting probes into the target program while ensuring the integrity of its original logic. These probes are executed during the program's execution and generate characteristic data about its runtime behavior. By analyzing this data, information about the program's control flow, data flow, and other dynamic information, such as logic coverage, can be obtained. This approach allows for effective testing of the program.

2.3 JavaScript Runtime Errors

ECMAScript [8] defines five runtime errors that can occur: RangeError, ReferenceError, SyntaxError, TypeError, and URIError. These errors are essential for the mutation strategies proposed in this paper.

- SyntaxError occurs when the JavaScript engine encounters a syntax that does not conform to the language specification.
- ReferenceError is triggered when an undefined variable is used.
- URIError indicates the improper usage of URI-related functions.
- RangeError occurs when a value that is outside the allowed range is used as a parameter in the program.
- TypeError is triggered when an operation is performed on a variable that is of a type that cannot perform that specific operation.

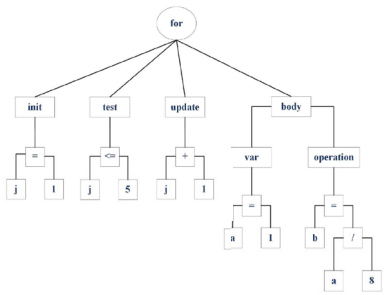
3 Methodology

This section describes the proposed mutation strategies include the simple method, the statement method and the incorporation of JIT optimization to expand the range of variations.

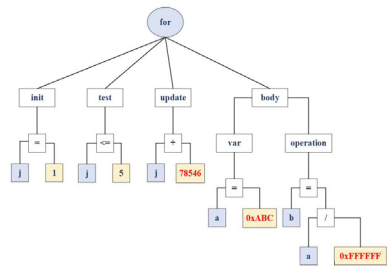
3.1 Mutating Methods

Figure 1 provides an illustrative explanation of the simple and the statement mutation methods. Figure 1a shows the conversion of a typical for loop program into an Abstract Syntax Tree (AST). Figure 1b represents the mutation method used in the Park et al. [13], which focuses on mutating only the numerical values of variables within a limited range (highlighted in yellow), and the gray blocks indicate immutable elements. The red text denotes the mutated values within the blocks. Figure 1c presents the mutation strategies in the proposed method,

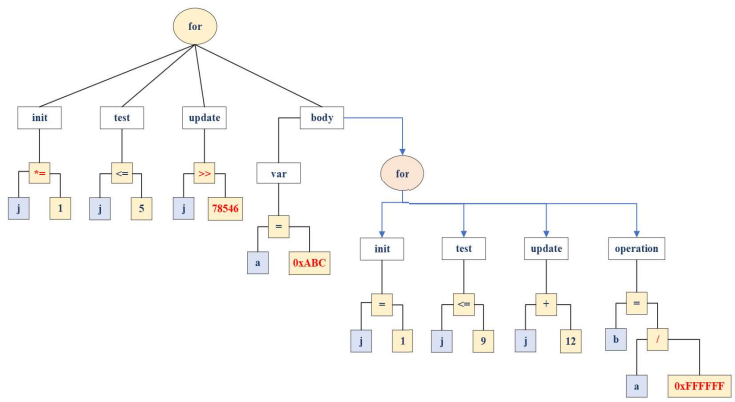
which goes beyond mutating variable values and includes mutation of operators, addition of for loops, elimination of root node for loops (resulting in only the body block of the AST), or modification to if statements (where the condition is also generated). The advantage of the method proposed in this paper is its ability to modify the seed structure, allowing for increased seed complexity. The larger the scope of variation, the higher the likelihood of triggering multiple vulnerabilities.



(a) Original AST



(b) DIE's Mutation Strategy



(c) The Proposed Mutation Strategy

Fig. 1: Comparison of AST Mutation Methods

Simple Mutation Literal, BinaryExpression, LogicalExpression, UnaryExpression, UpdateExpression, and AssignmentExpression are all targets for simple mutations in the AST. The following is a brief description of each object.

- Literal: Numbers and strings are both classified as literals. When mutating literals, it is important to consider the possibility of causing a TypeError. For numbers, the mutation range can be extended to a larger value, such as 2^{48}, which increases the likelihood of triggering a RangeError but also enhances the chances of buffer overflow vulnerabilities. For strings, the focus is on triggering format string attacks and buffer overflows. Therefore, mutations should be capable of transforming strings into ones with special characters, such as `%x`.
- BinaryExpression: The BinaryExpression encompasses both binary operators and binary conditionals, such as `==`, `>=`, `+`, and `|`. Although they fall under the same category, it is important to differentiate between binary operators and binary conditionals when applying mutations. Treating them interchangeably can lead to syntax errors. Therefore, it is crucial to accurately identify whether the program is performing calculations or conditional checks before applying mutations.
- LogicalExpression: Any logical operators, such as `&&` and `||`, fall under the category of the LogicalExpression. This category does not encounter runtime errors.
- UnaryExpression: Except for `++` and `--`, all other operators are classified as the UnaryExpression, such as $\sim$ and `!`. Unary operators are generally straightforward to use, and therefore, they do not typically result in runtime errors.
- UpdateExpression: Only the unary operators `++` and `--` belong to this category. This category can be adjusted by modifying the `prefix` property, placing `++` or `--` before or after a variable, to influence the program's flow of execution.
- AssignmentExpression: When the BinaryExpression combines with an equality symbol, it falls into the category of the AssignmentExpression. Examples of such symbols include `=, +=, |=, &=`, and so on. Due to the specification of IEEE-754[1], when a string is involved in this type of assignment operation, the result will output NaN, thus not causing a runtime error.

Statement Mutation The objects in the AST that serve as targets for state mutation include ForStatement, ForInStatement, IfStatement, DoWhileStatement, SwitchStatement, WithStatement, ContinueStatement, BreakStatement, and WhileStatement. These objects can be modified to alter the state of the program during its execution. These objects can be classified into two methods: addition and deletion.

The addition method can encompass program fragments that include a seed at any location, such as for and while loops selecting a program fragment with an assigned memory space seed or an executing operation seed. The addition method also increases the probability of triggering buffer overflow and UAF vulnerabilities. In additional, the deletion method can disrupt the original structure of the seed, allowing for greater diversity in the seed's variations. Overall, these mutation methods aim to enhance the complexity and diversity of the program, increasing the chances of triggering various vulnerabilities.

3.2 Different JIT Optimizations

JS engines operate similarly with an interpreter generating unoptimized bytecode and an optimizing compiler generating optimized bytecode. Here are the optimization strategies employed by different JS engines:

- V8: Chrome's JS engine uses Ignition as its interpreter to generate bytecode from the AST. Ignition analyzes the data and passes it to V8's optimizing compiler, TurboFan, which employs speculative optimization based on function call frequencies and parameter types. TurboFan infers operation types and simplifies them but degrades if parameter types change. The function `%OptimizeFunctionOnNextCall` can trigger optimization directly.
- ChakraCore (CH): IE's ChakraCore has two JIT compilers: SimpleJIT and FullJIT. SimpleJIT applies basic optimizations and passes complex optimizations to FullJIT. SimpleJIT triggers after 25 executions, and FullJIT triggers after 20,000 executions.
- SpiderMonkey (SM): Firefox's JS engine uses a baseline compiler with inline caching and IonMonkey for further optimizations. IonMonkey converts baseline information and bytecode into intermediate code. Testing shows JIT optimization occurs after around 1000 repeated function invocations.
- JavaScriptCore (JSC): Safari's JS engine has three JIT compilers. LLInt transitions to the Baseline compiler after 100 loops or 6 function calls. The Baseline compiler gathers type information. The DFG compiler is triggered after 1000 loops or 60 function calls, utilizing type information for further optimizations. The FTL compiler handles recursive functions, triggered after 1000 loops or 10,000 function calls, introducing advanced optimizations using LLVM.

Trigger JIT Optimization Each JS engine has its own unique optimization techniques, all of which can trigger optimization mechanisms through repeated execution, albeit with different frequencies. Therefore, when the code that triggers optimization undergoes changes in the internal variable values or types compared to the original code, the condition indicates the presence of one of the following vulnerabilities: buffer overflow, type confusion, or UAF.

3.3 Fuzzing Phase

The architecture diagram in Figure 2 represents the structure of a fuzzing. The seed (.js files) is obtained from a corpus, and the seed is converted into an AST. The AST is analyzed to identify properties that can be mutated. Corresponding mutation dictionaries are applied based on these properties. For example, the `=` operator can be mutated to `>=` or `!=` and the `+` operator can be mutated to `-` or `%`, among other possibilities. Numeric values can be mutated to larger numbers, and strings can be mutated into randomly generated strings composed of special characters. These are relatively intuitive and simple mutation methods.

To achieve more advanced types of confusion, memory overflows, and other vulnerabilities through mutation, more complex mutation techniques are required. These may involve removing loops through statement mutation methods or triggering JIT through mutation methods. The targets of the testing are the JS engines of web browsers, including V8 (Chrome), CH (Microsoft Edge), JSC (Safari), and SM (Firefox).

During the testing process, feedback information is collected, including register information, discovered paths, and overall coverage. If any abnormalities occur in the registers, the input at that moment, register information, and execution flow are recorded to generate a crash report. The paths and coverage information are used to update the corpus, improving the quality of the files in the corpus. The testing process then starts a new iteration. The duration of the execution depends on whether new paths can be found based on all the seeds, including the updated seeds, allowing the user to decide.

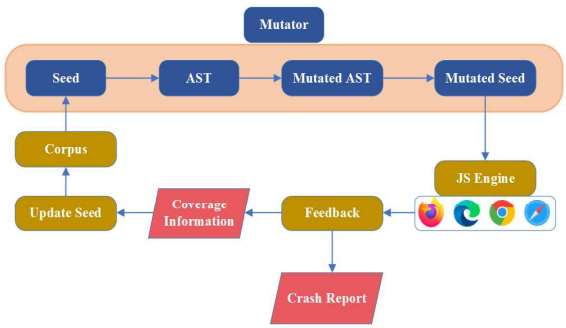


Fig. 2: Fuzzing Test Architecture Diagram

4 Evaluation Result Analysis and Comparison

In this section, the proposed scheme will be compared with other fuzzers in terms of coverage, semantic error rate, and the number of crashes discovered simultaneously. This aims to demonstrate the effectiveness of the proposed mutation method in reducing error compilation rates and finding a greater number of crashes.

4.1 Environment

The experimental environment used in this paper, which includes an Ubuntu 20.04 system, an i5-12400 CPU, and 48GB of RAM. The fuzzer architecture is

based on AFL++ (American Fuzzy Lop Plus Plus), an open-source fuzzing software. AFL++ extends the original AFL fuzzer and incorporates optimizations from previous research papers. It also supports various programming languages for developing mutation methods.

American Fuzzy Lop plus plus (AFL++) The proposed method utilizes a fuzzer framework based on AFL++ [5] . AFL++ combines AFL with optimization fuzzing techniques such as Mopt[11] and Redqueen [3] . The following are the default mutation strategies in AFL++.

- Bitflip: Flipping bits, changing 1 to 0 and 0 to 1.
- Arithmetic: Performing addition and subtraction operations on a byte-by-byte basis.
- Interest: Replacing a portion of the seed with special symbols or custom values.
- Havoc: Performing the aforementioned three actions (bitflip, arithmetic, and interest) as well as addition and deletion on the entire seed.

These mutation strategies are not suitable for use in interpreted programming languages. Therefore, these default mutation strategies will not be used during the fuzzing process.

Seed Collection Since this paper is based on mutation-based fuzzing, it requires pre-existing JS files obtained from two main sources. The first source is the open-source test suite called "js-test-suite" [10] provided by the project [17], which includes relevant files for ChakraCore, SpiderMonkey, V8, JavaScriptCore, and others. The second source is "js-vuln-db" [15], which primarily collects buffer overflow, type confusion, and UAF vulnerabilities.

Target Engines Four commonly used browsers' JavaScript engines were selected as targets for the experiments: Chrome's V8 (version 11.1.130), Edge's ChakraCore (version 1.11.24), Firefox's SpiderMonkey (version 109.0.1), and Safari's JavaScriptCore (version 2.39.5). The total lines of code for each JavaScript engine used in the experiments were 745,953 for V8, 474,641 for ChakraCore, 526,955 for SpiderMonkey, and 339,168 for JavaScriptCore. These engines include added instrumentation code during compilation to enable the detection of crashes during subsequent fuzzing.

Mutation Methods to Trigger JIT Optimization In section 3.2, the optimization mechanisms of each JS engine were discussed. However, the experimental tests considered the execution time and modified the frequency of triggering JIT optimizations. The methods to trigger different JIT modes in ChakraCore. The SimpleJIT mode, set to 25, was kept unchanged as it does not impose a significant burden on fuzzing test. However, the FullJIT mode requires 20,000

iterations to trigger. Therefore, the original code was modified before compilation to adjust the trigger to 200 iterations. To trigger the JIT optimization using the `%OptimizeFunctionOnNextCall` function, which automatically optimizes the desired code when executed. The JIT optimization frequency for SM, where the original code was looped 1,000 times. The triggering method for JSC, which was kept unchanged from its original settings.

4.2 Performance Comparison with Other Fuzzers

The Total Number of Tests Table 1 shows the total number of executions for each fuzzer over a seven-day period. The proposed method executed approximately 7 million times, while the other fuzzers executed over 7.2 million times. It can be observed that the less complex the mutation method, the lower the time taken for a single seed mutation.

Table 1: The Total Number of Tests

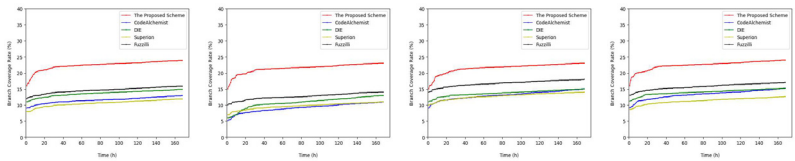
	Ours	CodeAlchemist [7]	DIE [13]	Superion [16]	Fuzzilli [6]
CH	7,157k	7,412k	7,751k	7,462k	7,394k
V8	7,014k	7,347k	7,513k	7,248k	7,326k
JSC	7,113k	7,446k	7,633k	7,319k	7,284k
SM	6,954k	7,235k	7,376k	7,367k	7,374k

Semantic Accuracy Rate Semantic accuracy rate refers to the percentage of generated seeds that can be compiled without errors. Semantically accurate seeds can trigger deeper program behaviors. Table 2 presents the semantic accuracy rates for each fuzzer based on the first 3 million seed executions. The proposed scheme achieved an average accuracy rate of 82.78%, CodeAlchemist had 33.93%, DIE [13] had 61.07%, Superion [16] had 32.16%, and Fuzzilli [6] had 66.73%. The proposed scheme shows a higher accuracy rate compared to other fuzzers because it references common semantically valid examples, maintaining the semantic validity of mutated seeds.

CodeAlchemist only eliminates some undeclared anomalies by matching functions and variables, without addressing other semantic errors, resulting in a lower accuracy rate. DIE's mutation method, similar to the proposed scheme's simple mutation, shows significantly higher semantic accuracy compared to other fuzzers. However, DIE requires pre-processing steps before fuzzing, leading to more erroneous seeds as testing time increases. Superion, using AFL's tree-based mutation method, has the lowest accuracy rate among the fuzzers. Fuzzilli uses a self-developed intermediate language for mutation, focusing on optimizations in other stages rather than the mutation method, resulting in an accuracy rate second only to the proposed scheme.

Table 2: Semantic Accuracy Rate and Branch Coverage Rate

	Semantic Accuracy Rate					Branch Coverage Rate				
	Ours	[7]	[13]	[16]	[6]	Ours	[7]	[13]	[16]	[6]
CH	**84.67%**	37.40%	61.42%	32.79%	65.13%	**23.56%**	12.27%	14.63%	11.13%	15.09%
V8	**82.39%**	29.48%	59.83%	34.15%	70.23%	**23.14%**	10.72%	12.54%	10.44%	13.26%
JSC	**81.33%**	35.73%	62.59%	27.55%	64.98%	**22.38%**	13.55%	14.41%	13.61%	17.52%
SM	**82.76%**	33.12%	60.46%	34.16%	66.58%	**22.66%**	12.28%	14.76%	12.75%	16.77%



(a) CH (b) V8 (c) JSC (d) SM

Fig. 3: The Relationship between Branch Coverage Rate and Execution Time in the JS engine.

Branch Coverage Rate The graph in Figure 3 illustrates the relationship between time and code coverage during the fuzzing of the JavaScript engine. The four subfigures can be observed that the initial code coverage ranges from approximately 8% to 15%. Afterward, the fluctuations in code coverage stabilize. Table 2 presents the branch coverage rates for the proposed scheme and various fuzzers, measured over one week. The proposed scheme achieves an average coverage rate of 22.93%. This is because the proposed scheme employs multiple complex mutation methods that provide more opportunities to reach deeper code paths. As CodeAlchemist is a generational fuzzer, the coverage rate reflects the limitation of generational fuzzers. The generated test seeds do not become complex enough, resulting in a coverage rate of only 12.28%. DIE, as a mutation-based fuzzer, slightly improves the coverage rate by mutating variables, reaching 14.08%. Superion's mutation method is prone to errors, causing only a portion of the seeds to execute successfully, resulting in a coverage rate of 11.98%. Fuzzilli achieves a coverage rate of 15.66%, ranking second after the proposed scheme.

Comparing Crash Mining Table 3 represents the number of crashes discovered during the mining process, with a statistical timeframe of one week and approximately 7 million executions per fuzzer. The proposed scheme has found the highest number of crashes across various JS engines, surpassing CodeAlchemist and DIE by more than three times and Fuzzilli by two times. However, Superion has only discovered crashes in CH.

Table 3: The Number of Unique Crashes

JS Engine	Ours	CodeAlchemist [7]	DIE [13]	Superion [16]	Fuzzilli [6]
CH	**13**	3	4	1	6
V8	**8**	1	3	0	4
JSC	**8**	1	1	0	3
SM	**6**	1	1	0	3

4.3 Experimental Conclusion

Based on the above results, it can be concluded that improving semantic correctness increases the effectiveness of most seeds and reduces wasted time on ineffective inputs. However, there is not necessarily an absolute correlation between branch coverage and semantic correctness. For example, in Table 2, CodeAlchemist has a correctness rate of 35.73% for JavaScriptCore, but its branch coverage is lower than that of Superion. This is because fuzzing involves random mutation and generation, so even with flaws in the mutation strategy, it is still possible to produce seeds that can delve deeper into the target program.

Another key aspect of improving coverage is the complexity of the seeds. Increasing complexity requires employing multiple mutation techniques. The proposed scheme utilizes 15 different mutation operations, including single variables, operators, logical symbols, control flow syntax, and triggering JIT optimization. These operations significantly increase seed complexity. Due to these mutation strategies, the proposed scheme achieves high coverage and discovers the most crashes compared to other fuzzers.

5 Conclusions and Future Works

The proposed AST-based mutation method enables a higher number of seed mutations compared to other fuzzers. Beyond basic mutation methods, it examines each JavaScript engine to identify conditions triggering JIT optimization, incorporating these into the mutations. A validation function checks the type and value of parameters post-optimization to identify potential vulnerabilities, leading to higher coverage and varied program paths. This comprehensive approach ensures the discovery of more potential vulnerabilities and achieves a semantic correctness rate above 80%, reducing wasted time on semantic errors. Additionally, it finds nearly twice as many crashes as other fuzzers.

The main contribution of this research is proposing mutation strategies to discover and trigger more crashes. Future work aims to develop an automated crash analysis and PoC generation system to streamline the crash analysis process, significantly reducing the time required and benefiting users.

Acknowledgments. This work was supported in part by the National Science and Technology Council of Taiwan under Grants NSTC 113-2634-F-110-001-MBK and 113-2221-E-110-082, in part by the Information Security Research Center at National Sun

Yat-sen University, and in part by the Intelligent Electronic Commerce Research Center from the Featured Areas Research Center Program through the Framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

References

1. Iso/iec/ieee international standard - floating-point arithmetic. *ISO/IEC 60559:2020(E) IEEE Std 754-2019*, pages 1–86, 2020.
2. Ioannis Agrafiotis, Jason RC Nurse, Michael Goldsmith, Sadie Creese, and David Upton. A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. *Journal of Cybersecurity*, 4(1), 2018.
3. Cornelius Aschermann, Sergej Schumilo, Tim Blazytko, Robert Gawlik, and Thorsten Holz. Redqueen: Fuzzing with input-to-state correspondence. In *NDSS*, volume 19, pages 1–15, 2019.
4. Aurèle Barrière, Sandrine Blazy, Olivier Flückiger, David Pichardie, and Jan Vitek. Formally verified speculation and deoptimization in a jit compiler. *Proceedings of the ACM on Programming Languages*, 5(POPL):1–26, 2021.
5. Andrea Fioraldi, Dominik Maier, Heiko Eißfeldt, and Marc Heuse. AFL++: Combining incremental steps of fuzzing research. In *14th USENIX Workshop on Offensive Technologies (WOOT 20)*. USENIX Association, August 2020.
6. Samuel Groß, Simon Koch, Lukas Bernhard, Thorsten Holz, and Martin Johns. Fuzzilli: Fuzzing for javascript jit compiler vulnerabilities. In *Network and Distributed Systems Security (NDSS) Symposium*, 2023.
7. HyungSeok Han, DongHyeon Oh, and Sang Kil Cha. CodeAlchemist: Semantics-aware code generation to find vulnerabilities in javascript engines. In *Network and Distributed System Security Symposium*, 2019.
8. Ecma International. ECMA-262, 13th edition, June 2022 ECMAScript® 2022 Language Specification. 2022.
9. D Rajya Lakshmi and S Suguna Mallika. A review on web application testing and its current research directions. *International Journal of Electrical and Computer Engineering*, 7(4):2132, 2017.
10. leeswimming. [online] WSP-LAB js-test-suite. 2020.
11. Chenyang Lyu, Shouling Ji, Chao Zhang, Yuwei Li, Wei-Han Lee, Yu Song, and Raheem Beyah. {MOPT}: Optimized mutation scheduling for fuzzers. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1949–1966, 2019.
12. Rebecca Nelson, Atul Shukla, and Cory Smith. Web browser forensics in google chrome, mozilla firefox, and the tor browser bundle. *Digital Forensic Education: An Experiential Learning Approach*, pages 219–241, 2020.
13. Soyeon Park, Wen Xu, Insu Yun, Daehee Jang, and Taesoo Kim. Fuzzing javascript engines with aspect-preserving mutation. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1629–1642, 2020.
14. Filip Pizlo. [online] Introducing the WebKit FTL JIT. 2014.
15. tunz. [online] Case Study of JavaScript Engine Vulnerabilities. 2019.
16. Junjie Wang, Bihuan Chen, Lei Wei, and Yang Liu. Superion: Grammar-aware greybox fuzzing. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 724–735, 2019.
17. Suyoung Lee, HyungSeok Han, Sang Kil Cha, and Sooel Son. Montage: A neural network language model-guided javascript engine fuzzer. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 2613–2630, 2020.

Detecting Phishing-targeted Web Push Notifications through Image Similarity Analysis

Rion Matsuzaki[1] and Masaya Sato[1][0000−0002−3944−2012]

Okayama Prefectural University, Soja, Okayama, Japan
masaya@c.oka-pu.ac.jp

Abstract. Web push notifications are used to induce users to phishing websites. To prevent this induction, keyword-based detection has been proposed. However, the words used in phishing web push notifications differ from those used in phishing websites. Thus, generating a keyword list for detecting phishing web push notifications is challenging. In this paper, we propose a method for detecting phishing-targeted web push notifications through image similarity analysis. Phishing-targeted web push notifications mostly contain images in their data. We utilized this image data to detect phishing-targeted web push notifications. In our prototype, we used mitmproxy to capture image data and analyzed image similarity using structural image similarity. Experimental results showed that our prototype can detect phishing-targeted web push notifications with high accuracy.

Keywords: Phishing detection · Web push notification · Image similarity.

1 Introduction

Damages caused by phishing websites are increasing, and adversaries are inducing users through various methods. From the report of Anti-Phishing Working Group, 2023 is the worst year for phishing [4]. Adversaries create phishing websites by imitating legitimate websites to deceive users and steal personal or account information. The number of unique phishing websites is 350,000 or more; however, a single phishing website may be advertised through various methods, including email, ads, notifications, or phone-based methods. Consequently, countermeasures against phishing attacks remain a challenging problem. In the early stages, the method of inducing users to phishing sites was often through email; however, the methods have now diversified to include short message service (SMS), social network service (SNS), two-dimensional barcodes (including QR codes), and push notifications.

Induction using web push notifications is an emerging threat. A web push notification is a function to display a notification from a website. A web browser supporting web push notifications can display a notification on the desktop of the computer by receiving messages from push servers. The sender can set an action to the notification, thus, the user may be induced to phishing websites by

clicking the displayed notification. Due to the fact that attacks using web push notifications for malware distribution or induction to phishing websites have been observed, countermeasures against this induction have become important.

The difficulty in detecting and preventing malicious web push notifications that induce users to phishing websites lies in the challenge of distinguishing whether a notification is malicious or not. The display area of notifications is limited and small, which means the information available to distinguish malicious notifications is insufficient. In addition, users cannot grasp the actions bound to the notification buttons because there is no correspondence between the label shown on the button and the actual action. As a result, clicking any button may induce the user to malicious websites.

In this paper, we propose an image-based method to detect malicious web push notifications. Web push notifications contain not only text but also image data. Visual information is frequently used to deceive, frighten, and confuse users in order to induce them to malicious websites. We hypothesized that the images used for these types of attacks have strong similarity. In fact, images used in phishing websites are similar to those on legitimate websites, whereas legitimate websites are not similar to each other. For this reason, we developed a method to calculate the image similarity between the images in web push notifications and malicious images. In this method, web push notifications with images similar to malicious images are blocked and not displayed. This protects users from being induced to phishing websites by clicking the notifications. This paper presents the design, implementation, and evaluation results of the proposed method.

2 Background

2.1 Phishing

Phishing is a deceptive attack targeting the theft of account information or the distribution of malware. A phishing site is a website used for phishing attacks. In many cases, phishing sites mimic legitimate companies' websites to deceive users into believing the site is legitimate. The increasing damages caused by phishing attacks highlight the importance of countermeasures against these attacks.

To induce users to phishing sites, adversaries exploit various methods including email, SMS, SNS, and push notifications. Countermeasures for attacks exploiting email, SMS, and SNS have been proposed [14, 18]. However, countermeasures against deceptive web push notifications are insufficient. Specifically, the prevention of displaying deceptive web push notifications has not been well studied.

2.2 Web push notification

A web push notification is a method for displaying notifications on the desktop from a website. Once a user permits sending notifications from a website, the website can send a push notification to the notification area of the desktop

through the web browser. A web push notification consists of a title, a body, an icon, a body image, and an action title. The behavior and design of the notified information differ across each OS and web browser.

There are two methods to display a web push notification: the Notifications API [11] and the Push API [12]. In the Notifications API, a web push notification is displayed through the Notification constructor or a Service Worker. The Push API uses a Service Worker to display a web push notification. The Service Worker is a thread that executes JavaScript programs independently from the thread that handles the display of websites. The Service Worker displays the notification by receiving messages from a push server. Additionally, the JavaScript program executed by the website thread can use the Service Worker to display a notification. For security reasons, the Service Worker communicates exclusively through HTTPS.

Figure 1 shows the flow of a web push notification using a Service Worker. The `Notification.requestPermission()` method is used to request a user for permission to send a web push notification. Once a user permits, a Service Worker is registered through communication with a Push Server. The communication related to a web push notification using the Push API occurs between the Service Worker and the push server. This means that the communication related to a web push notification bypasses the thread for the website. A web push notification using the Notifications API can also use a Service Worker method named `ServiceWorkerRegistration.showNotification()`. This method is common in both the Notifications API and the Push API.

`ServiceWorkerRegistration.showNotification()` method has two arguments: `title` and `options`. The title of the notification is stored in `title` (mandatory) and other data are stored in `options` (optional). Table 1 shows the properties available for setting in `options`.

2.3 Web push notifications inducing users to phishing websites

Web push notifications are used to induce users to malicious websites. Lee et al. analyzed relation between web push notifications and malicious websites [7]. Their analysis results show that third-party web push services are used for phishing notifications. In the report, rise of web push notifications with corporation and brand logos may cause malicious web push notification using imitated logo images by attackers.

To verify the above results and collect data from malicious websites, we collected and analyzed web push notifications that induce users to phishing websites. The analysis results showed that malicious web push notifications tend to use wording that stirs up anxiety in users. Specifically, we found that all the malicious web push notifications in our survey have an icon and data in their `options`. The `body` is also included at a high rate.

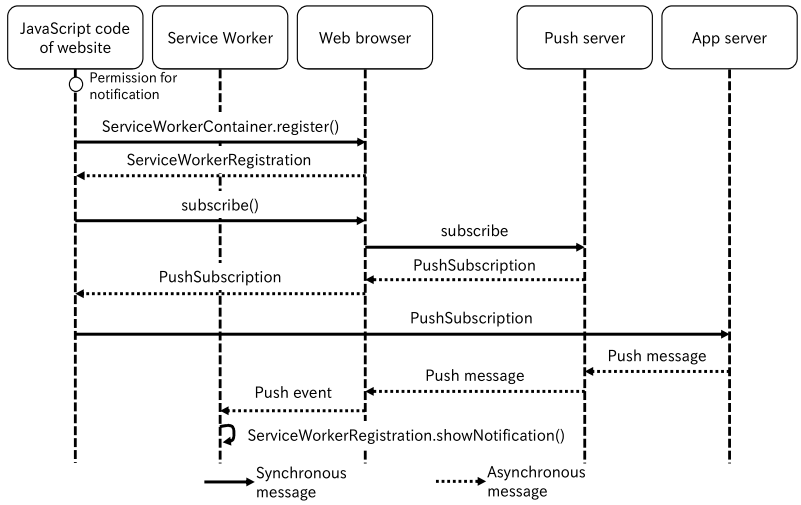


Fig. 1. Communication flow of web push notification.

2.4 Countermeasures and challenges

We have tried to detect malicious web push notifications using a keyword-based approach. This method monitors the JavaScript programs of websites and prevents the display of malicious web push notifications if they include words used for phishing websites. However, through our survey on malicious web push notifications and phishing websites, we found that constructing the keyword list is difficult in practice. In our survey, we found that the words used in the two groups, malicious web push notifications and phishing websites, had few commonalities, making it difficult to detect malicious notifications based on these words.

Based on the postmortem of our previous work and the analysis results in Section 2.3, we have summarized the observations for detecting malicious web push notifications as follows:

(O1) Keyword-based detection is not practical

As previously stated, malicious web push notifications and phishing websites have few commonalities in their wording. Additionally, constructing a malicious keyword dataset from malicious web push notifications is difficult due to the limited number of such notifications. Collecting a sufficient number of malicious web push notifications is challenging due to the nature of push notifications. To collect data on web push notifications, we need to develop a crawler that automatically visits websites, permits web push notifications, waits for the receipt of notifications, and finally determines whether the received notifications are legitimate or malicious. While Subramani et al. [19]

Table 1. Properties that can be set in the options for ServiceWorkerRegistration.showNotification()

Property	Description
body	A string representing additional content to be displayed within the notification
icon	A string containing the URL of an image to be used as an icon in the notification
image	A string containing the URL of an image to be displayed in the notification
actions.title	A string containing the action text to be displayed to the user
actions.icon	A string containing the URL of an image to be used as an icon in the action
badge	A string containing the URL of an image to represent the notification when there is not enough space to display the notification itself
data	Any data associated with the notification
dir	The direction of the notification
lang	The language used within the notification
silent	If set, no sound or vibration will occur
tag	A given notification ID to allow the script to search, replace, or delete the notification as needed
timestamp	The time associated with the notification, in milliseconds since UNIX epoch
renotify	A boolean value indicating whether to suppress vibration and audible alerts when reusing the tag value
vibrate	The vibration pattern to execute when displaying the notification

developed such a crawler, labeling each notification as legitimate or malicious remains a challenging problem.

(O2) Data inside web push notifications are not sufficiently used
As stated in Section 2.2, web push notifications contain various data within their options. However, our previous approach focused only on the `title` and `body`due to the characteristics of the keyword-based approach.

3 Detection of suspicious web push notifications by image similarity

3.1 Detection method using image similarity

Based on these observations, we decided to exploit the *image* data of web push notifications to protect users from the induction using malicious web push notifications. We propose a method to detect malicious web push notifications using image similarity. As stated in the previous section, malicious web push notifications tend to include images in their notification data. In addition, the image similarity scores among legitimate websites are low, but phishing websites are

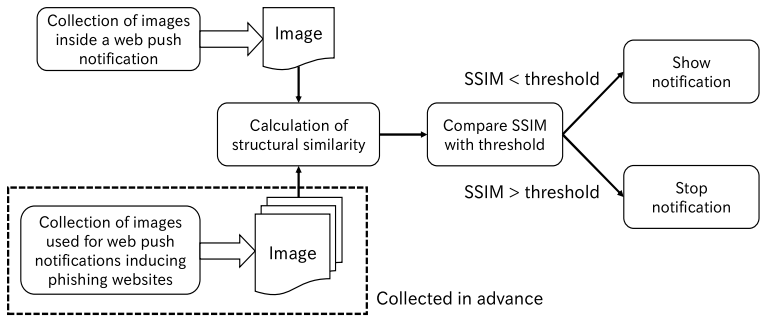


Fig. 2. Overview of image-based detection of suspicious web push notifications.

similar to legitimate websites. From this observation, we hypothesized that images of malicious web push notifications have strong similarity. Here, we define a malicious web push notification as a web push notification that induces users to phishing websites. For this reason, we propose a method to detect malicious web push notifications by focusing on image similarity.

Figure 2 shows an overview of the proposed method. The proposed method monitors images in web push notifications and calculates the similarity between the received image and pre-collected malicious images. Here, malicious images are collected in advance from known malicious web push notifications. If the similarity exceeds a certain threshold, the notification containing the image is detected as malicious and its display is prevented.

The proposed method does not employ the keyword-based detection method. This means that bypassing detection by changing keywords has no effect on the detection rate. Even if adversaries try to change their images for evasion, the proposed method is resistant to such types of evasion. If adversaries change their images slightly, the proposed method can still detect them as long as the images are similar to the original ones. As long as the threshold is not disclosed to adversaries, they cannot evade detection by the proposed method. Changing images significantly is much more costly compared to changing keywords. Additionally, the cost of preparing new images for phishing is also high, and this could affect their ability to induce users to phishing websites. For the above reasons, we believe that the proposed method has a certain degree of effectiveness in deterring attacker activities.

3.2 Image similarity

To calculate the image similarity, we used structural similarity (SSIM) [2]. The SSIM score is an indicator for evaluating the similarity between two images, considering differences in structure, luminance, and contrast. To calculate the similarity, the image is divided into small blocks, and these items are compared. Finally, the overall similarity is summarized using the scores of the small blocks.

Table 2. Items in SSIM equation

Item	Description
μ_x	Average luminance of image x
μ_y	Average luminance of image y
σ_x	Standard deviation of image x (variation of luminance)
σ_y	Standard deviation of image y (variation of luminance)
σ_{xy}	Covariance between images x and y (correlation of luminance)
C_1, C_2	Small constants (for stability improvement)

The higher the SSIM score, the higher the similarity. The upper bound of the SSIM score is one. The equation for the SSIM score is shown in Equation (1), and each item is detailed in Table 2. In our prototype, we used OpenCV to calculate the SSIM score, and the default values of OpenCV were employed for C_1 and C_2.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{1}$$

4 Implementation

4.1 Monitoring of HTTP communication

The proposed method collects images related to web push notifications and prevents the display of the notification if the image inside it has a high similarity to malicious images. To implement this system, we employed monitoring of HTTP communications related to web push notifications. Browser extensions and browser modifications are also considerable methods for the system; however, browser extensions cannot collect the required information and browser modifications require high costs for development and installation. For these reasons, we employed HTTP monitoring to capture images related to web push notifications.

We use *mitmproxy* [15] to monitor HTTP communications on a client computer. Mitmproxy is a proxy tool that can monitor and modify the communication data in the the computer. Mitmproxy can monitor secured HTTP (TLS) communications with HTTP/1, HTTP/2, and WebSocket. It also supports a Python API for monitoring, modification, and replay of communications. Thus, we implemented the proposed method as a Python program using the Python API of mitmproxy. Figure 3 shows the flow for monitoring HTTP communication using mitmproxy. The program monitors HTTP responses, saves images related to web push notifications, calculates the similarity of images, and prevents the display of notifications.

4.2 Procedure for detection

Figure 4 shows the flow for controlling the display of web push notifications. We implemented the following flow in the Python script shown in Figure 3.

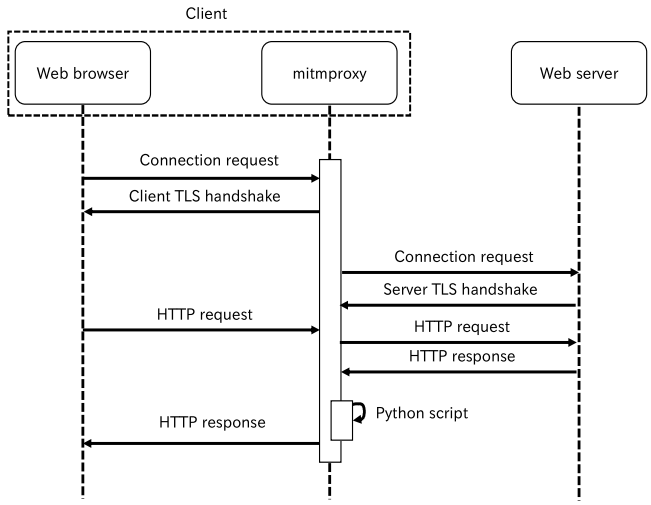


Fig. 3. Monitoring HTTP communication using mitmproxy

1. Image collection
   This part collects images inside a web push notification. Due to the difficulty of identifying which HTTP response is for images of web push notifications, we collected URLs for images included in responses for web push notifications (detailed in (2)). First, we monitor the `Content-Type` of HTTP responses. If the `Content-Type` is an `image`, we search for the URL from the collected URLs included in web push notifications. If we find that the response matches the collected URL, we save the received image. Finally, this part passes the acquired image to part (3).
2. URL collection
   Searches for `showNotification()` and `Notification` Constructor in the response. If either is found, save the URLs of the images. Then, this part returns the processing to part (1).
3. Calculation of image similarity
   This part compares the images collected in (1) with the malicious images collected in advance. We used structural similarity (SSIM) for comparison, and this part passes the SSIM value to part (4).
4. Control for showing notifications
   This part determines whether the web push notification is suspicious or not. If the SSIM score is greater than the threshold, this part stops the notification from being displayed. Otherwise, the notification is displayed.

The threshold in part (4) must be determined through experiments. We set the threshold at 0.8, considering the results in [6], which imply that legitimate sites use different (not similar) images, whereas malicious websites use images similar to those of legitimate sites.

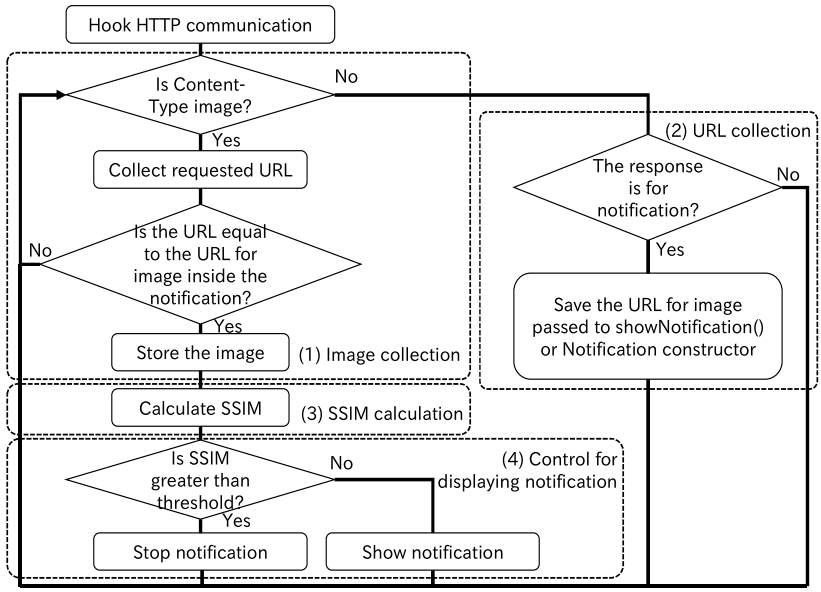


Fig. 4. Flow for control of showing web push notifications

Table 3. Environment for evaluation

CPU Core i5-8265U
Memory 8 GB
Web browser Google Chrome ver. 113.0.5672.93

5 Evaluation

5.1 Purpose and environment

We evaluated the proposed method from the viewpoint of the detection ability of malicious web push notifications with images. In addition, we measured the time required to detect malicious web push notifications. The environment for evaluation is shown in Table 3. In the evaluation, we used Windows 11 and Google Chrome for receiving push notifications. To monitor HTTP communications, we used mitmproxy and implemented our prototype as a Python script using the Python API of mitmproxy.

5.2 Accuracy for detection

We evaluated the accuracy of detecting malicious web push notifications. Prior to the evaluation, we collected websites that send web push notifications using PublicWWW [16]. To collect websites that send web push notifications, we used

Table 4. Detection results of malicious web push notifications. The number in parentheses indicates the total number of malicious web push notifications in our dataset. This includes notifications not targeted by our proposed method.

	# items
Websites sending web push notifications	257
Malicious web push notifications	5 (27)
Detected web push notifications	5

`Notification.requestPermission()` and `navigator.serviceWorker.register()` as search keywords. `Notification.requestPermission()` is for requesting notification permission, and `navigator.serviceWorker.register()` is for registering a service worker. Websites need to use these methods to send web push notifications.

Table 4 shows the results of our collection of websites with notifications. We collected 257 websites and found 27 malicious websites sending web push notifications that induce users to phishing websites. Through our evaluation, we confirmed that all malicious web push notifications are detected by our proposed method. Note that our prototype could detect 5 notifications but could not detect 22 websites due to implementation difficulties. Our prototype can detect URLs for images with an absolute path. However, some push notifications set the URL with a relative path or a variable. Additionally, we confirmed that no HTTP communication occurs if the same images are used in multiple notifications. In such cases, the web server returns HTTP 304, and thus, the web browser receives no images. As a result, the proposed method cannot collect the URL for images, leading to detection failure. Detecting these types of URLs is our future work.

5.3 Performance evaluation

We measured the performance of the proposed method. The proposed method prevents the display of malicious web push notifications using image similarity. This means that the user experience will degrade if the calculation of image similarity takes a long time. To evaluate whether the proposed method worsens the user experience or not, we measured the processing time of the proposed method. The detection rate of malicious push notifications depends on the number of malicious images used for detection; however, the greater the number of malicious images, the longer it takes to calculate image similarity. Thus, we measured the processing time of the proposed method by varying the number of malicious images used for detection from one to forty. We measured the time from when a malicious web push notification is received by the web browser to when the proposed method finishes making the decision to show the push notification. The processing time changes depending on whether the received web push notification contains an image for malicious notifications or not. Therefore, we measured the processing time both with and without an image for malicious notifications.

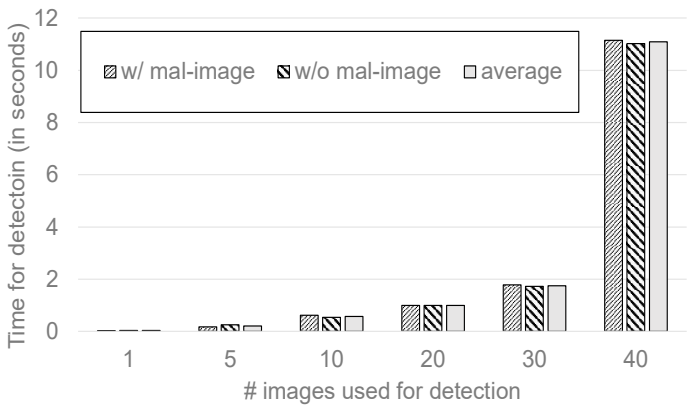


Fig. 5. Processing time with our proposed method.

Figure 5 shows the measurement results. The results show that it takes more than one second to decide whether the received notification is malicious or not when there are more than twenty images for detection. Specifically, it takes more than ten seconds when using forty malicious images for detection. According to Core Web Vitals [10], the user experience worsens if the loading time exceeds 4,000 milliseconds. Note that we carefully adhere to this standard; however, the time with forty malicious images is not acceptable. We need further evaluation in more complicated situations, including frequent notifications and large and complex images.

6 Related work

Text features are conventionally used for phishing detection [20, 17, 9]. Term Frequency-Inverse Document Frequency (TF-IDF) and heuristics are used to detect malicious websites from the body text [20]. Rosiello et al. exploit the structure of websites to enhance the detection of malicious websites [17]. They focus on the Document Object Model (DOM) to understand the layout of websites. In addition to body text and DOM, CSS is also a helpful feature for detection [9]. DOM and CSS strongly affect the structure of websites, and adversaries utilize these features to imitate legitimate websites. While these features are considered helpful for detecting malicious web push notifications, the entire document of a web push notification is shorter and less diverse compared to regular websites. Therefore, focusing on web push notification-specific features is required.

Visual features are used to detect malicious websites [13, 3]. Medvet et al. utilize similarities in text, images, and the overall appearance of websites to identify malicious websites [13]. With these features, they achieved no false positives, although two sites failed to be detected. An image-capture based method using optical character recognition (OCR) for malicious website detection has

also been proposed [3]. While the OCR-based method differs slightly from the visual-based detection scheme, it demonstrates that visual features can be helpful for detecting malicious websites. These contributions show that image-based methods are effective for detecting malicious websites; however, no work has yet been applied to the detection of malicious web push notifications.

Subramani et al. analyzed ads using web push notifications [19]. Through their analysis, they found and reported that web push notifications are also used for malware distribution. We built a prototype to monitor web push notifications similar to their work, controlling the display of notifications; however, the granularity of acquirable information differs from their work. Their work mainly focused on the automation of analysis, modifying the Chromium web browser and employing Puppeteer. Our approach uses mitmproxy to monitor and modify HTTP responses; thus, our prototype cannot collect information inside web browsers. This means that while our proposed method is less informative, it is more portable to other environments.

There are various image features available for detection [8, 1, 5]. In this paper, we employed SSIM [2] due to the balance of speed and accuracy; however, we plan to evaluate and adopt the optimal algorithm in terms of speed and detection accuracy.

7 Conclusions

We proposed an image-based detection method for suspicious web push notifications that induce users to phishing websites. The proposed method collects images from web push notifications and detects suspicious ones based on structural similarity between the image and malicious images. If the collected image has a higher SSIM score with malicious images, the proposed method stops displaying the notification to protect users from being induced to phishing sites. Experimental results showed that the proposed method can detect and prevent the display of malicious web push notifications. However, some images are not captured by the proposed method due to the difficulty of monitoring URLs with complex structures or variables. In addition, the performance overhead of the proposed method is not negligible. Our future work includes comprehensive monitoring of images and performance improvements.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. pp. 404–417. Springer (2006)
2. Brunet, D., Vrscay, E.R., Wang, Z.: On the mathematical properties of the structural similarity index. IEEE Transactions on Image Processing **21**(4), 1488–1499 (2011)
3. Dunlop, M., Groat, S., Shelly, D.: Goldphish: Using images for content-based phishing analysis. In: 2010 Fifth international conference on internet monitoring and protection. pp. 123–128. IEEE (2010)

4. Group, A.P.W.: Phishing attack trends report – 4q 2023. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf
5. Huang, C.R., Chen, C.S., Chung, P.C.: Contrast context histogram—an efficient discriminating local descriptor for object recognition and image matching. Pattern Recognition **41**(10), 3071–3077 (2008)
6. Le-Nguyen, M.K., Nguyen, T.C.H., Le, D.T., Nguyen, V.H., Tôn, L.P., Nguyen-An, K.: Phishing website detection as a website comparing problem. In: SN Computer Science. vol. 4 (2023)
7. Lee, J., Kim, H., Park, J., Shin, I., Son, S.: Pride and prejudice in progressive web apps: Abusing native app-like features in web applications. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1731–1746 (2018)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**, 91–110 (2004)
9. Mao, J., Tian, W., Li, P., Wei, T., Liang, Z.: Phishing-alarm: Robust and efficient phishing detection via page component similarity. IEEE Access **5**, 17020–17030 (2017)
10. McQuade, B.: Defining the core web vitals metrics thresholds. https://web.dev/articles/defining-core-web-vitals-thresholds
11. MDN Web Docs: Notification - web api | mdn. https://developer.mozilla.org/ja/docs/Web/API/Notification
12. MDN Web Docs: Push api - web api | mdn. https://developer.mozilla.org/ja/docs/Web/API/Push_API
13. Medvet, E., Kirda, E., Kruegel, C.: Visual-similarity-based phishing detection. In: Proceedings of the 4th international conference on Security and privacy in communication netowrks. pp. 1–6 (2008)
14. Mishra, S., Soni, D.: Dsmishsms-a system to detect smishing sms. In: Neural Computing and Applicatoins. vol. 35, pp. 4975–4992 (2023)
15. mitmproxy: Github - mitmproxy. https://github.com/mitmproxy/ mitmproxy
16. PublicWWW: Search engine for source code-publicwww.com. https://publicwww.com
17. Rosiello, A.P., Kirda, E., Ferrandi, F., et al.: A layout-similarity-based approach for detecting phishing pages. In: 2007 third international conference on security and privacy in communications networks and the workshops-securecomm 2007. pp. 454–463. IEEE (2007)
18. Seng, S., Al-Ameen, M.N., Wright, M.: Understanding users' decision of clicking on posts in facebook with implications for phishing. In: Workshop on Technology and Consumer Protection (ConPro 18) (2018)
19. Subramani, K., Yuan, X., Setayeshfar, O., Vadrevu, P., Lee, H., K., Perdisci, R.: When push comes to ads: Measuring the rise of (malicious) push advertising. In: Proceedings of the ACM Internet Measurement Conference. pp. 724–737 (2020)
20. Zhang, Y., Hong, J.I., Cranor, L.F.: Cantina: a content-based approach to detecting phishing web sites. In: Proceedings of the 16th international conference on World Wide Web. pp. 639–648 (2007)

Dissecting the Infrastructure Used in Web-based Cryptojacking: A Measurement Perspective

Ayodeji Adeniran, Kieran Human, and David Mohaisen

University of Central Florida, Orlando. USA

Abstract. This paper conducts a comprehensive examination of the infrastructure supporting cryptojacking operations. The analysis elucidates the methodologies, frameworks, and technologies malicious entities employ to misuse computational resources for unauthorized cryptocurrency mining. The investigation focuses on identifying websites serving as platforms for cryptojacking activities. A dataset of 887 websites, previously identified as cryptojacking sites, was compiled and analyzed to categorize the attacks and malicious activities observed. The study further delves into the DNS IP addresses, registrars, and name servers associated with hosting these websites to understand their structure and components. Various malware and illicit activities linked to these sites were identified, indicating the presence of unauthorized cryptocurrency mining via compromised sites. The findings highlight the vulnerability of website infrastructures to cryptojacking.

Keywords: Cryptojacking, cryptocurrencies, mining.

1 Introduction

A cyberattack known as "cryptojacking" occurs when unauthorized individuals or entities utilize another's computer resources to mine cryptocurrencies. This attack type is also called cryptocurrency mining malware or malicious crypto mining. To maintain the security and validity of blockchain transactions, mining cryptocurrencies requires solving intricate mathematical problems, which demand a considerable amount of computational power and energy [16].

A cryptojacking attack allows an adversary to gain unauthorized access to a computer, server, or network of devices and installs malicious software on them [14]. This software aims to use the computing resources of the compromised systems for cryptocurrency mining. This activity takes place without the owners' knowledge or permission. The cryptocurrency industry has, unfortunately, been plagued by malicious activities, and cryptojacking remains a significant threat [13]. Due to the high costs of establishing and maintaining cryptocurrency infrastructures, attackers frequently exploit platforms owned by others to carry out their nefarious activities [6]. By commandeering existing infrastructures, they can launch attacks against their desired targets without incurring any of the associated expenses.

The process of cryptocurrency mining is both intricate and demanding, calling for substantial investment. Unfortunately, potential profits often tempt attackers who actively search for systems that can generate the greatest returns [11].

In the past, private servers—which typically consume vast amounts of energy—were primarily used for mining. Nowadays, cloud-based servers offer more accessible, cost-effective options. However, some miners will hijack others' infrastructures to maximize their profits.

This paper aims to analyze and understand the infrastructure of certain websites previously linked to cryptojacking activities. We aim to determine whether these websites remain malicious through cryptojacking malware or other malicious software. We also seek to understand the geographical distribution of these sites and extract additional information to gain insights into their operations. Furthermore, we will discuss the security risks and threats associated with cryptojacking infrastructures and activities.

2 Related Work

Numerous studies have explored different facets of cryptojacking [26]. Jayasinghe *et al.* [12] explore cryptojacking within public cloud infrastructures, providing insights into how these environments are exploited for illicit cryptocurrency mining. Meanwhile, Saad *et al.* [24] analyze end-to-end in-browser cryptojacking, examining how cryptojacking scripts operate within web browsers. Both studies significantly inform our understanding of cryptojacking across different platforms and contexts. Burgess *et al.* [5] developed MANiC (Multi-step Assessment for Crypto-miners), a system designed to detect cryptocurrency mining scripts. MANiC extracts parameters that can be used to identify suspicious behaviors associated with mining activities, although it does not focus on web infrastructure. Carlin *et al.* [7] conducted a similar study focused on detecting cryptojacking websites by dynamic opcode analysis. Xiao *et al.* [29] explored GPU cryptojacking and introduced MagTracer, a detection system for GPU-based cryptojacking. MagTracer boasts a detection accuracy of 98%.

Naseem *et al.* [21] proposed MINOS, a lightweight detection system designed to identify cryptojacking activities in real time. Tekiner *et al.* [27] presented an approach for detecting cryptojacking in IoT environments. Saad *et al.* [25] conducted an in-depth analysis of cryptojacking samples, focusing on content, currency, and code-based categorization. Rajba *et al.* [23] presented an analysis highlighting the limitations of web cryptojacking detection methods. Lachtar *et al.* [15] discussed a cross-stack approach to defending against cryptojacking.

Research Gap. Our research paper distinguishes itself from others by focusing on websites engaged in cryptojacking. While other studies concentrate on detecting these websites, we thoroughly analyze the infrastructure, including their geographic distribution and other pertinent characteristics. Additionally, we perform a follow-up scan to identify any malicious content or malware and reclassify them as benign or malicious.

3 Problem Statement and Research Questions

Generating a new cryptocurrency involves mining [10]. This complex process includes solving intricate mathematical problems that ensure the validation and security of transactions on the blockchain network [1]. The process of mining

cryptocurrency differs from one type to another, and using consensus algorithms, such as Proof of Work (PoW) used by Bitcoin, and Proof of Stake (PoS), which determine how new blocks are added to the blockchain [4].

Mining cryptocurrencies demands substantial computational power, particularly in PoW systems. Miners must have access to robust hardware to solve intricate mathematical problems, which has led to the emergence of cryptojacking. We have analyzed the components and properties of cryptojacking infrastructures to gain insight into their operations. Our goal is to address the challenges posed by these infrastructures by answering three crucial research questions.

1. **RQ1: What are the underlying affinities between websites and malicious content, and how can these affinities be measured to assess the potential for cryptojacking activities?**. This enables us better to understand the relationship between the websites and malicious content.
2. **RQ2: what categories of malicious content and malware are common in these cryptojacking infrastructures?** We analyzed various websites and categorized different types of malicious content and malware commonly associated with crypto-jacking activities. Our findings can provide valuable insights into the different categories of threats.
3. **RQ3: What are the hosting patterns and the geographical distribution of cryptojacking infrastructure?** The identified malicious websites are hosted in various countries globally. Analyzing this distribution will provide insights into the geographical spread of these sites.

4 Technical Approach

This study analyzed websites utilizing mining scripts for cryptocurrency mining to gain insights into the nature of cryptojacking infrastructures. It aimed to explore potential correlations between cryptojacking and other malicious activities, such as phishing campaigns [17], malware distribution networks [20,18,8,3], or botnets [28]. The analysis began with the `Whois` tool to gather comprehensive domain information, including IP addresses, name servers, and registrars.

We obtained a dataset of 887 websites associated with cryptocurrency mining from the MANiC dataset and subjected them to a comprehensive scan for malicious content using `VirusTotal`. Out of the 887 websites, we successfully scanned approximately 880 of the websites. The results revealed that 371 websites were clean, while 518 contained malicious content. Our objective is to analyze the 518 websites detected to contain malicious content. This analysis aims to extract valuable information regarding the types of malicious content, the security engines that identified the threats, and the specific categories classified as malicious or suspicious. By decisively examining these factors, we assert our commitment to understanding the nature and extent of the threats posed by these websites.

4.1 Dataset and Preprocessing

Associated Cryptojacking websites. This paper used data from the "MANiC: Multi-step Assessment for Crypto-miners" dataset, which is also referenced in

the paper titled "Detecting Cryptomining through Dynamic Analysis." The data was initially collected from the top 1 million websites on Alexa, as indexed by Censys, in July 2018. The dataset consists of multiple files, and our analysis focused on the malicious content file to examine its specifics.

After collecting the data in 2018, we conducted a re-scan to identify malicious content. Our re-scan revealed that 518 websites now contained malicious content. We utilized a domain query service to gather comprehensive information about all the websites in the dataset. This service allowed us to extract detailed domain information and other relevant data. Subsequently, we scanned the entire dataset to differentiate between infected and clean websites. This process aimed to enhance our understanding of the current threat landscape associated with cryptojacking and other malicious activities.

Security Data Attributes. Afterward, we further analyzed the websites in the dataset with malicious infections by scanning them with VirusTotal. This analysis allowed us to obtain information about the security engines performing the scans and categorize the malicious detections into distinct groups. The output from the security engines is classified into three components: method, category, and result. VirusTotal uses the blacklist method to identify potentially harmful websites with malicious content [19]. The security engine category can indicate suspicious or malicious activity, and the scan results can classify the content as malicious, suspicious, malware, phishing, or spam. We further analyzed the different types of malware and malicious content detected by each security engine. Various categories of malicious content were classified based on their threat levels and purposes. It is important to note that not all malicious content is related to cryptojacking activity. This classification helps us identify and concentrate on the malware and other malicious content that may be directly associated with cryptocurrency mining. Through this detailed analysis, we aimed to understand the types of threats these websites pose and differentiate between cryptojacking-related activities and other forms of malicious content.

4.2 Analysis Dimensions

This study investigates the infrastructure of cryptojacking by analyzing related websites. The goal is to identify malicious content and assess the current status of these websites. The research addresses specific questions outlined in section 3 and covers various analytical dimensions.

❶ **Cryptojacking infrastructure websites distribution.** This dimension explores the geographic and hosting distribution of websites engaged in cryptojacking. Analyzing the locations and hosting providers allows us to detect trends and hotspots in the infrastructure supporting cryptojacking activities.

❷ **Threat categories and classifications.** Here, we analyze and categorize the threats from cryptojacking websites, focusing on identifying malicious behaviors like malware distribution, phishing, and unauthorized cryptocurrency mining.

❸ **Categorization of cryptojacking websites (current) into as malicious and benign.** We analyze and categorize the current status of websites engaged in cryptojacking activities, distinguishing between those that are actively malicious, potentially harmful, or benign (no longer active or posing a threat).

❹ **Malicious contents in the cryptojacking infrastructures.** We investigate the specific types of malicious content present in cryptojacking websites' infrastructure, analyzing scripts, software, and techniques used for cryptojacking and other malicious activities. Our study offers a comprehensive analysis of cryptojacking infrastructure, providing insights into its distribution, threat landscape, current status, and the nature of malicious content it hosts.

5 Results

5.1 Cryptojacking malware in the malicious dataset

We analyzed the 919 websites included in the malicious file of the MANiC dataset. Our detailed examination revealed eight distinct types of cryptojacking malware within the dataset, as shown in Figure 1. One of the most significant cryptomining services identified in our analysis was CoinHive, which accounted for approximately 74% of the malicious presence within the dataset. CoinHive was initially created as a legitimate service for cryptocurrency mining, enabling website owners to monetize their content by using visitors' CPU resources for mining. However, cybercriminals hijacked CoinHive for cryptojacking purposes, embedding its scripts into websites without users' consent to exploit their computing power for mining. Our findings indicate that even after its discontinuation (since 2019), the legacy of CoinHive continues to influence the prevalence and distribution of cryptojacking activities. This enduring impact is evident in the continued presence and proliferation of similar cryptomining malware. These insights underscore the need for vigilance and robust cybersecurity measures to mitigate the threats posed by such malicious activities.

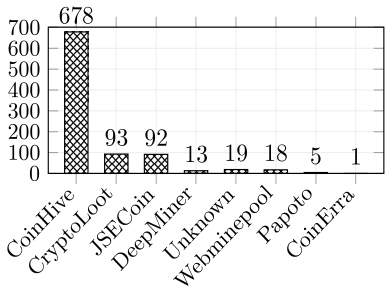


Fig. 1: This figure shows the prevalence of various cryptojacking malware types found within the malicious dataset, with CoinHive being the most significant contributor

5.2 Cryptojacking infrastructure websites distribution

This study extensively analyzes numerous cryptojacking websites hosted across diverse geographic locations. Understanding the correlation between these locations is crucial for two main reasons. Firstly, examining the geographic distribution enhances our understanding of these websites' locations and the specific countries hosting them. This mapping helps identify trends and patterns that highlight regional vulnerabilities and the prevalence of cryptojacking infrastructure. Secondly, the analysis provides insights into how technological advancements and internet penetration contribute to cryptojacking activities. By correlating the presence of cryptojacking websites with regional technological

landscapes, we can determine if higher technological development and internet usage correlate with increased cryptojacking incidents. This investigation also explores whether factors like broadband availability and digital literacy influence the spread of cryptojacking.

Addressing these questions will aid in identifying regions most impacted by cryptojacking. This data is crucial for devising strategies to combat cryptojacking, guiding policy decisions, and allocating resources to mitigate threats in vulnerable areas. Implementing effective discovery practices will enhance vigilance and security measures to safeguard networks against these pervasive threats. Of the 518 websites identified with malicious content, 116 were redacted for undisclosed reasons, limiting the analysis of country distribution to the remaining unredacted sites. Analyzing these sites allows for a focused exploration of geographical trends and patterns associated with malicious online activities.

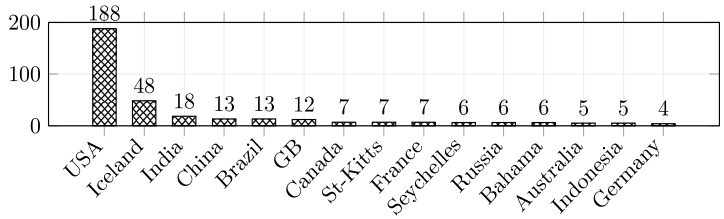


Fig. 2: Cryptojacking websites distribution. A heavy-tailed distribution regarding the number of websites associated with cryptojacking activities.

Observations. Figure 2 displays the top 15 countries with the most websites infected with cryptojacking malware, as per our dataset. The websites are spread out in various countries across the Americas, Asia, South America, and Europe. We observed that the countries are advanced and have high internet penetration. The USA has more than 55% of the websites, followed by Iceland, which has about 15%. The USA is known to be well-advanced technologically and hosts most of the domains. The prevalence of cryptojacking websites in the United States may be attributed to the country's significant number of registered websites. Cryptocurrency has gained significant global recognition in recent years. However, it is notable that most cryptocurrency miners reside in regions with high internet penetration and reliable electricity access. Despite the widespread popularity of cryptocurrency investment and trading, the concentration of mining activities in such regions highlights the importance of favorable infrastructure for this practice. This observation suggests a correlation between the two variables and warrants further investigation.

5.3 Threat categories and classifications

VirusTotal is the primary tool we use to scan malicious websites. After conducting scans, we detected various malicious activities, such as phishing, malware, and spyware identified by different security engines. We focused on these security engines and their reported malicious content to gain deeper insights into

Table 1: Security vendors with the threat types.

Vendor	Threat type
Forcepoint ThreatSeeker	media file, compromised website, proxy avoidance, application and software download, hacking, potentially unwanted, suspicious content, p2p file sharing, and uncategorized.
Dr.Web	adult content, known infection source, gambling.
Webroot	malware, p2p, Bitdefender, porn, proxy avoidance and anonymizers, phishing and other frauds, spyware, and adware.
alphaMountains.ia	suspicious, malicious, unrated, anonymizers, JSEcoin, scam, illegal, unethical, coinhives, piracy, plagiarism.
Sophos	spyware and malware, pua and others, phishing and fraud, suspicious, spam URLs, proxies.
XcitiumverdicCloud	media sharing, spyware, malware.
Bitdefender	proxies, file sharing, webproxy

the nature of the threats. While the direct results from VirusTotal did not explicitly indicate crypto mining-related scripts, further analysis of the referrer files revealed the presence of such scripts. This was done by examining the code and behavior of the files, looking for patterns commonly associated with cryptomining scripts. The scan identified a total of 27 security engines, out of which 7 reported the presence of malicious content. Table 1 summarizes the security engines and the corresponding malicious and suspicious contents they detected. Moreover, Figure 3 contains the list of the top 15 security engines and the number of malicious contents, including malicious, suspicious, and undetected contents.

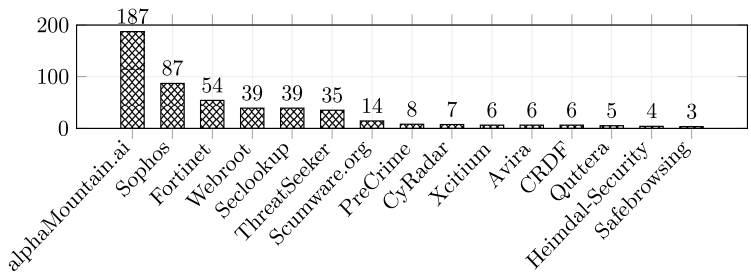


Fig. 3: Security engines categories with the number of occurrences.

Observations. The outcome of a security scan is determined by the security engine's ability to accurately identify and classify a malicious item. Sometimes, it may give false positives. It is important to note that the scan results are based on a specific period and could change over time due to website or security engine updates to correctly identify undetected or misclassified files.

5.4 Malicious contents in the cryptojacking infrastructures

We organize the results from the VirusTotal scan into four distinct categories: harmless, malicious, suspicious, and undetected. Harmless files and contents are

Table 2: The outcomes of security engines' classification.

Category	Harmless	Malicious	Suspicious	Undetected
Count	35,370	948	395	8,394

benign, posing no direct or indirect threat to users or systems. The scan results presented in Table 2 indicate that most detected files fall into the harmless category, suggesting they are safe for use without any security concerns.

On the contrary, malicious file content can be hazardous and potentially harmful. These malicious contents can be injected into infrastructure for various nefarious purposes, including cryptocurrency mining. The potential harm of these contents underscores the importance of security measures and the need for caution. Cryptocurrency mining involves leveraging website resources for mining purposes without the explicit consent of website owners or visitors.

Additionally, malicious files can disrupt websites, temporarily unavailable or hijacking them until a ransom is paid [2]. The potential impact of these files on websites underscores the severity of the threat and the need for robust security measures. Such files may contain malware, spyware, or phishing components to compromise user data or system integrity.

We analyzed 887 websites and found that 58% are malicious and contain either suspicious or malicious files. The remaining 42% are harmless or have undetected files. For further analysis, Table 3 summarizes the malicious websites. We categorize the files into three: benign, suspicious, and malicious. These websites contain benign or harmless files, but the presence of malicious and suspicious files qualifies them as malicious websites.

Table 3: Malicious contents and their distribution.

Security Engines	Benign	Suspicious	Malicious
Forcepoint ThreatSeeker	21	10	4
Dr. Web	0	2	1
Webroot	0	1	5
alphaMountain.ia	7	4	3
Sophos	7	7	5
XcitiumVerdictCloud	1	0	2
BitDefender	12	5	0

Observations. Table 3 data clearly shows that the number of benign and suspicious files significantly exceeds that of malicious files. The table visually represents the distribution of files according to their threat levels. Notably, benign files, which are not considered harmful, comprise most of the files analyzed. These files pose no threat to the system and are safe for use. Following benign files, suspicious files constitute the next largest group. While these files are not confirmed to be harmful, they exhibit behaviors or characteristics that warrant further investigation to determine their true nature and potential threat level. In contrast, the number of malicious files known to be harmful and pose a significant risk to the system is comparatively smaller. The presence of these files can cause various types of damage, including data breaches, system corruption, and unauthorized access. This distribution highlights the dataset's predominance of

non-malicious files while scrutinizing suspicious files to ensure system security and mitigate potential threats.

5.5 Correlation and statistical test

To understand the correlation between malicious, suspicious, and undetected content on the website and conduct statistical analyses, we conducted two specific statistical tests: one between malicious and suspicious contents and another between malicious and undetected contents. The objective of these tests is to verify the existence of any relationship between the contents and their potential effects. Our primary aim is to identify the contributing factors to the presence of malicious activity on the websites. Table 2 and Table 3 show the count of the threat classifications and the website categories. The Pearson correlation coefficient data is based on the frequency of malicious and suspicious content on 517 malicious websites. This dataset enables us to measure and examine the linear relationship between the presence of malicious and suspicious content on these websites. The alpha (α) for the null hypothesis is chosen as 0.05

Malicious and suspicious content in the websites. The Pearson correlation coefficient is a quantitative statistical measure utilized to assess the presence and strength of a linear correlation between malicious and suspicious content. It quantifies the degree to which the two variables are linearly related, providing valuable insights into the extent of their association. From Table 4, The Pearson correlation coefficient indicated a small but significant negative relationship between malicious and suspicious content. The negative coefficient suggests that changes in these two variables occur in opposite directions. The difference between the malicious and suspicious content is substantial enough to be statistically significant. Due to the significance and the high p-value, we rejected the null hypothesis because there is a relationship between the two datasets.

Malicious and undetected content in the websites. We conducted a Pearson correlation analysis between malicious and undetected content using a dataset of 517 websites to explore their potential relationship. From Table 5, the analysis reveals a significantly positive correlation between these variables. This positive correlation suggests that changes in undetected content correspond proportionally with changes in malicious content in the same direction. Despite a relatively small difference, the correlation is statistically significant, supported by a sufficiently low p-value to reject the null hypothesis. This finding indicates that an increase in undetected content is associated with a proportional increase in malicious activities on these websites.

Suspicious and malicious content from the security engines. We investigate the correlation between malicious and suspicious content detected by security engines. The Pearson correlation coefficient from Table 6 indicates a non-significant positive relationship between these variables. This slight positive correlation suggests that changes in one variable have minimal impact on the other, indicating little mutual influence. Due to the small and statistically insignificant difference, we cannot reject the null hypothesis that there is no correlation between suspicious and malicious content as identified by the security engines. Therefore, the small p-value does not provide enough evidence to support a meaningful relationship between these variables.

Table 4: Correlation between malicious and suspicious content.

Parameter	Value
Coefficient (r)	-0.2143
r^2	0.04594
p-value	8.696e-7
Covariance	-0.3048
Sample size (n)	517
Statistic	-4.9799

Table 5: Correlation between malicious and undetected content.

Parameter	Value
Coefficient (r)	0.2335
r^2	0.05452
p-value	7.834e-8
Covariance	1.9366
Sample size (n)	517
Statistic	5.4497

Table 6: Correlation between malicious and suspicious content.

Parameter	Value
Coefficient (r)	0.2694
r^2	0.07257
p-value	0.5591
Covariance	1.8571
Sample size (n)	7
Statistic	0.6255

5.6 Summary of Results and Findings

While some content is flagged as suspicious or undetected, it may contain elements contributing to the websites' malicious nature. Some of the suspicious and undetected files likely harbor hidden malicious content, which suggests that the security scanning software may not have correctly classified them. Collectively, this increases the number of malicious instances on these websites. Therefore, examining and reclassifying suspicious and undetected files is imperative to understand better and mitigate their contribution to malicious activity. Additional findings are illustrated below:

Correlation between website contents. Our analysis reveals a correlation between websites hosting malicious and suspicious content, such as phishing links, malware, or suspicious scripts, and those with malicious and undetected content. This correlation suggests that websites exhibiting such content may harbor malicious elements. Understanding these correlations is crucial for effectively identifying and mitigating cyber threats.

Geographical concentration of cryptojacking websites. Most cryptojacking websites are concentrated in a few countries with high internet penetration. We can attribute this concentration to various factors, including lax cybersecurity regulations, widespread use of outdated software, and a need for more awareness among website owners and users. Addressing this concentration requires coordinated efforts to raise awareness about the risks of cryptojacking.

Redacted domain information. Many domains exhibit redacted or "unavailabl" information, challenging ascertaining ownership and origins. This lack of transparency hinders tracking and addressing malicious activities associated with these domains. Improving transparency and accountability in domain registration processes is necessary to enhance cybersecurity measures.

Observation of malicious contents. We identified several instances of malicious content associated with cryptojacking. These contents, including scripts and malware, exploit vulnerabilities in websites for illicit mining using visitors' computational resources. Detecting and removing such malicious content is critical for safeguarding users' security and privacy [9,22].

Cleanup of previously malicious websites. Many websites previously flagged as malicious have undergone cleanup, removing the malicious content. This cleanup indicates proactive measures website owners or security professionals take to mitigate the impact of cryptojacking activities. However, continuous monitoring is necessary to prevent reoccurrences of such incidents.

Return of previously malicious websites to benign status. Some previously identified as malicious websites have returned to a benign state, indicating a potential temporary hijacking for cryptomining purposes. This phenomenon underscores the dynamic nature of cyber threats, where threats can evolve and change over time, and the importance of timely detection and response mechanisms to thwart cryptojacking attempts effectively.

Absence of correlation between malicious and suspicious contents in security engines. Our analysis indicates no correlation between malicious and suspicious contents in security engines. This finding suggests that while security engines may flag specific contents as suspicious, they may not necessarily classify them as malicious. Understanding this distinction is crucial for refining threat detection algorithms and improving the accuracy of security assessments.

6 Concluding Remarks

We investigated websites compromised for cryptocurrency mining. Using the whois tool, we identified the geographic distribution of these websites, primarily located in regions with significant internet usage. According to our analysis using virustotal.com, out of the 887 websites previously classified as malicious, 370 no longer contain malicious content or crypto-jacking scripts. This improvement may stem from enhanced website security measures or reduced attractiveness to attackers. However, some sites still harbor crypto-jacking-related malware, clandestinely exploiting users' computing resources for unauthorized cryptocurrency mining. Our findings suggest these websites remain susceptible to re-infection. We acknowledge the limitations of our study and emphasize the need for future research to provide deeper insights into cryptojacking dynamics and lifecycle.

References

1. M. Aloqaily and V. Pandit. Ieee conference on blockchain and cryptocurrency(icbc'23). *IEEE Communications Magazine*, 61(11):6–9, 2023.
2. O. Alrawi, M. Ike, M. Pruett, R. P. Kasturi, S. Barua, T. Hirani, B. Hill, and B. Saltaformaggio. Forecasting malware capabilities from cyber attack memory images. In *USENIX Security*, pages 3523–3540, 2021.
3. O. Alrawi, C. Zuo, R. Duan, R. P. Kasturi, Z. Lin, and B. Saltaformaggio. The betrayal at cloud city: An empirical analysis of cloud-based mobile backends. In *USENIX Security*, pages 551–566, 2019.
4. N. Bhatia, S. Bansal, and S. Desai. A detailed review of blockchain and cryptocurrency. *CoRR*, abs/2303.06008, 2023.
5. J. Burgess, P. O'Kane, D. Carlin, and S. Sezer. Manic: Multi-step assessment for crypto-miners. In *International Conference on Cyber Security and Protection of Digital Services*. IEEE.
6. D. Carlin, J. Burgess, P. O'Kane, and S. Sezer. You could be mine(d): The rise of cryptojacking. *IEEE Secur. Priv.*, 18(2):16–22, 2020.
7. D. Carlin, P. O'Kane, S. Sezer, and J. Burgess. Detecting cryptomining using dynamic analysis. In *IEEE PST*, 2018.
8. Y. Du, O. Alrawi, K. Z. Snow, M. Antonakakis, and F. Monrose. Improving security tasks using compiler provenance information recovered at the binary-level. In *CCS*, pages 2695–2709. ACM, 2023.

9. R. Duan, A. Bijlani, Y. Ji, O. Alrawi, Y. Xiong, M. Ike, B. Saltaformaggio, and W. Lee. Automating patching of vulnerable open-source software versions in application binaries. In *NDSS*, 2019.
10. M. Garriga, M. Arias, and A. D. Renzis. Blockchain and cryptocurrency: A comparative framework of the main architectural drivers. *CoRR*, abs/1812.08806, 2018.
11. M. Hajiaghapour-Moghimi, E. Hajipour, K. A. Hosseini, M. Tavakkoli, S. Fattaheian-Dehkordi, M. Vakilian, and M. Lehtonen. Hedging investments of grid-connected PV-BESS in buildings using cryptocurrency mining: A case study in finland. *IEEE Access*, 11:66327–66345, 2023.
12. K. Jayasinghe and G. Poravi. A survey of attack instances of cryptojacking targeting cloud infrastructure. In *ACM APIT*, pages 100–107, 2020.
13. N. Kshetri, M. M. Rahman, S. A. Sayeed, and I. Sultana. cryptoran: A review on cryptojacking and ransomware attacks w.r.t. banking industry - threats, challenges, & problems. *CoRR*, abs/2311.14783, 2023.
14. N. Kshetri and J. M. Voas. Cryptojacking. *Computer*, 55(1):18–19, 2022.
15. N. Lachtar, A. A. Elkhail, A. Bacha, and H. Malik. A cross-stack approach towards defending against cryptojacking. *IEEE Comput. Archit. Lett.*, 19(2):126–129, 2020.
16. A. Menati, Y. Cai, R. E. Helou, C. Tian, and L. Xie. Optimization of cryptocurrency mining demand for ancillary services in electricity markets. In *HICSS*, pages 3052–3061, 2024.
17. A. Mohaisen. Towards automatic and lightweight detection and classification of malicious web contents. In *IEEE HotWeb*, pages 67–72, 2015.
18. A. Mohaisen and O. Alrawi. Unveiling zeus: automated classification of malware samples. In *WWW*, pages 829–832, 2013.
19. A. Mohaisen and O. Alrawi. Av-meter: An evaluation of antivirus scans and labels. In *DIMVA*, volume 8550, pages 112–131. Springer, 2014.
20. A. Mohaisen, O. Alrawi, and M. Mohaisen. AMAL: high-fidelity, behavior-based automated malware analysis and classification. *Comput. Secur.*, 52:251–266, 2015.
21. F. N. Naseem, A. Aris, L. Babun, E. Tekiner, and A. S. Uluagac. MINOS: A lightweight real-time cryptojacking detection system. In *NDSS*, 2021.
22. R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. Iotfinder: Efficient large-scale identification of iot devices via passive DNS traffic analysis. In *IEEE EuroS&P*, pages 474–489. IEEE, 2020.
23. P. Rajba and W. Mazurczyk. Limitations of web cryptojacking detection: A practical evaluation. In *ARES 2022: The 17th International Conference on Availability, Reliability and Security, Vienna,Austria, August 23 - 26, 2022*, pages 52:1–52:6. ACM, 2022.
24. M. Saad, A. Khormali, and A. Mohaisen. End-to-end analysis of in-browser cryptojacking. *CoRR*, abs/1809.02152, 2018.
25. M. Saad and D. Mohaisen. Analyzing in-browser cryptojacking. *IEEE Transactions on Dependable and Secure Computing*, 1, 2024.
26. M. Saad, J. Spaulding, L. Njilla, C. A. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen. Exploring the attack surface of blockchain: A comprehensive survey. *IEEE Commun. Surv. Tutorials*, 22(3):1977–2008, 2020.
27. E. Tekiner, A. Acar, and A. S. Uluagac. A lightweight iot cryptojacking detection mechanism in heterogeneous smart home networks. In *NDSS*, 2022.
28. A. Wang, W. Chang, S. Chen, and A. Mohaisen. Delving into internet ddos attacks by botnets: Characterization and analysis. *IEEE/ACM Trans. Netw.*, 26(6):2843–2855, 2018.
29. R. Xiao, T. Li, S. Ramesh, J. Han, and J. Han. Magtracer: Detecting GPU cryptojacking attacks via magnetic leakage signals. In *ACM MobiCom*, pages 68:1–68:15, 2023.

A Survey on Attack Cases with VBS Malware in Windows

GyuHyun Jeon[1[0009-0000-7204-4406]], Seungho Jeon[1[0000-0002-7116-6062]], and Jung Taek Seo[1*[0000-0003-0971-8548]]

[1] Gachon University, Seongnam-daero 1342, Seongnam-si, Republic of Korea
{pengchan88, shjeon90, seojt}@gachon.ac.kr

Abstract. Visual Basic Script(VBS)-based malicious code exists in various forms such as document type, executable file, and LNK file. After disguising it as a normal file or exploiting a vulnerability to trick the user into executing it, malicious actions are performed when the file is executed. Until recently, malware attacks targeting companies that exploited VBS continued to occur. Therefore, research is needed on effective defense techniques that can detect and respond to VBS malware attacks. Therefore, this survey examined comprehensive information about VBS. Related literature was selected by referring to papers and reports from Google Scholar, Web Science, and security-specialized companies. First, the main characteristics of VBS are explained. Afterward, cases of malicious code attacks targeting Windows systems were presented and analyzed. We also investigated research related to methods for detecting VBS malware. Finally, we describe some defense techniques for detecting malware using VBS. We explain real-time monitoring, blocking, and automation of double-extension file monitoring and deletion based on Python Watchdog. In addition, we analyzed file paths frequently used in VBS attacks and proposed VBS file filtering. In addition, we explained defense methods by detecting VBS malware through event ID and XML-based Windows PowerShell log analysis in Windows Event Viewer and blocking and deleting processes by checking whether network configuration has been changed.

Keywords: Malware, VBS, Windows, Case Analysis, MITRE ATT&CK

1 Introduction

Visual Basic Script(VBS)-based malicious code exists in various forms such as document type, executable file, and LNK file. It disguises itself as a normal file or exploits a vulnerability to induce the user to execute the file and then performs malicious actions when executed[1]. Until recently, malware attacks such as DarkGate, Emotet, and QakBot that exploit VBS and Remote Access Trojan (RAT) series malware attacks have continued to occur[2]. In March 2024, North Korea's hacking group Konni APT[3] executed a malicious VBS targeting South Korean companies using spear phishing and disguised files. In April 2024, the Ande Loader[4] malware was distributed targeting North American manufacturing companies by exploiting

obfuscation, image switching, and PowerShell techniques. Through this, the final payload, NjRAT malware, was executed. In this way, after initial access to the attack surface, such as the target's system and network, VBS malware causes additional damage, such as stealing personal information such as keylogging and downloading malicious payloads through loaders. Additionally, various attack techniques, such as VBS obfuscation, are being developed to avoid malware detection. Therefore, research is needed on effective defense techniques that can detect and respond to VBS malware attacks.

The primary objective of this survey is this survey is to find out the basic knowledge and characteristics of VBS technology. Afterward, we analyze the latest attack cases using Windows VBS malware and derive the attack techniques used. The derived attack technique uses the MITER ATT&CK Framework to perform a formalization process. Finally, we investigate defense techniques against VBS malware attacks. The defense techniques investigated include studies on detection methods for malware using VBS. Related research in this survey referred to papers provided on the Google Scholar website. In this survey, VBS technology and VBS attack cases were referenced from malware reports or projects provided by security-specialized companies.

The structure of this paper is as follows. In Chapter 2, the scope of the survey is explained. Chapter 3 presents the main characteristics of VBS. Chapter 4 provides information about VBS malware attack cases. Chapter 5 examines research related to methods for detecting malware using VBS. Chapter 6 discusses research on countermeasures against VBS malware attacks. Chapter 7 concludes by explaining the conclusions of this survey.

2 Survey Scope and Screening Process

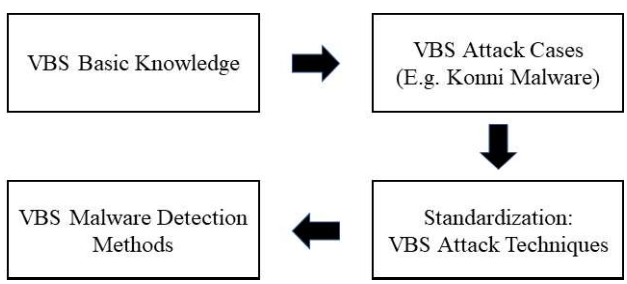


Fig 1. Literature Selection Process.

The literature selection process is shown in [Figure 1]. First, VBS-related technical reports published by security-specialized companies were selected as literature on basic knowledge of VBS technology. In addition, we referred to the MITER ATT&CK Framework [5], a security framework targeting Windows systems that classifies information on various attack techniques. For VBS attack cases, only VBS malware attack cases that occurred between March 2024 and May 2024 were selected to ensure the

novelty of this survey. The attack cases were chosen from analysis reports on malware attack cases using VBS technology based on data released by a security company specializing in malware detection and analysis. Literature related to VBS malware detection methods was provided by Google Scholar and Web Science websites, and papers published between June 2021 and May 2024 were examined. First, papers that mention VBS technology are selected. This is relevant if a brief description of the VBS technology exists in the paper. Second, we choose papers that describe the VBS attack detection method. This paper is about detecting attacks based on the characteristics of VBS technology.

3 Visual Basic Script

VBS is a lightweight active script language based on Visual Basic developed by Microsoft[3]. VBS is basically written in a Function/EndFunction configuration and uses scripting techniques that use Microsoft Component Object Model (COM) objects to access elements of the running environment.

Table 1. Key Attribute of VBS.

Attribute	Description
Purpose	Task automation.
Operation System	Windows (E.g. IE, IIS, WSH)
Function	E.g. Change network configuration, System backup and account management, Document program macro
Reconnaissance	E.g. Disguised as legitimate emails or downloaded files

[Table 1] shows the main features of VBS, including its purpose of use, execution environment, reconnaissance method for accessing the attack surface, and main functions. The VBS function is built into Windows 98 and later versions and can be run mainly in Windows host environments such as Internet Explorer (IE), Internet Information Services (IIS), and Windows Scripting Host(WSH)[6]. It is used to automate repetitive tasks in the system management area such as system backup and account management when changing network configuration in the environment. Recently, it is often used as a macro in applications that support documentation functions, such as HWP, PDF files, and MS Office's Word and Excel[7].

Table 2. Key Vulnerabilities and Obfuscation Techniques of VBS.

Attribute	Techniques
Vulnerabilities	E.g. Allow remote/arbitrary code execution, VBS code injection, Personal information theft
Obfuscation	E.g. Wscript.Shell, Base64, XOR

[Table 2] shows frequently used vulnerabilities and obfuscation techniques to perform attacks that exploit vulnerabilities in VBS main functions. First, in the attack reconnaissance phase, a method of disguising a file containing a malicious VBS as a legitimate email or download file and then distributing it is often used[8]. Afterward, various VBS vulnerabilities are exploited to achieve the attack goal. There are 102 VBS vulnerabilities based on Common Vulnerabilities and Exposures(CVE), a list of publicly known security vulnerabilities[9]. Major vulnerabilities exploited in attacks include allowing remote and arbitrary code execution, VBS code injection, and personal information theft. When writing attack code containing the vulnerability, various obfuscation techniques such as Wscript,Shell, Base64, and XOR encoding bypass attack detection, and multiple obfuscation is possible[10].

4 VBS Malware Attack Cases

Until recently, various malware attacks exploiting VBS have occurred. This session, cases of VBS malware attacks targeting Windows systems will be investigated and explained.

4.1 Konni APT Malware

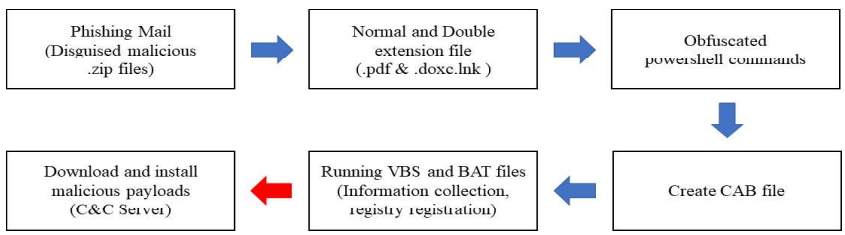


Fig 2. Konni APT Malware Flow Chart.

In March 2024, the 'attachment.zip' malicious file was distributed targeting Korean companies through an attack by an APT group believed to be North Korea's Konni[3]. As shown in [Figure 2], the file consists of a total of two subfiles. It includes LNK files and normal PDF files, each with two extensions. The LNK (shortcut) file is a malicious file that carries out the attack. This exploits a double extension vulnerability and is disguised as a Microsoft Word DOCX document. The malicious file contains obfuscated PowerShell commands, and when executed, it checks the entire length of the shortcut, 0x16EF7F1A. The DOCX file is created with the same name and path as the original, and a Windows CAB file named 'UHCYbG.cab' is created in the public folder path. When the CAB file is unzipped, 'start.vbs' and several BAT files are created, as shown in [Table 3]. As the BAT file specified in the 'start.vbs' script is executed, user information is collected and leaked, additional malicious files are installed, and additional malicious files are installed. It performs malicious tasks such as obtaining persistence through registry registration.

Table 3. start.vbs code.

Row	CMD Commands
1	set obj = GetObject("new:9BA05972-F6A8-11CF-A442-00A0C90A8F39")
2	set itemObj = obj.Item()
3	jHxescZHDcebgaNZ = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\") - 1)
4	itemObj.Document.Application.ShellExecute jHxescZHDcebgaNZ & "\" & "09402649" & ".b" & "at", Null, jHxescZHDcebgaNZ, Null, 0
5	set obj = Nothing

[Table 3] is the 'start.vbs' code that runs a specific file in the file path through the Windows Component Object Model (COM). First, create a new COM object, 'obj' based on the GUID, and then search for items in 'obj.' Calculates the currently executing directory path for search purposes. You can check the full path of the script through 'WScript.ScriptFullName' and find the directory by checking the '\' location. Then, run the '09402649.bat' file in the same location as the script. Finally, we free the 'obj' object.

4.2 Ande Loader Malware

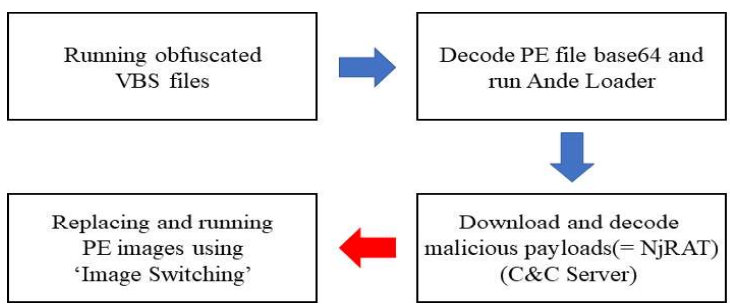


Fig 3. Ande Loader Malware Flow Chart.

In April 2024, Ande Loader malware was distributed targeting North American manufacturing companies [4]. The malware is a VBS containing a Base64-encoded PE file. It is decoded with PowerShell to call a function, pass parameters containing C&C server address information, and directly call the 'VAI' method that can execute the PE file. Afterward, it connects to the C&C server to download and decode the 'NjRAT' malware, which is the final payload encoded in Base64. Image switching technique is used to execute the 'NjRAT' malware. [Table 4] shows the process-related functions used in the image-switching technique.

Table 4. Process-related functions used for image-switching.

Function Name	Description
CreateProcess	Process creation
WriteProcessMemory	Copy (write) data from the specified buffer of the current process to the address range of the specified process.
SetThreadContext	Set the context of the specified thread, After loading the context for the execution address, modify and set it.
ResumeThread	Starting or restarting a thread

In [Table 4], the processes that comprise the .NET framework are randomly selected first. Afterward, use the 'ZwUnmapViewOf-Section' API, which creates a process and unmaps the code of a normal process from memory. A malicious PE image to be replaced is inserted into the released area, the execution address of the thread is set to the Entry Point of the inserted PE image, and the thread is started. After image switching, the 'NjRAT' malware is executed disguised as a previously randomly selected normal process name, collects system keylogging data, stores it in the registry, and uploads it to the C&C server.

4.3 Remcos RAT Malware

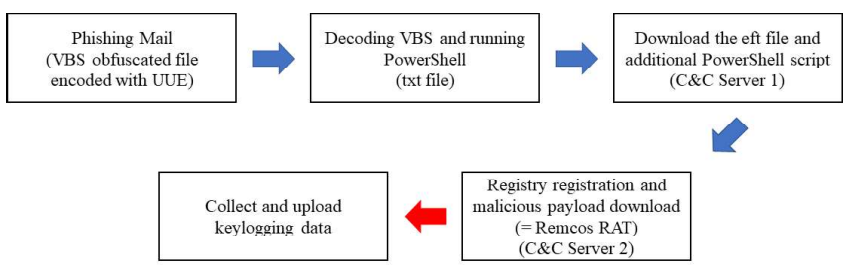


Fig 4. Remcos RAT Malware Flow Chart.

In June 2024, the Remcos RAT malware downloader was distributed through a VBS obfuscated file encoded with Unix-to-Unix Encode (UUE) as a phishing email forged with a quotation, etc. [11]. UUE is a method of encoding binary data in ASCII text format and is believed to have been applied to bypass detection by anti-virus programs. By decoding the UUE file, you can check the obfuscated VBS. The VBS script is executed after saving PowerShell with the file name 'Talehmmedes.txt' in the "C:\Users\Users\AppData\Local\Temp" path. The TXT file contains a 'Haartoppens.Eft' file that performs the PowerShell script execution function in the "C:\Users\User\AppData" path and a C&C server string for downloading additional PowerShell scripts. An additional PowerShell script registers the registry to achieve

persistence and ultimately connects to another C&C server to download and execute the Remcos RAT malware. Remcos RAT malware collects keylogging data from the system, stores it in the "C:\Users\User\AppData" path, and uploads it to the C&C server.

4.4 Standardization of Attack Techniques Used in VBS Malware Attack Cases

As shown in [Table 5], attack techniques using VBS malware used in three cases from 4.1 to 4.3 were derived and standardized by mapping them to the attack techniques of the MITER ATT&CK framework. The MITER ATT&CK framework is a security framework that classifies information about various attack techniques developed by MITER Corporation [6]. Based on actual cyber-attack cases, attackers' actions can be categorized and standardized into various tactics and techniques. As a result of standardizing the derived attack techniques, it was confirmed that the attack techniques used by Konni APT and Remcos RAT malware were similar to each other. In the case of Ande Loader malware, it can be seen that the proportion of process-related attack techniques, such as using process image switching techniques, is high.

Table 5. Standardization of Attack Techniques.

Attack Cases	Description	MITRE ATT&CK Techniques
Konni APT	Spear phishing	T1566(Phishing)
	Execute user's double extension files	T1204(User Execution)
	Running PowerShell and VBS	T1059(Command and Scripting Interpreter)
	PowerShell obfuscation	T1027(Obfuscated Files or Information)
	Securing continuity through registry registration	T1547(Boot or Logon Autostart Execution)
	Collection of user information through keylogging	T1056(Input Capture)
	C2 server communication (downloading malicious payload and uploading keylogging information)	T1071(Application Layer Protocol) & T1041(Exfiltration Over C2 Channel)
Ande Loader	Running PowerShell and VBS	T1059(Command and Scripting Interpreter)
	Base64 obfuscation of 'NjRAT' malware	T1027(Obfuscated Files or Information)
	Use 'ZwUnmapViewOfSection' API (image switching)	T1055(Process Injection)
	Disguises itself as a normal process name and executes malicious code.	T1574.002(DLL Search Order Hijacking)
	Collection of user information through keylogging	T1056(Input Capture)
		T1071(Application Layer Protocol) & T1041(Exfiltration Over C2 Channel)

8

	C2 server communication (downloading malicious payload and uploading keylogging information)	
Remcos RAT	Phishing emails containing forged quotations, etc. User executes forged file Run VBS and PowerShell scripts Obfuscation through UUE encoding Securing continuity through registry registration Collection of user information through keylogging C2 server communication (downloading malicious payload and uploading keylogging information)	T1566(Phishing) T1204(User Execution) T1059(Command and Scripting Interpreter) T1027(Obfuscated Files or Information) T1547(Boot or Logon Autostart Execution) T1056(Input Capture) T1071(Application Layer Protocol) & T1041(Exfiltration Over C2 Channel)

5 VBS Malware Detection Methods

Barr-Smith, Frederick, et al[12] proposed an Exorcist system that detects the insertion of malicious functions into supply chains through differential analysis of binaries to detect VBS attacks used in evasive malware targeting supply chains. The proposed system was evaluated through experiments analyzing 12 APT campaigns (DarkSide, Not-Petya, Ewon, etc.). However, because the Exorcist system only detects PE files in Windows operating systems, it cannot detect attacks on Linux or Mac systems.

TAI, Dang Quy, et al[13] proposed activating the Microsoft Antimalware Scan Interface (AMSI) to detect Visual Basic for Application (VBA) and VBS macro malware running on Microsoft Excel. Because ASMI connects to Office applications through user-level DLLs, it can detect malicious activity caused by macro scripts in the applications.

Xiong, Chunlin, et al[14] defined obfuscation based on the difference in impact on the abstract syntax trees of PowerShell scripts to detect Windows PowerShell attacks using VBS and proposed a new emulation-based recovery technique. They designed a semantic-aware PowerShell attack detection system that utilizes an objective-oriented association algorithm and newly identifies 31 semantic signatures. Experiments on 2342 benign samples and 4141 malicious samples showed that the proposed deobfuscation method took less than 0.5 seconds on average and increased the similarity between the obfuscated script and the original script from 0.5% to 93.2%. Additionally, an average detection rate of 96.7% was confirmed.

Ongun, Talha, et al[15] studied a method of expressing command line text using the word-embedding technique and designed an ensemble boosting classifier to distinguish malicious samples from normal samples based on the embedding expression. An average F1 score of 96% was achieved, but the focus was on classifying malware that used VBS technology rather than on how to defend against VBS attacks.

6 Discussion

This session, we explain ways to protect against VBS malware attacks and describe analysis using real-time monitoring and Windows log analysis.

6.1 Detection Through Real-Time Monitoring

VBS malware files must be detected and automatically stopped or deleted before creation or execution. In the previous attack case, a double extension vulnerability was exploited to induce the user to execute the LNK file directly. Therefore, file detection and deletion for double extension file creation is necessary. [Table 6] shows the environment for implementing and verifying the function and blocking the normal operation of malicious code. I installed the Windows 7 and 10 operating systems in the VM virtual machine, installed the Python package, and wrote the source code.

Table 6. Implementation and Verification Environment

Function Name	Description
Virtual Machine	VMWare Workstation 16 Pro
Operating System	Windows 7, 10
Programming Language	Python
Source Code Editing Tools	Visual Studio Code

Python Watchdog is a file monitoring module that monitors file system events in a specified file path. [Table 7] implements the code that uses Python Watchdog to perform monitoring to detect double extension files.

Table 7. Implementation and Verification Environment

Class	Main Code
Watcher	DIRECTORY_TO_WATCH = " file_path " def __init__(self): self.observer = Observer()

Handler	parts = file_name.split('.')
	if len(parts) > 2:
	try:
	os.remove(file_path)
	print(f"Complete: {file_path}")
	except Exception as e:
	print(f"Faie: {file_path}")

Watchdog is divided into Watcher and Handler, and Watcher specifies the file path to be monitored and creates an Observer object to perform real-time monitoring. In the handler, the detected files were automatically deleted after setting the double extension string detection conditions. After executing the written code and adding a

monitoring process, a Konni APT malware file was created. As a result, double extension files with two or more '.' strings were successfully detected and automatically deleted. Meanwhile, many document-type malware creates malicious VBS files in the %temp%, %AppData%, and %Public% paths. Since these file paths are frequently abused as paths to store malicious files, deletion, and execution must be blocked by setting the path in the Watcher and filtering additional VBS and BAT extension files.

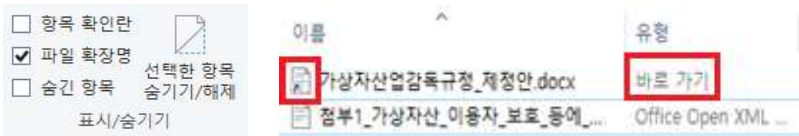


Fig 5. Activating File Extensions and LNK File Types.

Additionally, the user must check the file extension by activating the file extension setting as shown in [Figure 1]. The LNK extension is a shortcut file type and has an arrow image on the file icon as shown in [Figure 2], so be careful when running it.

6.2 Analyze Windows Log

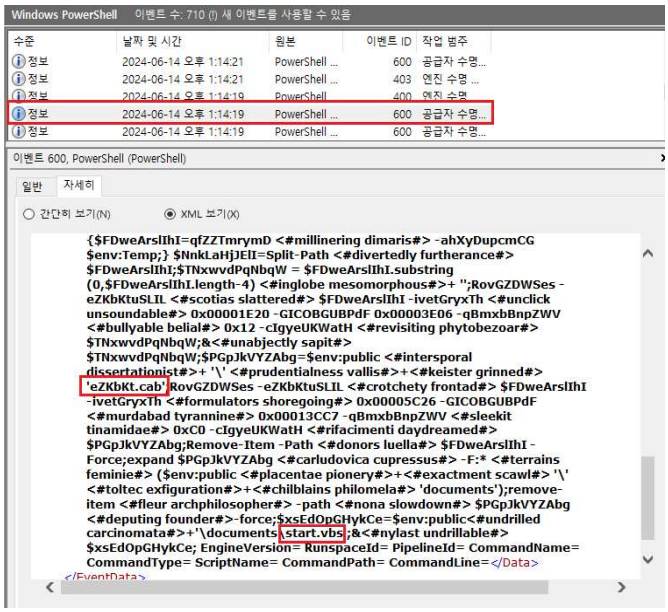


Fig 6. Windows Event Viewer, Konni APT Malware Analysis Results.

In the case of Windows, you must use the Windows Event Viewer (eventvwr.msc) function, which can check event logs that occur in the system[16] to check, audit, and track logs and block any abnormal behavior if it is detected. As shown in [Figure 6], when the object access audit function is executed in the Windows Event Viewer, a security log is saved, and the ID saved is different depending on the event's content. Therefore, when an attack occurs or after an attack, the recorded security log must be checked, and the contents corresponding to each event ID must be analyzed to detect and block the attack. The security log consists of the Event ID, Event Type, Event Category of the event that occurred, and Event Message Text, which outputs the event's details. Since Konni APT malware uses PowerShell, Windows PowerShell events were checked and analyzed. First, event ID 400 indicates PowerShell start, ID 403 indicates PowerShell stop, and ID 600 indicates PowerShell code execution event. As a result of analysis in XML format, PowerShell command argument values that create 'UHCYbG.cab' and 'start.vbs' files were detected when creating malicious code.

In addition, because the attacker uses a C&C server connection to download additional payloads, the network is newly configured, so you must check if the network configuration changes. Generally, blocks must be deleted after monitoring to see if non-executing or unrecognized processes are created.

7 Conclusions

Data such as cyber threat reports indicate that damage caused by VBS malware continues. In this survey, analysis and defense techniques for VBS malware attacks were investigated. Data on the definition and vulnerabilities of VBS were presented. In addition, the latest attack cases regarding VBS malware attack trends are briefly presented and summarized. As for VBS malware detection, research on detecting malware in PE files, macros, and PowerShell containing VBS using data sets was confirmed. In response to VBS malware attacks, real-time monitoring, blocking, and automatic deletion of double-extension files were implemented based on Python Watchdog. In addition, we analyzed file paths frequently used in VBS attacks and proposed VBS file filtering. In addition, we proposed detecting VBS malware through event ID and XML-based Windows PowerShell log analysis in Windows Event Viewer and blocking and deleting processes by checking whether the network configuration has been changed.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00493, 5G Massive Next Generation Cyber Attack Deception Technology Development, 50%) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2024-00354169, Technology Development of Threat Model/XAI-based Network Abnormality Detection, Response and Cyber Threat Prediction, 50%)

12

References

1. KHUSHALI, Vala. A Review on Fileless Malware Analysis Techniques. International Journal of Engineering Research & Technology (IJERT), 9.05, 2020
2. Boannews, VB scripts, which attackers have been using instead of MS Office macros, will soon disappear, https://m.boannews.com/html/detail.html?idx=122647, 2023.10 [last access 2024/06/25]
3. Genians, Hacking damage warning due to rapid rise in Bitcoin price, https://www.genians.co.kr/blog/threat_intelligence/bitcoin, 2024.03 [last access 2024/06/25]
4. INCA Internet, Ande Loader using image switching, https://isarc.tachyonlab.com/5628, 2024.04 [last access 2024/06/25]
5. MITRE, MITRE ATT&CK, https://attack.mitre.org/, [last access 2024/06/25]
6. tutorialspoint, VBScript – Overview, https://www.tutorialspoint.com/vbscript/vbscript_overview.htm, [last access 2024/06/25]
7. RoleCatcher, VBScript: The Complete Skill Guide, https://rolecatcher.com/en/skills/knowledge/information-and-communication-technologies/software-and-applications-development-and-analysis/vbscript/, 2023.12 [last access 2024/06/25]
8. MITRE, Command and Scripting Interpreter: Visual Basic, https://attack.mitre.org/techniques/T1059/005/, [last access 2024/06/25]
9. CVE, https://cve.mitre.org/index.html, [last access 2024/06/25]
10. Koutsokostas, Vasilios, et al., Invoice# 31415 attached: Automated analysis of malicious Microsoft Office documents, Computers & Security, 114, 102582, 2022
11. ASEC, Remcos RAT Distributed as UUEncoding (UUE) File, https://asec.ahnlab.com/en/66463/, 2024.06 [last access 2024/06/25]
12. Barr-Smith, Frederick, et al.: Exorcist: Automated Differential Analysis to Detect Compromises in Closed-Source Software Supply Chains. Proceedings of the 2022 ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses. pp. 51-61. (2022).
13. Tai, D. Q., Gallus, P., & Franti š, P. (2023, May). Macro Malware Development Issuses. In 2023 International Conference on Military Technologies (ICMT) (pp. 1-6). IEEE.
14. Xiong, C., Li, Z., Chen, Y., Zhu, T., Wang, J., Yang, H., & Ruan, W. (2022). Generic, efficient, and effective deobfuscation and semantic-aware attack detection for PowerShell scripts. Frontiers of Information Technology & Electronic Engineering, 23(3), 361-381.
15. Ongun, Talha, et al.: Living-off-the-land command detection using active learning, In: Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses. p. 442-455. (2021)
16. Microsoft, https://learn.microsoft.com/en-us/host-integration-server/core/msdrdaservice-event-logs, [last access 2024/06/25]

Adversarial Manhole: Challenging Monocular Depth Estimation and Semantic Segmentation Models with Patch Attack

Naufal Suryanto[1], Andro Aprila Adiputra[1], Ahmada Yusril Kadiptya[1], Yongsu Kim[1,2], and Howon Kim[1,2]

[1] Pusan National University, Busan, South Korea
[2] SmartM2M, Busan, South Korea

Abstract. Monocular depth estimation (MDE) and semantic segmentation (SS) are crucial for the navigation and environmental interpretation of many autonomous driving systems. However, their vulnerability to practical adversarial attacks is a significant concern. This paper presents a novel adversarial attack using practical patches that mimic manhole covers to deceive MDE and SS models. The goal is to cause these systems to misinterpret scenes, leading to false detections of near obstacles or non-passable objects. We use Depth Planar Mapping to precisely position these patches on road surfaces, enhancing the attack's effectiveness. Our experiments show that these adversarial patches cause a 43% relative error in MDE and achieve a 96% attack success rate in SS. These patches create affected error regions over twice their size in MDE and approximately equal to their size in SS. Our studies also confirm the patch's effectiveness in physical simulations, the adaptability of the patches across different target models, and the effectiveness of our proposed modules, highlighting their practical implications.

Keywords: Adversarial Attack · Adversarial Patch · Monocular Depth Estimation · Semantic Segmentation · Autonomous Driving.

1 Introduction

Self-driving cars rely on complex algorithms and are equipped with sensing devices such as cameras to navigate intricate environments. The core of a camera-based self-driving system utilizes computer vision techniques and deep learning to perceive objects around the car and make decisions. These cars must be constantly aware of their surroundings to ensure a safe driving environment.

Deep learning-based computer vision technologies used in self-driving cars include Monocular Depth Estimation (MDE) and Semantic Segmentation (SS). MDE calculates object distances by extracting depth information from camera

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

images, which is crucial for determining the distance ahead of the car. However, the impact of compromised MDE on autonomous driving tasks remains largely unknown [3]. If MDE provides inaccurate depth information, it can increase the risk of collisions during split-second decisions.

Semantic Segmentation (SS), on the other hand, helps identify traffic objects such as vehicles, pedestrians, and traffic signs, which is vital for safe driving [11]. If SS fails to accurately recognize these objects, it can lead to dangerous driving decisions and result in severe accidents.

Ensuring the robustness of MDE and SS methods is crucial to prevent critical failures. Adversarial attacks, which deceive models into producing incorrect outputs, pose a significant threat [13]. These attacks often use methods like adversarial patches to manipulate predictions [16,3,6,7,9]. However, the practical implementation of such attacks is often overlooked. This work addresses this gap by focusing on realistic patch-based attacks targeting MDE and SS models with patches that resemble road manholes. Manholes, with their diverse designs, are ideal attack vectors. By using manhole-like patches, we highlight the real-world applicability and implications of our approach.

Our contributions can be summarized as follows:

- We present the Adversarial Manhole, a practical adversarial attack targeting MDE and SS in autonomous driving scenarios. To our knowledge, this is the first approach to attack both models simultaneously.
- We propose a Depth Planar Mapping technique that utilizes depth information to accurately position adversarial patches on the road surface, going beyond simple perspective transformations.
- Our framework generates a stealthy adversarial patch designed as a manhole cover, which can be deployed anywhere on the road, exploiting vulnerabilities in autonomous driving systems.
- We conduct an extensive evaluation to assess the robustness of adversarial manholes across various scenarios, including physical simulations, applicability to different models, and ablation studies to analyze the effectiveness of our proposed modules.

2 Related Works

2.1 Monocular Depth Estimation (MDE)

Monocular Depth Estimation (MDE) can be achieved through various methods. Unsupervised methods, addressing labeling costs, treat MDE as a reconstruction problem using dense correspondence between stereo images. Self-supervised learning enhances this by extracting ego-motion, improving accuracy through automated pixel masking and projection error calculation [5]. Another method sharpens depth representation for thin and close objects by enhancing photometric cues [15]. Techniques addressing inconsistencies between viewpoints, pose estimation, and cost volume loss further improve depth estimation accuracy [14].

2.2 Semantic Segmentation (SS)

Semantic Segmentation (SS) assigns semantic labels to each pixel in an image, categorizing them into classes such as vehicles, pedestrians, roads, and buildings.

Deep learning, particularly convolutional neural networks, has significantly advanced image segmentation. PSPNet [19] uses a Pyramid Pooling Module to capture rich contextual information but struggles with real-time performance. ICNet [18] achieves real-time speed (30 FPS) with 70% accuracy using an image cascade network. DDRNet [10] improves upon this with dual-resolution networks, achieving 77.4% mIoU at 102 FPS by using a novel DAPPM module to extract and merge multi-scale context information.

2.3 Patch-based Adversarial Attacks

Research on physical adversarial attacks targeting MDE includes generating noticeable adversarial patches [16], printable stealthy patches with distance limitations [3], and adaptive patches that sacrifice stealthiness for effectiveness [6]. Another approach introduced stealthy patches resilient to transformations but limited to MDE [7]. Our work aims to target both MDE and SS.

Adversarial attacks methods on SS, like the Fast Gradient Sign Method (FGSM) have been explored but is impractical in real-world scenarios as they modify the entire pixel space [1]. Other approaches optimized adversarial patches using cross-entropy loss on pre-trained models, showing effectiveness in digital simulations but reduced performance in real-world applications [9].

Tab. 1 compares existing adversarial patch methods targeting MDE and SS with the proposed Adversarial Manhole. The comparison focuses on stealth capabilities, victim tasks, and patch projection methods. Our approach supports stealth features, pioneering simultaneous attacks in MDE and SS, and introducing Depth Planar Mapping for optimized road patch projection.

Table 1: Comparison of proposed and existing patch-based adversarial attacks.

Method	Stealthy	Victim Task	Patch Projection
Adversarial Patch [16]	No	MDE	Perspective Transformation
SemSegAdvPatch [9]	No	SS	Perspective Transformation
OAP [3]	Yes	MDE	Spatial Transformation
APARATE [6]	No	MDE	Spatial Transformation
SAAM [7]	Yes	MDE	Perspective Transformation
Adversarial Manhole (Ours)	Yes	MDE, SS	Depth Planar Mapping

3 Methodology

3.1 Problem Definition

The primary objective of our proposed method is to create an adversarial manhole that, when placed on the road, causes monocular depth estimation (MDE)

and semantic segmentation (SS) systems to misinterpret the scene. This misinterpretation should lead the vehicle to falsely detect a near obstacle or non-passable object, potentially causing it to stop or choose an alternate path.

Consider a benign input image x. In a patch-based adversarial attack, the goal is to alter the image by introducing an adversarial patch θ, creating an adversarial example x'. To ensure realism, the patch is applied to the target surface using a patch mapping function P. The adversarial example with the patch is formulated as:

$$x' = P(x, \theta, t_o, t_s) = x \odot (1 - m(t_o, t_s)) + M(\theta, t_o, t_s) \odot m(t_o, t_s), \tag{1}$$

where M maps the patch to the desired texture offset t_o and texture scale t_s, m provides a mask indicating where the patch is applied, and $\odot$ denotes element-wise multiplication.

For the MDE model, $D(\cdot)$, the input image x is used to estimate the depth d. To manipulate the depth prediction towards a target depth d_{target} using an adversarial patch θ, we define a loss function L_d. The optimization problem is:

$$\arg\min_{\theta} L_d \left(D(P(x, \theta, t_o, t_s)), d_{\text{target}} \right). \tag{2}$$

For the SS model, $S(\cdot)$, the input image x is used to output the label y for each pixel. We design a loss function L_{ua} for an untargeted attack to predict an incorrect label $y' \neq y$, and L_{ta} for a targeted attack to predict a desired label y_{target}. The optimization problem is:

$$\arg\min_{\theta} \begin{cases} L_{ua} \left(S(P(x, \theta, t_o, t_s)), y \right), & \text{if untargeted} \\ L_{ta} \left(S(P(x, \theta, t_o, t_s)), y_{\text{target}} \right). & \text{if targeted} \end{cases} \tag{3}$$

To simultaneously attack both $D(\cdot)$ and $S(\cdot)$, we combine the losses and apply Expectation over Transformation (EOT) [2] to enhance the robustness of the patch θ over different transformations T and background images X. The combined optimization problem is:

$$\begin{aligned} \arg\min_{\theta} \mathbb{E}_{t \in T, x \in X} \Big[& \alpha \cdot L_d \left(D(P(x, \theta, t_o, t_s)), d_{\text{target}} \right) \\ & + \beta_{ua} \cdot L_{ua} \left(S(P(x, \theta, t_o, t_s)), y \right) \\ & + \beta_{ta} \cdot L_{ta} \left(S(P(x, \theta, t_o, t_s)), y_{\text{target}} \right) \Big], \end{aligned} \tag{4}$$

where α, β_{ua}, and β_{ta} are hyperparameters controlling the influence of each loss.

3.2 Adversarial Manhole

We propose the Adversarial Manhole Framework (Fig. 1) to address these issues. The framework uses differentiable components and losses to optimize the adversarial manhole θ through gradient-based methods.

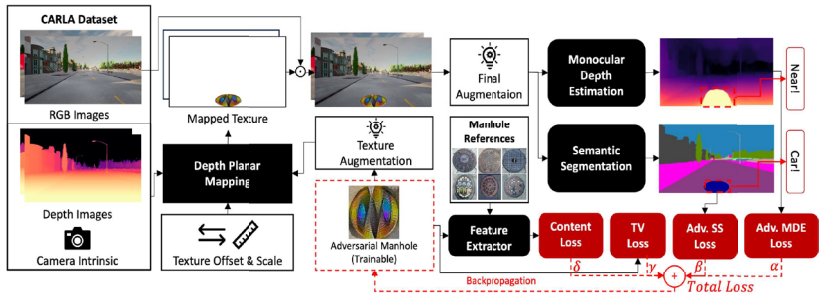


Fig. 1: Our proposed framework for generating adversarial manhole.

Depth Planar Mapping. We propose Depth Planar Mapping to accurately map adversarial textures onto target surfaces, considering texture offset and scale as defined in Eq. 1. Inspired by [12], our approach uses depth information and camera intrinsics from the corresponding RGB image to calculate local surface coordinates and map the texture accordingly. Unlike their tri-planar mapping, we use single-planar (z-plane) mapping due to the upward normal of road surfaces. This method achieves more accurate texture mapping than typical perspective transformations in patch-based attacks. The detailed is described in Alg. 1.

Texture and Final Augmentation. To enhance the robustness of the adversarial manhole, we apply digital augmentations as in the EOT method [2]. These include random brightness and contrast adjustments for various lighting conditions, and random flips and rotations for texture robustness. We formalize our Texture Augmentation (A_t) and Final Augmentation (A_f) as follows:

$$\theta_t = A_t(\theta, t_T), \tag{5}$$

$$x'_t = A_f(x', t_F), \tag{6}$$

where t_T represents texture augmentation variables, and t_F represents final augmentation variables.

Adversarial MDE Loss. One goal of our adversarial manhole is to deceive the MDE model into detecting a near object in front of the vehicle. The MDE model typically outputs a disparity map, which is the inverse of depth: higher disparity values correspond to closer objects. We design the Adv. MDE Loss (L_d) to minimize Eq. 2 by enforcing zero distance predictions in the manhole area (m). The loss function is:

$$L_d(x', m) = -\log(D_{disp}(x')) \odot m. \tag{7}$$

This maximizes the disparity in the manhole area, making the perceived distance close to zero.

Adversarial SS Loss. The adversarial manhole aims to trick the SS model into detecting an obstacle or non-passable object. For untargeted attacks, we

Algorithm 1 Depth Planar Mapping (DPM) Algorithm

Input: Depth image d, Camera intrinsics (c_x, c_y, f_x, f_y), Texture θ, Texture scales t_s, Texture offsets t_o, Texture resolution t_{res}
Output: Mapped texture M and Texture masks m
Calculate surface coordinates SC_x, SC_y, SC_z
for each pixel (u, v) in depth image d **do**
    $SC_x = d(u,v) \cdot \frac{u-c_x}{f_x}$, $SC_y = d(u,v) \cdot \frac{v-c_y}{f_y}$, $SC_z = d(u,v)$
end for
Form surface coordinates tensor (z-upward, x-forward): $SC = \{SC_z, SC_x, SC_y\}$
Adjust and normalize coordinates: $SC_{norm} = ((SC - t_o) \bmod t_s)/t_s$
Compute UV indices: $UV = \max(0, \min(\text{round}(SC_{norm} \cdot t_{res}), t_{res} - 1))$
Map texture to surface using z-plane: $M = \theta[UV_x, UV_y]$
Calculate bounds for masking:
$x_{min} = t_{o_x}$, $x_{max} = t_{o_x} + t_s$, $y_{min} = t_{o_y}$, $y_{max} = t_{o_y} + t_s$
Create texture mask:
$m = (SC_x > x_{min})\&(SC_x < x_{max})\&(SC_y > y_{min})\&(SC_y < y_{max})$
return Mapped texture M and Texture masks m

use L_{ua} to force predictions other than the road class, and for targeted attacks, L_{ta} to achieve a desired class. The losses are:

$$L_{ua}(x', m) = -\log(1 - S_{road}(x')) \odot m \tag{8}$$

$$L_{ta}(x', m) = -\log(\max_{c \in C} S_c(x')) \odot m. \tag{9}$$

These losses adjust the probabilities of the road class and target classes ($c \in C$) in the manhole area.

Total Variation Loss. To ensure the adversarial manhole texture is smooth, we use Total Variation (TV) Loss [8], which minimizes abrupt color changes that are unrealistic in printed and captured images. The TV Loss (L_{tv}) is:

$$L_{tv}(\theta) = \frac{1}{N} \sum_{i,j}^{N} |\theta_{i,j} - \theta_{i,j+1}| + |\theta_{i,j} - \theta_{i+1,j}|, \tag{10}$$

where $\theta_{i,j}$ is the pixel at index (i, j). This loss smooths the texture by minimizing differences between neighboring pixels.

Content Loss. To make the adversarial manhole appear natural and more stealthy, we use Content Loss (L_c) [3], formulated as:

$$L_c(\theta) = \min_{r \in R} |F_l(\theta) - F_l(r)|, \tag{11}$$

where F_l is the CNN feature extractor output at layer l, and R is the set of reference manhole images. This loss minimizes the feature differences between the adversarial and reference images, ensuring visual similarity.

Algorithm 2 Adversarial Manhole Generation

Input: RGB Image X, Depth image D, Camera intrinsics C, Texture resolution t_{res}, Depth Planar Mapping DPM, Texture Augmentation A_t, Final Augmentation A_f, Random function R, Texture scales t_S, Texture offsets t_O, Texture transformations t_T, Final transformations t_F
Output: Adversarial Manhole θ
Initialize θ with random values: $\theta = R(0, 1)$
for each iteration **do**
    Select a minibatch sample of data: $x \in X$, $d \in D$, $c \in C$
    Initialize random transformations: $t_s \in t_S$, $t_o \in t_O$, $t_t \in t_T$, $t_f \in t_F$
    Augment the manhole texture with A_t: $\theta_t = A_t(\theta, t_t)$ ▷ Eq. 5
    Output the mapped texture M and texture mask m using DPM:
$M, m = DPM(d, c, \theta_t, t_s, t_o, t_{res})$ ▷ Alg. 1
    Combine the RGB image and mapped texture: $x' = x \odot (1 - m) + M \odot m$
    Augment the final image: $x'_t = A_f(x', t_f)$ ▷ Eq. 6
    Calculate losses: $L_d(x'_t, m)$, $L_{ua}(x'_t, m)$, $L_{ta}(x'_t, m)$, $L_{tv}(\theta)$, $L_c(\theta)$ ▷ Eq.7 - 11
    Update θ to minimize the total loss L_{total} via backpropagation ▷ Eq. 12
end for
return Adversarial Manhole θ

Adversarial Manhole Generation. We initialize θ with random values and generate an adversarial manhole using our framework. We iteratively update the texture to minimize the final loss (L_{total}):

$$L_{total}(x', m, \theta) = \alpha \cdot L_d(x', m) + \beta_{ua} \cdot L_{ua}(x', m) + \beta_{ta} \cdot L_{ta}(x', m) + \gamma \cdot L_{tv}(\theta) + \delta \cdot L_c(\theta), \tag{12}$$

where α, β_{ua}, β_{ta}, γ, and δ are hyperparameters. The detailed is in Alg. 2.

4 Experiments

4.1 Implementation Details

Datasets. We generate autonomous driving scene datasets using CARLA [4] including RGBs, depths, and camera intrinsics as described in Fig. 1. We randomly selected positions from seven towns and captured the datasets resulting in 2,656 images. Each data has 1024×780 resolution from a 1-meter height camera. We split the dataset into train/val/test with a 60/20/20 ratio.

Frameworks. We implemented our framework using PyTorch[1], with MonoDepth2 [5] as the target MDE model and DDRNet [10] as the SS model. For adversarial manhole generation, we use the Adam optimizer with a learning rate of 0.01, batch size of 8, and 25 epochs. We set $T_{O_x} \in [0.0, 0.4]$ to randomly map the texture between 1.8 - 2.6m from the camera, and $T_{O_y} \in [-0.4, 0.4]$ to randomly map the texture between -0.8m and 0.8m from the center of the camera. For t_F and t_T, we use 0.2 for random brightness and 0.1 for random contrast.

[1] Code and datasets: `https://github.com/naufalso/adversarial-manhole`.

The total loss hyperparameters are $\alpha = 2.0$, $\beta_{ua} = 0.5$, $\beta_{ta} = 0.5$, $\gamma = 1.0$, and $\delta = 0.5$. For targeted attacks, we set the target label to non-walkable categories, such as buildings, walls, fences, poles, persons, and vehicles.

Evaluation Metrics. To evaluate patch performance against the MDE model, we introduce the relative error distance ($Rel.\ Ed$) and region-affected error distance (Ra_{Ed}) metrics:

$$Rel.\ Ed = \frac{\sum(|d - d'|/d) \odot m_{patch}}{\sum m_{patch}}, \tag{13}$$

$$Ra_{Ed} = \frac{\sum B((|d - d'|/d) > Rel.\ Ed_{thres}) \odot m_{road}}{\sum m_{patch}}. \tag{14}$$

Here, d and d' are the distance predictions for clean and adversarial images, m_{patch} is the patch mask, m_{road} is the road mask, $B()$ returns 1 if the condition is met, and $Rel.\ Ed_{thres}$ is the threshold for the affected area set to 0.25. $Rel.\ Ed$ measures the mean error caused by the adversarial patch, while Ra_{Ed} gauges the ratio of affected road regions relative to the mask area. These metrics refine E_d and R_a from prior MDE patch attacks [3,6,7].

To evaluate patch performance against the SS model, we use the Attack Success Rate (ASR) for untargeted (ASR_{ua}) and targeted (ASR_{ta}) attacks:

$$ASR = \begin{cases} ASR_{ua} = \frac{\sum B(s' \neq s_{road}) \odot m_{patch} \odot s_{road}}{\sum m_{patch} \odot s_{road}}, & \text{if untargeted} \\ ASR_{ta} = \frac{\sum B(s' = c \in C) \odot m_{patch} \odot s_{road}}{\sum m_{patch} \odot s_{road}}, & \text{if targeted} \end{cases} \tag{15}$$

Here, s' is the semantic prediction of the adversarial image, s_{road} is the road prediction of the original image, and c represents target labels for targeted attacks. Multiplying m_{patch} by s_{road} ensures evaluation only on valid road predictions within patch areas.

Finally, we adapt the Ra metric for the SS model to measure the ratio of affected error regions in the road area relative to the mask area:

$$Ra_{ss} = \begin{cases} Ra_{ua} = \frac{\sum B(s' \neq s_{road}) \odot s_{road}}{\sum m_{patch}}, & \text{if untargeted} \\ Ra_{ta} = \frac{\sum B(s' = c \in C) \odot s_{road}}{\sum m_{patch}}. & \text{if targeted} \end{cases} \tag{16}$$

Unlike ASR, Ra also considers the error outside the patch area.

4.2 Experiment Results

Patch Effectiveness. We evaluate our generated adversarial manhole on test sets, comparing it to baselines like naive, artistic, and random pattern manholes. We also compare it to related work, such as Adversarial Patch [16] and SemSegAdvPatch [9], using their official code but with our settings. As shown in Tab. 2 and Fig. 2, the models are robust against the baselines, correctly predicting depth and segmentation as normal roads. While Adversarial Patch and

Table 2: Evaluation results of patches effectiveness in our test sets.

Evaluated Patches	MonoDepth2 [5] (MDE)		DDRNet [10] (SS)			
	Rel. Ed	Ra_{Ed}	ASR_{ua}	ASR_{ta}	Ra_{ua}	Ra_{ta}
Naïve Manhole	0.03	0.02	0.00	0.00	0.00	0.00
Artistic Manhole	0.04	0.04	0.00	0.00	0.00	0.00
Random Patch	0.03	0.01	0.00	0.00	0.00	0.00
Adversarial Patch [16]	0.35	1.62	0.00	0.00	0.00	0.00
Ours MDE (L_d+L_{tv}+L_c)	**0.43**	**2.39**	0.00	0.00	0.00	0.00
SemSegAdvPatch [9]	0.06	0.04	0.40	0.04	0.95	0.09
Ours SS (L_{ua}+L_{ta}+L_{tv}+L_c)	0.05	0.05	**0.96**	0.95	**1.08**	0.96
Ours All (L_{total})	**0.43**	2.34	**0.96**	**0.96**	1.02	**0.98**

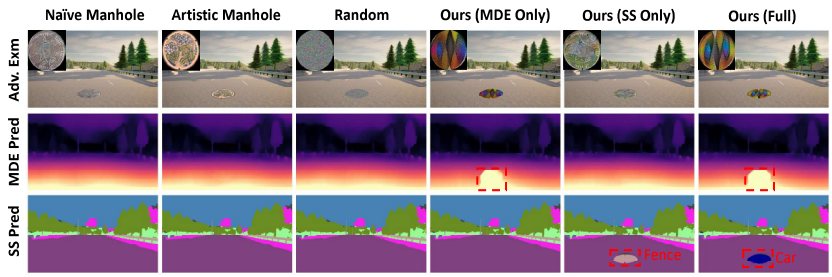
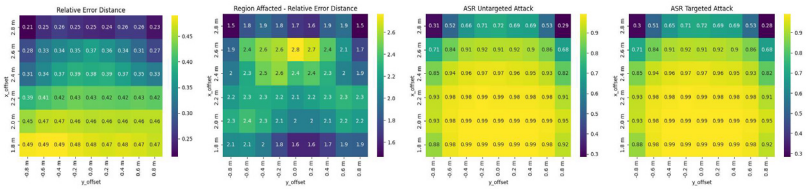


Fig. 2: Qualitative results of our patch effectiveness: The results illustrate that naive, artistic, and random manholes do not affect the model predictions, while our methods successfully fool the models. Zoom for details.

SemSegAdvPatch disturb MDE and SS models, their performance declines due to uncovered manhole orientation during optimization. Note that SemSegAdvPatch supports only untargeted attacks, and we used the EOT variant since the manhole location isn't fixed. Our adversarial manholes, however, can effectively fool the models based on the applied adversarial loss. If L_d is applied, the adversarial manhole causes an average relative error distance of 43% to the MDE model. When L_{ua} and L_{ta} are applied, it achieves a 96% attack success rate in fooling segmentation. Combining all adversarial losses, the manhole effectively attacks both MDE and SS with similar impact. Notably, the adversarial manhole can cause an error to MDE larger than the patch size, affecting a region 239% relative to the patch size, despite the loss focusing only on the patch area.

Patch Robustness. To evaluate the robustness of our adversarial manhole against different placements on the road surface, we conducted experiments across various offsets. As summarized in Fig. 3, our patch remains robust regardless of location but performs best when centrally located and near the camera. For MDE metrics, the relative error distance decreases linearly as the patch is placed further away, from 48% relative error to 25%. However, the region-affected error for MDE shows optimal performance when the patch is 2.8m from the camera's center. We observe this because the affected error region extends



(a) *Rel. Ed* (b) Ra_{Ed} (c) ASR_{ua} (d) ASR_{ta}

Fig. 3: Robustness evaluation against different placements. Zoom for details.

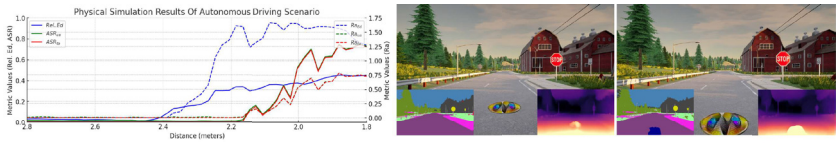


Fig. 4: Physical simulation of approaching car in CARLA. Zoom for details.

from the patch region to the bottom of the image. In contrast, the patch demonstrates a more robust performance on the SS model regardless of position, with an average attack success rate of 98% except for edge placements.

Physical Simulation. We perform physical evaluation by applying the adversarial patch as a manhole cover in the CARLA simulator. We capture video of an approaching car to mimic an autonomous driving scenario. As illustrated in Fig. 4, the adversarial manhole begins affecting the MDE model from a distance of 2.4 meters and the SS model from 2.2 meters. The attack's effectiveness increases as the camera approaches the manhole, reaching a peak performance of 45% relative error distance and 166% affected region for MDE, and a 75% attack success rate with a 78% affected region for SS. These results highlight the substantial impact of the adversarial patch in a more realistic setting.

Applicability to Different Models. We regenerated adversarial manholes with different target models to assess their applicability against various MDE and SS models. As summarized in Table 3, our patch has the highest overall impact

Table 3: Evaluation results of targeting different models.

Target MDE Model	Target SS Model	MDE Metrics		SS Metrics			
		Rel. Ed	Ra_{Ed}	ASR_{ua}	ASR_{ta}	Ra_{ua}	Ra_{ta}
MonoDepth2 [5]	DDRNet [10]	**0.43**	**2.34**	0.96	0.96	1.02	0.98
DepthHint [15]	DDRNet [10]	0.33	0.83	0.97	0.97	1.01	0.98
ManyDepth [14]	DDRNet [10]	0.28	1.51	0.96	0.96	1.02	0.97
MonoDepth2 [5]	PSPNet [19]	0.40	2.03	0.94	0.93	1.00	0.95
DepthHint [15]	PSPNet [19]	0.24	0.53	0.94	0.94	0.98	0.95
ManyDepth [14]	PSPNet [19]	0.25	1.32	0.94	0.93	1.00	0.95
MonoDepth2 [5]	ICNet [18]	0.42	2.32	**0.99**	**0.99**	2.65	2.59
DepthHint [15]	ICNet [18]	0.31	0.76	**0.99**	**0.99**	2.98	2.92
ManyDepth [14]	ICNet [18]	0.29	1.67	**0.99**	**0.99**	**3.13**	**3.09**

Table 4: Ablation studies with the full baseline under random distortion during evaluation. Values in parentheses indicate the difference from the full baseline.

Modules	MonoDepth2 [5] (MDE)		DDRNet [10] (SS)			
	Rel. Ed	Ra_{Ed}	ASR_{ua}	ASR_{ta}	Ra_{ua}	Ra_{ta}
full	0.42	2.18	0.97	0.96	1.01	0.98
fixed t_O	0.19 (-0.24)	0.51 (-1.68)	0.27 (-0.69)	0.26 (-0.71)	0.24 (-0.78)	0.20 (-0.78)
*no DPM	0.25 (-0.17)	0.93 (-1.25)	0.01 (-0.95)	0.00 (-0.96)	0.02 (-0.99)	0.00 (-0.98)
no A_t	0.26 (-0.16)	1.12 (-1.06)	0.41 (-0.55)	0.40 (-0.56)	0.45 (-0.56)	0.42 (-0.56)
no A_f	0.42 (0.00)	2.13 (-0.05)	0.90 (-0.07)	0.90 (-0.07)	0.96 (-0.05)	0.93 (-0.05)
no L_{tv}	0.42 (0.00)	2.13 (-0.06)	0.91 (-0.05)	0.91 (-0.05)	0.94 (-0.07)	0.91 (-0.07)
no L_c	**0.49 (0.06)**	**2.72 (0.53)**	**0.98 (0.02)**	**0.98 (0.02)**	**1.07 (0.06)**	**1.02 (0.04)**

when targeting MonoDepth2 with DDRNet for MDE metrics (43% relative error distance with 2× affected regions) and ManyDepth with ICNet for SS metrics (99% attack success rate with 3× affected regions). However, the lowest overall impact is observed when targeting DepthHint and PSPNet (24% relative error distance with 53% affected regions for MDE and 94% attack success rate with 95% affected regions for SS), indicating that these models are more robust.

Ablation Studies. We performed ablation studies by removing each proposed module, regenerating the patch, and applying random brightness and contrast adjustments to test robustness. As shown in Table 4, removing any module decreased performance, except for content loss (L_c), which trades off stealthiness and effectiveness. Optimizing the patch at a fixed offset (t_O) reduced robustness against varied placements. Replacing DPM with random spatial and perspective transformations (*no DPM) from previous works [16,9,3,6,7] significantly impaired performance in the manhole setting. These findings emphasize the need to cover expected transformations during optimization, as noted in EOT [2].

5 Conclusion

We presented a novel adversarial attack using practical patches that mimic manhole covers to deceive MDE and SS models in autonomous driving systems. Our patches achieved a 43% relative error in MDE and a 96% attack success rate in SS. Physical simulations validated the effectiveness of our adversarial patches in an autonomous driving scenario. Ablation studies confirmed the critical role of our proposed modules and highlighted the applicability of patches in different models. Our work underscores the vulnerability and emphasizes the need for robust defense mechanisms, such as certified defense against patch attacks [17].

References

1. Arnab, A., Miksik, O., Torr, P.H.: On the Robustness of Semantic Segmentation Models to Adversarial Attacks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 888–897. IEEE, Salt Lake City, UT (2018)
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: International conference on machine learning. pp. 284–293. PMLR (2018)

3. Cheng, Z., Liang, J., Choi, H., Tao, G., Cao, Z., Liu, D., Zhang, X.: Physical Attack on Monocular Depth Estimation with Optimal Adversarial Patches (Jul 2022), arXiv:2207.04718
4. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017)
5. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction (2019)
6. Guesmi, A., Hanif, M.A., Alouani, I., Shafique, M.: APARATE: Adaptive Adversarial Patch for CNN-based Monocular Depth Estimation for Autonomous Navigation (Nov 2023), arXiv:2303.01351
7. Guesmi, A., Hanif, M.A., Ouni, B., Shafique, M.: SAAM: Stealthy Adversarial Attack on Monocular Depth Estimation. IEEE Access **12**, 13571–13585 (2024)
8. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5188–5196 (2015)
9. Nesti, F., Rossolini, G., Nair, S., Biondi, A., Buttazzo, G.: Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2826–2835. IEEE, Waikoloa, HI, USA (2022)
10. Pan, H., Hong, Y., Sun, W., Jia, Y.: Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. IEEE Transactions on Intelligent Transportation Systems (2022)
11. Schwonberg, M., Niemeijer, J., Termöhlen, J.A., schäfer, J.P., Schmidt, N.M., Gottschalk, H., Fingscheidt, T.: Survey on unsupervised domain adaptation for semantic segmentation for visual perception in automated driving. IEEE Access **11**, 54296–54336 (2023)
12. Suryanto, N., Kim, Y., Larasati, H.T., Kang, H., Le, T.T.H., Hong, Y., Yang, H., Oh, S.Y., Kim, H.: Active: Towards highly transferable 3d physical camouflage for universal and robust vehicle evasion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4305–4314 (October 2023)
13. Wang, D., Yao, W., Jiang, T., Tang, G., Chen, X.: A survey on physical adversarial attack in computer vision (2023)
14. Watson, J., Aodha, O.M., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth (2021)
15. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints (2019)
16. Yamanaka, K., Matsumoto, R., Takahashi, K., Fujii, T.: Adversarial Patch Attacks on Monocular Depth Estimation Networks. IEEE Access **8**, 179094–179104 (2020)
17. Yatsura, M., Sakmann, K., Hua, N.G., Hein, M., Metzen, J.H.: Certified defences against adversarial patch attacks on semantic segmentation. In: The Eleventh International Conference on Learning Representations (2023)
18. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: ICNet for Real-Time Semantic Segmentation on High-Resolution Images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018, vol. 11207, pp. 418–434. Springer International Publishing, Cham (2018)
19. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid Scene Parsing Network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6230–6239. IEEE, Honolulu, HI (2017)

Plotting OSS-based Supply Chain attack strategies and the defense failure

Arpita Dinesh Sarang[1][0000−0003−4461−711X], Sang-Hoon Choi[2][0000−0002−9549−0887], and Ki-Woong Park[3][0000−0002−3377−223X] ⋆

[1] SysCore Lab, Department of Information Security, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea
arpita.sarang@sju.ac.kr
[2] SysCore Lab, Sejong University, Seoul 05006, South Korea
csh0052@gmail.com
[3] Department of Information Security, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea
woongbak@sejong.ac.kr

Abstract. The supply chain attack, which targets open-source software, is currently the most discussed cyberattack. This is due to the recent open-source XZ utils and PyPI projects based attacks where the attackers employed similar strategies. A backdoor was injected in the libraries developed by these projects for future exploitation when software users installed them on systems. These attacks were executed by the trusted developers of these projects and influenced by unknown promoters. Therefore, there is a need to scrutinize Open Source Software(OSS) Security to protect the legacy of Open Software development for the future. This paper provides an overview of these two attack strategies through their case studies, which aid in the creation of a generic attack framework for supply chain attacks on OSS. We present the existing detection methods for OSS security in this work and their lacunas taking into account at various stages of a supply chain attack. This makes it necessary to build platforms that allow the current OSS security detection modules to be utilized in sequence. Therefore, we introduce an OSS security Detection platform that does not completely address the limitations of the detection technique, but when combined, they can maximize the effectiveness of an OSS security quality check.

Keywords: Supply Chain Attack · Linux distribution · Cyber Security · Supply Chain Attack as a Service(SCAaaS) · Open Source Software(OSS) Security · Software Bill of Materials(SBOM)

1 Introduction

A software-based supply chain attack is a cyber-attack that distributes software or its library with a backdoor that can exploit the critical data on systems they

⋆ Corresponding author.

get installed on [1]. These backdoors are injected during the development process of the software or its library. They are induced by the developers with malicious intent so strategically that they remain undetected even after distribution. This cyber-attack has been in action since the first "Ken Thompson Hack" in 1984, which warned about the upcoming attacks through the great speech on " Reflections on Trusting Trust" [2]. Such attacks have been planned meticulously for many years and malicious code for them is induced periodically to the software. A global report highlighted that a total of 210,031 software packages were affected by supply chain attacks over the past two years [3]. The number of supply chain incidents on open-source software surged to 245,032 in 2023—a 178% increase from 2022 [4].

Open Source Software(OSS) is the potential target for the software-based supply chain attack. These software are widely consumed by the technical community due to the benefits they provide such as free license, customizability, availability, and so on. Also, the software developers take great interest in developing these software projects by contributing valuable suggestions through comments and code fixes on the software project development open discussions and support emails. The core project developers and maintainers consider the suggestions and notice the valuable contributions to OSS development. In this development process the core developer, maintainers or the developers outside the project comment or promote fixes cannot be trusted as they may have malicious intent. A little ignorance in evaluating development due to trust in these entities can lead to a Supply Chain Attack through OSS. An enhanced taxonomy for Open source software-based supply chain attacks (OSSCAs) that listed some affected vectors, include users, systems, project maintainers, and root nodes [5]. A multi-year social engineering initiative, the XZ Util attack on February 24, 2024 [6], and PyPI attack for extraction of the web browser and cryptocurrency data on March 29, 2024 were well-known OSS security breaches for the first half of the year. Due to the prolonged and organised nature of attacks, they are challenging to detect. Attackers typically introduce backdoors by embedding granular program blocks. By the time exploitation is discovered, it is often too late for the victims.

In this paper, we discuss the two famous OSS-based Supply Chain attacks i.e., the XZ Util and the PyPI Project attack with relevant case studies. These recent attacks on the OSS supply chain allowed us to study their attack strategies closely and draw a generic attack framework for OSS-based Supply Chain Attacks. With the advantage of this attack framework built based on the current attack strategies, we were motivated to study the existing OSS security mechanisms for the detection and their lacunas for such attacks. This assists in understanding why these existing defense mechanisms failed and where is the necessity for upgrade them.

The remainder of this paper is as follows: Section 2, provide insight based on previous detection techniques for OSS based Supply Chain Attack. Section 3, describe the OSS-based Supply Chain Attacks in detailed case studies. Section 4, conveys the Generic framework based recent attacks. Section 5, discusses

the prevention and detection strategies and their lackings. Finally, Section 6, concludes our Learnings and future works.

2 Related Work

OSS projects are the most targeted entity for the software-based supply chain attack. Developers are welcomed to the software development environment of these projects based on their development skills, not their good or bad intentions. This allows the malware developer to participate in the development process easily. This is a cyber-security challenge that OSS development projects began to face due to rising supply chain attacks. This victimizes all the OSS users trusting and utilizing the software. When it comes to OSS the well-known ones are downloaded from trusted sites as they are free and can be more customised to use. For studies conducted on OSS security. Detection of supply chain attacks on such software needs to be reviewed thoroughly.

Table 1 summarizes the types of detection techniques introduced by the previous study. These studies have limitations and can fail in cases mentioned in section 3. The Detection techniques are based on various aspects of the OSS code as discussed. The Query-based detection type performs a scan on OSS code for malicious queries and performs a match malicious query database [16]. In [17] and [18], the library-based detection type scan the OSS project dependency libraries if they are outdated or malicious. Whereas, in [19], [20], [21] and [22]

Table 1: OSS Supply Chain Attack Detection Techniques

Related Work	**Technique Type**	**Limitations**
[16]	Query-based detection	It is dependent on source data integrity, normal queries with malicious arguments will be ignored, cannot detect the behavior of queries outside database.
[17], [18]	Library-based detection	Some resource might ignore the less vulnerable packages.
[19], [20], [21], [22]	Vulnerability-based detection	It can prove a package to be less vulnerable due to low vulnerability detected when compared with datasets available even though it is critically malicious.
[23]	Combination of different detection modules	Combining different modules for detection makes this technique type accurate can differ, third party contracts for different applications are risky due to delayed updates.
[24], [25]	Third party based detection	It depends on application detection capability and compatibility with OSS not causing delays.
[26], [27], [28]	Stakeholder-based detection	It is a quality analysis therefore, cannot be measured and trusted.
[29]	Network-based detection	This detection fails detect malicious packets when network traffic use encryption techniques, or the malicious traffic trigger not passed.

the Vulnerability-based detection type detects and matches with the Vulnerability database for flaws in OSS. For, instance SBOM is majorly consumed by vulnerability assessment tools. These three types of detection techniques depend majorly on data resource integrity. In [23], combination of different detection module types is to combine different detection tools available for attack-based malware detection. Third-party-based detection in [24], [25] is implementing the monitoring, protocol-based authentication techniques for security of OSS project development. While the Stakeholder-based detection in [28] is trying the detect the developer's intentions in the project. The Network-based detection in [29] of malicious packages or activity during network communication with other resources. Due to unpredictable patterns and the non-generalized design of Supply chain attacks, the detection techniques are not well defined. They depend on previous attack analysis outcomes and fail to detect new attacks that are polymorphic in nature and should be detected before distribution of the OSS stage.

3 Case studies

We conducted these case studies based on various analysis and resources to outline their attack strategies.

3.1 XZ Utils Project attack

XZ Utils Project(Backdoor target): In 2005, Lasse Collin launched Project LZMA Utils with a small team of engineers. The goal of the project was to create a library for data compression that would support the Linux operating system. This library uses the Lempel–Ziv–Markov chain algorithm (LZMA) for compression and decompression. The xz and lzma file formats can be compressed and decompressed using it. Other software also utilised this library to compress data. Later on, the project became known as XZ utils. Many developers and programming experts contributed to this project, since it's open source, by offering their insightful comments and useful code snippets. Lasse Collins had the authority to approve and merge the code blocks to the project since he was its primary developer and maintainer. Under Lasse Collin's guidance, several iterations of the XZ Utils were introduced throughout the year. These included the compromised versions 5.6.1, which was issued on March 9, 2024, and 5.6.0, which was released on February 24, 2024 [8]. The malicious file was present in these version's release tarballs. Through this version's backdoor, an attacker can command the system of the victim with administrator rights. Version 5.4.6 is presently the most stable version of this project that is available.

Malicious Actors: For years, a number of players worked their part to create this backdoor in the data compression library of the Linux distribution. Some made significant contributions by changing code blocks. While others applied

pressure to have the backdoor code added. In this analysis, we attempt to arrange the attack's chronology in order to obtain the event sequence for the supply chain attack carried out on the Linux distribution's XZ Utils project. Major elements of this attack are contributed by the attack contributors mentioned above. We draw a timeline for the attack events(see Fig. 1). Jia Tan(with developer

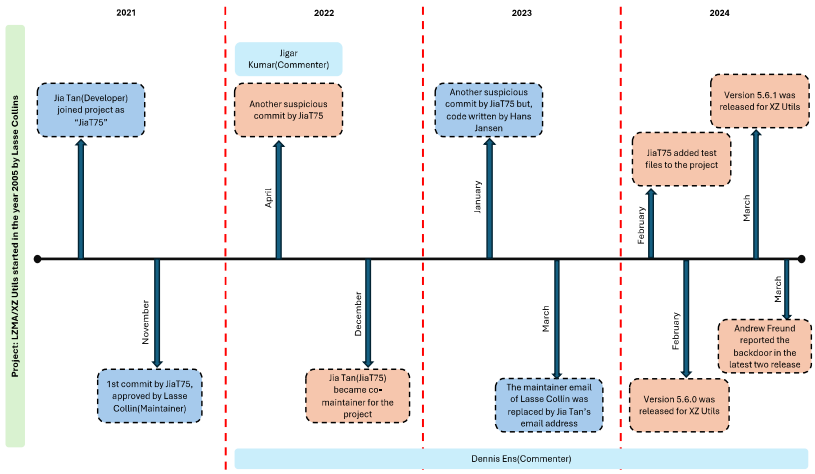


Fig. 1: Attack activity timeline for XZ Utils project

ID JiaT75) joined the XZ Utils project in 2021 and had a significant role in creating this backdoor for supply chain attacks in the XZ Utils library [9]. He provided Lasse Collin with various code blocks for XZ Util development that contained ignorant or obfuscated malicious code that initially was approved and integrated into the main project. Everything maintained on GitHub, including the "xz.tukaani.org" subdomain, was accessible to him. Lasse Collin was the only one with access to the main "tukaani.org" page [10]. JiaTan contributed a number of code blocks to the error-fixing and update of the XZ util project. In November 2021, JiaT75's first commit to xz utils was accepted and merged by Lasse Collin. There was a suspicious code edit in this commit that altered the secure "safe_fprintf()" to the "fprintf()" that code_1 indicated. This was done by employing an insecure function to perform a format string attack. An attacker may run arbitrary programs on the server, read variables off the stack, and cause segmentation faults or application crashes if they supply a format string made up of "fprintf()" conversion characters like %f, %p, %n, etc. as parameter values.

Later, JiaT75 contributed a second suspicious commit to XZ Utils in code_2 in April 2022. The code that ignored the memory and CPU use indication was hardware-related. This was done with the goal of not notifying XZ Util's consumers about any CPU or memory-related issues. By December 2022, Lasse Collin designated JiaT75 as a reliable co-maintainer for the XZ util project. JiaT75 committed a third controversial code commit on January 7, 2023. The code block that was merged was originally composed by Hans Jansen, another

developer, rather than JiaT75. This commit carried out an IFUNC()-based exploit using glibc. In order to safeguard the offset tables and procedure linkage tables, the glibc feature known as IFUNC (GNU indirect function) acts as a resolver, helping to replace the function addresses to default at startup. Since these tables can be modified at startup time, the function from the external library was used to replace the pointer to the RSA_Public_Decrypt. JiaT75 gained greater authority when the official email address for XZ Utils code for component testing infrastructure was updated from "lasse.collin@tukaani.org" to "jiat0218@gmail.com" in March 2023. Eventually, on February 23, 2024, he added the test files "good-large_compressed.lzma" and "bad-3-corrupt_lzma2.xz". These two malicious files are executed when the macro "build-to-host.m4". Additional small contributors Jigar Kumar and Dennis Ens made contributions to the project by pressuring the inclusion of malicious code as essential functionality and by causing pointless conflicts with the patch release schedule during the years 2022 to 2024 [11].

Microsoft developer "Andres Freund"'s suspicion on the unusual CPU usage patterns and Valgrind failures enabled him to track down this supply chain attack. This curious Microsoft engineer saw that the SSHD process was utilising a lot of CPU power, which was causing the Debian Linux distribution's SSH-based login to function slowly. This gave Andres the opportunity to investigate the reason for the delay, conduct the necessary analysis based on his expertise, and report the issue to the relevant Linux distribution authorities i.e., oss-security via mail on 29 March 2024 [7]. He notified them with the high-level dynamics of the attack that he had discovered during his analytical process. He discovered that the XZ utils package contained a backdoor committed to the open source linux distribution package code. The backdoor was archieved in the download package known as the tarballs and targeted the x86-64 linux.

3.2 PyPI Project attack

Pytoileur(Backdoor target): The supply chain attack's target programme is the PYPI project and its users, through introducing the package pytoileur. The purpose of this package is to automate and manage tasks in Python projects. It mostly automates the work of repetitive development. By offering a streamlined workflow for routine operations like configuring development environments, handling dependencies, and conducting tests, it improves productivity. Include linting and testing tools in your code to guarantee quality. Automate the deployment procedures. This package was actually trojonised with malicious code. That targeted at the user's crypto-based application data and web browser data.

Malicious Actors: On May 25, 2024, a developer named "PhilipsPY" joined the PYPI(Python Package Index) project. PYPI is an open source library that allows Python developers to reuse code; there are a lot of packages available for download and installation. He uploaded the malicious code-encoded in Pytoileur package. After joining the open community discussion on stackoverflow, "EstAYA G" (promoter) began responding to other users' questions about Python

development. By responding to questions on stackoverflow, he promoted the installation of the pytoileur package as solution to their problems.

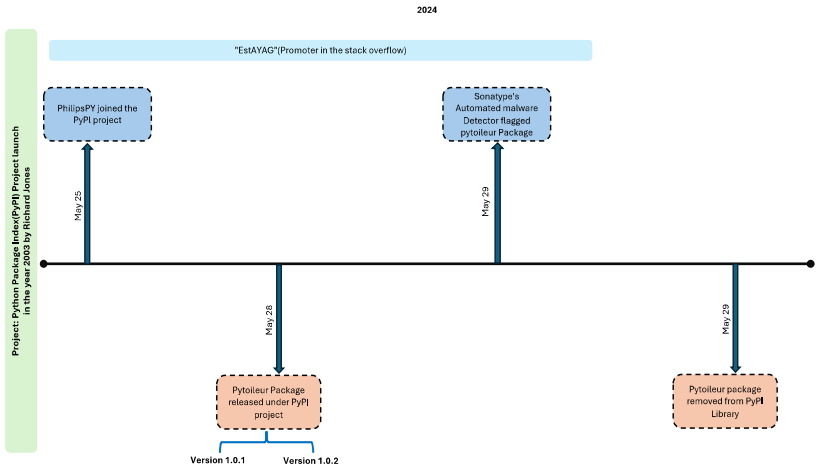


Fig. 2: Attack activity timeline for PyPI project

The Project PYPI, included a package Pytoileur version 1.0.1 and version 1.0.2. This package was published on the same day by the author PhilipsPy in the PYPI library on 28 May 2024. We draw a timeline for the attack events(see Fig. 2). It had a total number of 264 downloads This package setup.py included a hidden malicous code. In its metadata, the package calls itself a "Cool package." The HTML webpage description refers to the package as a "API Management tool written in Python." The package is also trying to typosquat users of legitimate packages like "Pyston," as evidenced by the code's reference to "pystob," a now-defunct package (more on that later). The setup.py was injected with whitespaces after the print() statement in its code which can be avoided by the text editors those use the wordwrap. The whitespaces actually was hiding the code that executed the base64-encoded that connected to the remote server. Its payload stated that it connected "http: 51.11.140.144:8086/dl/runtime". Connecting to this external server it retrieved the windows binary called "runtime.exe". This file was undetected by the file scanning software and firewall. The "runtime.exe" was executed using the powershell and vbscript of the system. Once the execution of the "runtime.exe" is performed it dropped more suspicious file that modify the windows registry settings and deploy harmful payloads. This leads to the execution of the main.exe file which gathers information from the web browsers and the crypto currency services. webbrowsers such as Brave, firefox, google etc and well known crypto currency services such as the Coinbase, Binance, Paysafecard, crypto.com, exodus etc data were retrieved by this execution. Sonatype's automated malware detection engine for sonatype repository firewall detected and flagged the pytoileur package in PYPI repository on 28 May 2024. The Sonatype security researcher Jeff Thornhill noticed

the line 17 whitespaces in the setup.py of the pytoileur package was hiding a code. This allowed him to further analyse this backdoor and track the issue as "SONATYPE-2024-1783" [12]. Later raised an alert to the PYPI admins and declared it a software based supply chain attack. The developer "PhilipsPY" and the promoter ID "EstAYA G" for pytoileur package were discontinued from development community [13]. Sonatype is a business that specialises in supply chain management software. The Sonatype company's automatic malware detection tool was one of their tools that assisted us in finding the backdoor in the earlier mentioned Python programme. This business develops tools and services for identification and the fixing of vulnerabilities in OSS with a focus on security of the OSS's weakness and danger. They support the creation of software security policies. They encourage community engagement and tool enhancement.

4 Generic framework

These days, supply chain attacks based on software have become commodified. On the dark web, configurations, and portions of code for exploiting software vulnerabilities during development are offered for sale. This commodification is majorly known as a Supply chain attack as a service(SCAaaS). Since supply chain attacks are becoming more accessible and commercialized, they can be easily replicated by targeting software development environments and taking advantage of the update process. Software libraries, pirated or open-source development tools, and their distribution networks are its primary targets. As a result, the frequency of attacks increased, having a widespread impact and making the process of detection and attribution more difficult. Based on the case studies this is a preliminary analysis.

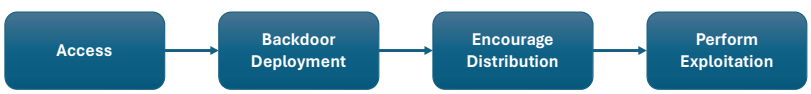


Fig. 3: OSS-based Supply Chain Attack strategy

There are two primary objectives of a supply chain attack: embed malicious code into software and to promote its use and distribution. As a result, there are two types of criminals involved in a supply chain attack: those who create and insert malicious code and those who promote or encourage on download of software that contains harmful code. These attackers work together to gain access to the target software project, deploy the backdoor code, encourage the software distribution of that project that includes the backdoor and finally exploit the backdoor to gain access to the system and the data (as shown in Fig. 3).

We provide a generic architecture for supply chain attacks that are OSS-based(see Fig. 4). This figure clarifies the trade-off that exists between the attacker, the vulnerability, and the exploitation of it. The attacking group's developers attempt to get associated with one of the targeted software's core development projects. The malicious developers begin injecting malicious code into the

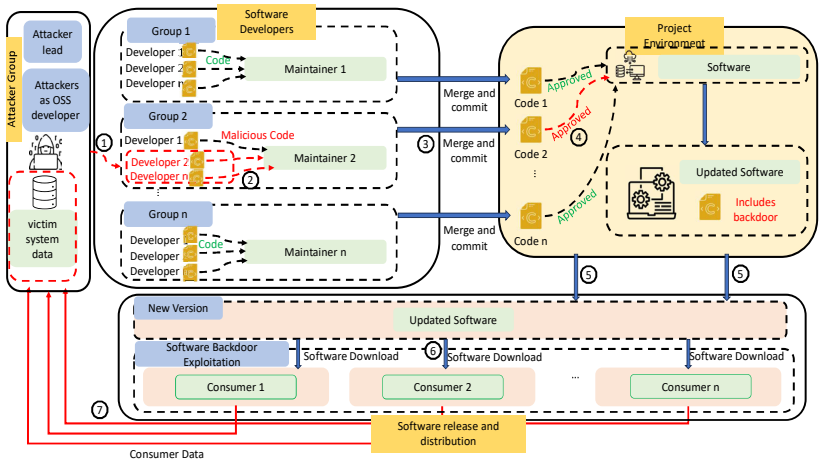


Fig. 4: OSS-based Supply Chain Attack framework

program's code as soon as they have access to it as authorized developers. The malicious code can be an argument, single-line code or a dependent executable file. To prevent being discovered, this code has been encoded or obfuscated. It is also possible to hide or remove logs produced by this type of programming. During development, the attacker group's developer deactivates alerts for administrative commands, memory utilization, or intrusion detectors. This is accomplished over a longer period by gradually incorporating the malicious code while remaining undetected by other legitimate developers or maintenance leads. Even if any potentially harmful code is discovered, the promoters defend it so well that others are led to believe that the code is necessary and urgent for the project. Since the maintainer or project lead must take the demands of the public into account, they approve the code because they think the developer is legitimate and has good intentions. Once the code is deployed into the software with all its dependencies satisfied, it becomes a backdoor for that software. Later when the software is released by the project authors on their official websites. The promoters working with the malicious developer and team initiate discussions suggesting to download and install the software as solution for most of the queries of users. The users of the OSS then start downloading and installing the dependent packages or software. Finally, when the package or the software is in use the injected backdoor gets executed on the user's system. This backdoor allows exploitation of user data by exposing it to the attacker groups.

To avoid OSS-based attacks, the development of software bill of materials(SBOM) for instance has made it possible to identify dependencies, which aren't directly part of an application but are installed or launched when the application is deployed as well as components that developers directly integrated into an application by importing them into its source code as a potential solution [15]. As a result, after researching current detection techniques, we suggest com-

bining their use with development platforms during the software development lifecycle.

5 Proposed Platform

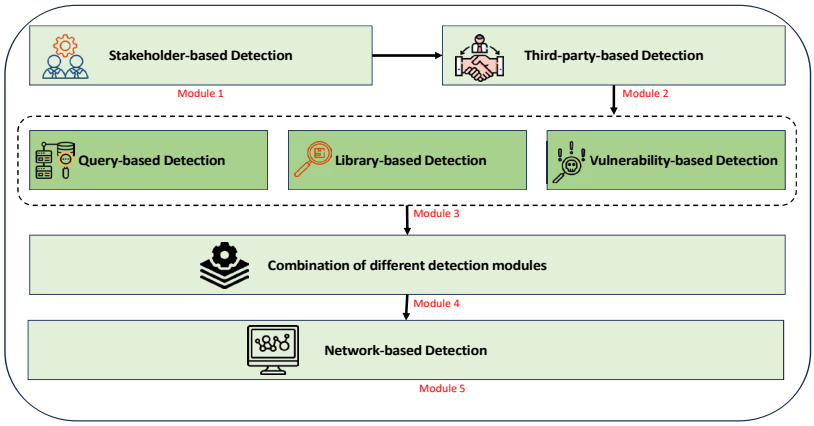


Fig. 5: OSS-Security Quality testing Platform

Software security quality checks require detection prior to OSS software distribution. This makes the OSS development Lifecycle require a Security platform. The lack of data labeling for malicious semantics, the uncertainty of the attack design, and the irregular patterns of supply chain attacks based on OSS make the detection approaches in table 1 insufficient. We propose an OSS Security Quality testing platform using the prior study-based detection approach type in 5. This platform will provide software is secure quality tests based on all the detection approaches based on current standards by incorporating the best modules available. Module 1, The Stakeholder-based detection provides their rightful intent, background verifications, and control. Module 2, Third-party-based detection allows authentication, and control privileges and monitors the stakeholder's activity once they are part of the project. Module 3, is a combination of three different detection approaches Query-based, Library-based, and Vulnerability-based detection as they all focus on OSS semantics, dependency, and logic. After Semantic and logic check the OSS should go through Module 4, Combinations of different detection modules based on components of the OSS. For instance, if the OSS Utilizes outside resources the connectivity, data transfer and infrastructures should be analyzed well with security tools that are developed resource-based for security. Finally, Module 5, Network-based detection, scans for malicious package exchange by providing different combinations of benign and malicious parameters.

By using the OSS-Security Quality Testing Platform with the best detection techniques for each module, OSS development, and utilization, organizations can improve the security of their software and systems and increase their capacity for identifying and thwarting supply chain attacks on OSS.

6 Conclusion

The XZ Utils and PyPI Project attacks were multiyear strategized attacks. They followed the common attack strategy. This motivated us to create the Generic framework for the OSS-based Supply Chain Attack. These attacks remained undetected throughout the development and release process of the packages. The study detection strategies during the development process and after the release of the software is crucial which assisted us in finding the lacunas in these defences. The most critical lacuna for this strategy is ignorance based on trust. Protocols and Standards should be followed strictly to reduce the possibility of such attacks. Our proposed OSS Security Quality Testing platform is an attempt towards securing OSS. For Future works we plan to propose more detailed combinations of the modules in the platform and its feasibility as potential solution for OSS Security.

7 Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) (Project No. 2022-0-00701, 10%; Project No. RS-2023-00228996, 10%, Project No.RS-2022-00165794, 10%), the ICT R&D Program of MSIT/IITP (Project No. 2021-0-01816, 10%), and a National Research Foundation of Korea (NRF) grant funded by the Korean Government (Project No. RS2023-00208460, 60%).

References

1. Ladisa P, Plate H, Martinez M, Barais O. Taxonomy of attacks on open-source software supply chains. arXiv preprint arXiv:2204.04008. 2022 Apr 8.
2. Thompson K. Reflections on trusting trust. Communications of the ACM. 1984 Aug 1;27(8):761-3.
3. Statista Homepage, https://www.statista.com/statistics/1375128/supply-chain-attacks-software-packages-affected-global/
4. Statista Homepage, https://www.statista.com/statistics/1268934/worldwide-open-source-supply-chain-attacks/
5. Ladisa P, Plate H, Martinez M, Barais O. Sok: Taxonomy of attacks on open-source software supply chains. In2023 IEEE Symposium on Security and Privacy (SP) 2023 May 21 (pp. 1509-1526). IEEE.
6. Seah J. Sliced cables and cyber-attacks: How safe is our internet?. News Weekly. 2024 May(3164):22-3.
7. Openwall Homepage, https://www.openwall.com/lists/oss-security/2024/03/29/4
8. Security Week: Supply Chain Attack: Major Linux Distributions Impacted by XZ Utils Backdoor https://www.securityweek.com/supply-chain-attack-major-linux-distributions-impacted-by-xz-utils-backdoor/
9. Lins M, Mayrhofer R, Roland M, Hofer D, Schwaighofer M. On the critical path to implant backdoors and the effectiveness of potential mitigation techniques: Early learnings from XZ. arXiv preprint arXiv:2404.08987. 2024 Apr 13.
10. The Tukaani Project, https://tukaani.org/xz-backdoor/
11. Evan Boehs, https://boehs.org/node/everything-i-know-about-the-xz-backdoor

12. Sonatype, https://www.sonatype.com/blog/pypi-crypto-stealer-targets-windows-users-revives-malware-campaign
13. Black Hat Ethical Hacking, https://www.blackhatethicalhacking.com/news/
14. SNYK Security Database, https://security.snyk.io
15. America's Cyber Defense Agency, https://www.cisa.gov/sbom
16. Andreoli A, Lounis A, Debbabi M, Hanna A. On the prevalence of software supply chain attacks: Empirical study and investigative framework. Forensic Science International: Digital Investigation. 2023 Mar 1;44:301508.
17. Malka J. Increasing Trust in the Open Source Supply Chain with Reproducible Builds and Functional Package Management. In46th International Conference on Software Engineering (ICSE 2024)-Doctoral Symposium (DS) Track 2024.
18. Nahum M, Grolman E, Maimon I, Mimran D, Elyashar A, Brodt O, Elovici Y, Shabtai A. Ossintegrity: Collaborative Open Source Code Integrity Verification. Available at SSRN 4711134.
19. Singla T, Anandayuvaraj D, Kalu KG, Schorlemmer TR, Davis JC. An empirical study on using large language models to analyze software supply chain security failures. InProceedings of the 2023 Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses 2023 Nov 30 (pp. 5-15).
20. O'Donoghue E, Reinhold AM, Izurieta C. Assessing Security Risks of Software Supply Chains Using Software Bill of Materials. In2nd International Workshop on Mining Software Repositories for Privacy and Security, MSR4P&S,(SANER 2024), Rovaniemi, Finland 2024 Mar.
21. Haque BM. An Analysis of SBOM in the Context of Software Supply-chain Risk Management (Master's thesis).
22. Mirakhorli M, Garcia D, Dillon S, Laporte K, Morrison M, Lu H, Koscinski V, Enoch C. A Landscape Study of Open Source and Proprietary Tools for Software Bill of Materials (SBOM). arXiv preprint arXiv:2402.11151. 2024 Feb 17.
23. Gokkaya B, Karafili E, Aniello L, Halak B. Global supply chains security: a comparative analysis of emerging threats and traceability solutions. Benchmarking: An International Journal. 2024 Mar 4.
24. Younis AA, Hu Y, Abdunabi R. Analyzing Software Supply Chain Security Risks in Industrial Control System Protocols: An OpenSSF Scorecard Approach. In2023 10th International Conference on Dependable Systems and Their Applications (DSA) 2023 Aug 10 (pp. 302-311). IEEE.
25. Nygård AR, Katsikas SK. Ethical hardware reverse engineering for securing the digital supply chain in critical infrastructure. Information & Computer Security. 2024 Jan 15.
26. Vashisth M, Verma SK. State of the Art Different Security Challenges, Solutions on Supply Chain: A Review. In2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA) 2023 Mar 14 (pp. 427-431). IEEE.
27. Turksonmez H, Ozcanhan MH. ENHANCING SECURITY OF RFID-ENABLED IOT SUPPLY CHAIN. Malaysian Journal of Computer Science. 2023 Jul 31;36(3):289-307.
28. Boughton L, Miller C, Acar Y, Wermke D, Kästner C. Decomposing and Measuring Trust in Open-Source Software Supply Chains. InProceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results 2024 Apr 14 (pp. 57-61).
29. Chauhdary SH, Alkatheiri MS, Alqarni MA, Saleem S. An efficient evolutionary deep learning-based attack prediction in supply chain management systems. Computers and Electrical Engineering. 2023 Jul 1;109:108768.

A Proposal of a Supply Chain Security Model for Generative AI

Keun Young Lee[1*] and Jiyeon YOO[2]

[1] Financial Security Institute, 143 Uisadang-daero, Yeongdeungpo-gu, Seoul, Republic of Korea
starnyou2@gmail.com
[2] Sangmyung University, 20 Hongjimun 2-gil, Jongrogu, Seoul, Republic of Korea
yooo@smu.ac.kr

Abstract. As technology continues to evolve and integrate into every aspect of our lives, effective technology management is becoming increasingly critical. Among the latest significant topics in this field is supply chain security management. The growing complexity of technology and the expansion of open-source platforms, cloud services, and other technologies have amplified the roles and authorizations of various third parties involved in the technology supply chain. This has led to a rise in supply chain attacks leveraging these third parties, as such attacks are relatively accessible for attackers and highly efficient; a single successful attack can compromise similar systems.

Generative AI systems are particularly vulnerable to supply chain attacks due to their complex and interconnected nature. These systems comprise various elements such as data, models, and infrastructure that continuously evolve, forming a dynamic system ecosystem. As a result, compromising even a minor component of the generative AI system ecosystem can jeopardize the entire system.

In this paper, we emphasize the importance of supply chain security for generative AI. We define supply chain security for generative AI through an analysis of the structure of these systems and propose a security model for managing supply chain security. This model is based on a comprehensive examination of major security standards and guidelines related to generative AI.

We conceptualize generative AI supply chain security as the implementation of strategies, processes, and controls to secure the entire lifecycle of a generative AI system—from source, design, and development to deployment and maintenance. Our aim is to protect data preprocessing, source code, learning algorithms, third-party libraries, and related components such as prompts and infrastructure from potential vulnerabilities, threats, and attacks. This includes securing the AI system development process, ensuring the reliability of service providers, and implementing continuous surveillance and vulnerability management techniques, and more.

Keywords: AI(Artificial Intelligence), Generative AI, Adversarial Machine Learning, AI Security, Cyber Supply Chain Security

* Corresponding Author

1 Instruction

Starting with the launch of ChatGPT (Generative Pre-trained Transformer) ('22.10), the spread and development of many services based on generative AI, including the launch of various Large Language Models (LLM) mainly by overseas big tech companies, With the popularization of AI, it is having a significant impact on our way of life.

Generative AI has the characteristics of collecting large amounts of data, training through this, and creating text, images, codes, and videos, but reliability and safety issues, hallucinations, jailbreaks, and deepfakes that arise from the use of the algorithm. Concerns about the adverse effects of generative AI, including social and ethical issues, are increasing. The World Economic Forum (WEF, '23.1.) report mentioned both the advantages and risks of generative AI, and stated that the impact of this technology must be carefully managed [1].

Recently, security researchers on the Huntr bug bounty platform discovered important vulnerabilities in famous platforms such as Hugging Face, MLflow, and ClearML, raising concerns about the security of AI solutions used in various industries [2]. This vulnerability can manipulate data sets through remote code execution and read sensitive files, which can pose a serious risk to the entire service. The uses of generative AI are diverse, including publicly generative AI applications and web services, publicly generative AI APIs, cloud solutions, and in-house development. In general, the resources of AI systems are highly dependent on third-party IT infrastructure and data supply chain. In these cases, the data cannot be easily controlled, and tracking the data being used in the model is also very difficult. And vulnerabilities in supply resources can lead not only to the integrity of training data, models, and deployment platforms, but also to biased results in AI model training, security breaches, or even overall system failures. As such, supply chain security, including potential risks related to various suppliers related to AI systems, their data, and the processing of data, is very important.

In terms of the learning and creativity characteristics of generative AI, generative AI systems create models through collecting and learning a large amount of data from unverified sources such as the Internet, and products utilizing the algorithm may have problems with bias and errors inherited from pre-learned data. This can lead to reliability and safety issues in AI systems. Therefore, the security risks of generative AI begin with data and occur during each step of the model improvement process, such as the use of the foundation model and fine tuning, as well as distribution and operation. There is a need for discussion on ways to strengthen risk management.

Accordingly, in order to strengthen the security of generative AI, specific security considerations are required throughout the data and models that make up the generative AI system, as well as the resources, design, development, deployment, operation, and management processes. In this paper, we conceptualize how to define supply chain security for generative AI and propose a security model for generative AI for supply chain security management through analysis of major security standards and guidelines related to generative AI. Through this model, we present specific security considerations to strengthen the security of generative AI.

2 Theoretical Background

2.1 Concept and structure of generative AI

Generative AI is a subset of AI consisting of a model that can generate new content by responding to prompts based on training data [3]. The representative model of generative AI is the foundation model, a concept that first appeared in a Stanford University paper [4]. It is a general-purpose AI model that is pre-trained on a wide range of datasets and can be applied to various use cases, including Transformer Model, LLM, diffusion model, etc. It was analyzed that neural networks constitute an important category called foundation models. Generative AI models are largely divided into foundation models that use pre-trained models as is, and fine-tuned models such as fine tuning, which trains again and adjusts pre-trained models for new data sets, depending on the data training method.

2.2 Concept and structure of ICT supply chain security

As ICT(Information and Communication Technology) is applied to the supply chain and interconnectivity increases in the hyper-connected era, the supply chain is becoming more complex. As the resources and service providers required when building a system in a company are complexly connected, security management becomes more difficult and important in the process of outsourcing and re-entrusting the design, development, production, operation, maintenance, and disposal of products and services.

According to NIST SP 800-161r1 [5], supply chain elements include organizations, entities, and tools used in the research and design, development, manufacture, purchase, distribution, integration, operation, and maintenance or disposal of systems and components. A supply chain refers to a set of resources and processes linked between multiple levels of an enterprise, each beginning with the sourcing of products and services and extending through the product and service life cycle. Supply chain-wide cybersecurity risk refers to the potential for damage or damage to a supplier, its supply chain, products or services.

3 Supply Chain Security Model of Generative AI

3.1 Supply Chain Security Issues and Components of Generative AI

We derived four major issues that should be considered in the supply chain security of generative AI and the most important components of the generative AI supply chain security model.

First, as an issue of generative AI supply chain security, it is necessary to consider generative AI security at the supply chain level.

In the ICT supply chain, the share of open-source software in recent software development is rapidly increasing, so to ensure transparency and reliability of the SW supply chain, we focus on SW components, create A software bill of materials (SBOM) at each stage of the SW supply chain, and build on this. Ways to build a SW supply chain

security management system are being explored. SBOM declares the inventory of components used to build a software artifact such as a software application. The U.S. National Telecommunications and Information Administration (NTIA) It was announced [6] that as the minimum SBOM components, the data fields must include seven items: Supplier Name, Component Name, Version of Component, Other Unique IDs, Dependency Relationship, Author of SBOM Data, and Timestamp. The components of the ICT supply chain include hardware, operating systems, commercial software, binaries, intellectual property rights, embedded software, and open-source software, etc., so it is important to secure visibility based on the characteristics of each component, which is important throughout the entire supply chain process. It must be controlled and managed. The supply chain of generative AI includes the resources, technologies, and other elements involved in the process of developing and maintaining products and services. As generative AI providers may seek to integrate software, data, models and/or remote services provided by third parties into their own systems, traceability of the resources that make up the system becomes more difficult. The recent emergence of various open source-based LLMs and their increased use, as well as the high dependence of resources forming AI systems on third-party IT infrastructure and data supply chains, are forming an increasingly complex supply chain. Therefore, considering the characteristics of open-source models and high data dependency compared to existing software, we must consider strategies, processes, and controls for security management in the complex supply chain of generative AI.

Second, technical security issues must be considered from the perspective of generative AI operating principles.

Data selection and curation are essential to developing and deploying trustworthy generative AI models. Because models may reproduce inaccurate or incorrect information in the training data, it is also important for model developers to consider the representativeness or accuracy of the data in the dataset and to assess whether the data contains personal information or bias [3]. Training data poisoning is a problem that occurs across AI, but in generative AI, training data can be supplied from a variety of sources that can be difficult to control, such as the Internet, making it more vulnerable to such attacks, and external training data published on the Internet. There is a risk that bias may be inherited or lead to sensitive training data output. In addition, generative AI models that use foundation models as they are or undergo fine-tuning may be at increased risk of such attacks through the use of transfer learning, a model optimization method that uses a pre-learned AI model for one task and applies it to other similar problems when the pre-learned model is manipulated. Therefore, generative AI has the main characteristic of starting from data and having a training stage of the model, and technical security issues that arise during this process must be considered.

Third, it is necessary to consider generative AI risk issues according to the training process.

OWASP presents the top 10 vulnerabilities of LLM applications, including Prompt Injection, which manipulates LLM through crafted input to execute the attacker's intention; Supply Chain Vulnerabilities, which compromise the security of the model, training data, and ML models and deployment platforms, causing biased results, security breaches, or complete system failures; and Insecure Plugin Design, which causes

harmful results such as data leakage, remote code execution, and privilege escalation due to insufficient access control and improper input validation [7]. Generative AI is vulnerable to a variety of attacks and is segmented into the three most common AI-related attacks: poisoning attacks, evasion attacks, and privacy attacks [8]. Abuse attacks of generative AI systems include attacks that involve hate speech, promoting discrimination, and specific attacks. Images, text, or malicious code, such as inciting group violence, are created and used for attacks. Adversarial Attack[9], a major type of AI attack, is an attack using adversarial examples that causes the AI model to produce incorrect results by adding small-sized perturbations to the data that cannot be distinguished with the naked eye. Types of hostile attacks are divided into Poisoning Attack, Evasion Attack, and Exploratory Attack. Representative examples of Exploratory Attack include Model Inversion, Membership Inference, and Model Extraction via APIs [10]. The main attack details are as shown in table 1.

Table 1. Adversarial Attack Types of Generative AI Risks.

category	Main Content
Poisoning Attack	• During the training phase, an attacker changes the data or model to cause the AI algorithm to malfunction [11] • The attack goal is to cause maximum malfunction by using poison data, which is malicious data that is trained to lead to incorrect results of the training model[12]
Evasion Attack	• An attacker interferes with the normal operation of the model by causing a small perturbation to the input data of the trained AI model[10][12] • It is an attack that deceives AI by manipulating input data at the application stage after training, and has a wide range of applications [10][12]
Model Inversion	• An attack that leaks input values from the output values of the AI model [11][12][13]
Membership Inference Attack	• The attacker infers whether the data he or she has is used to train the AI model [11][12][13]
Model Extraction Attack	• An attack in which the attacker repeatedly queries the AI model to extract the entire model based on the input and output values [14]

Poisoning attacks occur during the training stage, and evasion attacks and model inversions occur during the utilization stage after training is completed [11][13]. The attacks in table 1 do not occur only at specific stages of the AI lifecycle, but occur at multiple stages. It can appear throughout.

Generative AI risks continue to develop into various types of attacks, such as using Adversarial Machine Learning to intervene in the training and inference process to produce unintended results or to bias certain results.

Attacks based on the training stage of the generative AI system, including the pre-training stage of the foundation model and the fine-tuning stage of the AI model, can be classified into ① Training-time Attack and ② Inference-time Attack [15]. In the case of training time attacks, the foundation model is vulnerable to poisoning attacks where the attacker controls a subset of the training data, usually by scraping data from public sources. During the AI model fine-tuning stage, it may be vulnerable to poisoning attacks depending on the attacker's knowledge and capabilities. In the case of inference time attacks, LLM and RAG (Retrieval-Augmented Generation) applications do

not provide separate channels for data and instructions, so inference time attacks using data channels are possible.

Therefore, Adversarial Machine Learning continues to develop into various types of attacks, such as interfering in the training and reasoning process to produce unintended results or biased adjustments to specific results, making it possible to mitigate generative AI risks that occur at each stage of the machine learning process. A solution is needed.

Fourth, specific technical and managerial security considerations are needed throughout the supply chain of generative AI.

Research on real-world security vulnerabilities for machine learning suggests that security is best addressed comprehensively, including software, data and model supply chains, networks, and storage systems [15]. The supply chain is vulnerable and can impact the integrity of training data, machine learning models, and deployment platforms. These vulnerabilities can lead to biased results, security breaches, or complete system failures. Machine learning is expanded through pre-trained models and training data provided by third parties, which are vulnerable to tampering and poisoning attacks [7]. Due to this scalability, if connected to a vulnerable service, risks of internal information leakage and model malfunction may occur, so generative AI supply chain security must be considered throughout the system.

OECD (oecd.ai) presented five comprehensive basic principles (① Inclusive and sustainable growth and well-being ② Human rights and democratic values, including fairness and privacy ③ Transparency and Explainability ④ Robustness, Security and Safety ⑤ Accountability) for trustworthy AI in the “AI Recommendation ('19.5.)” unanimously adopted by member countries through national AI policy analysis, and traditional Security considerations for AI systems were presented with a focus on declarative considerations, such as reviewing ethical issues to satisfy reliability requirements.

However, in order to secure the supply chain of generative AI, it is necessary to present specific security requirements for each core component regarding technical security issues, reflecting the structure and operating principles of the generative AI system as well as reliability requirements. Therefore, if some security threats in the generative AI supply chain are considered to have a ripple effect on the entire process, from the preparation of data required for training to the results returned when using the AI system, security issues should be considered at all stages rather than at specific processing stages.

Finally, the life cycle of a generative AI system compared to a traditional AI system of prediction and judgment is as shown in Table 2. When mapping the life cycle stages of a generative AI system to the dimensions of the system's main components, the technical elements of data, models, and systems, as well as the management elements of business understanding, risk management, governance, and compliance, are derived. In particular, the components called ‘data’ and ‘model’ are different from from the SDLC (Software Development Lifecycle) process of ISO/IEC 12207: Requirement Analysis→Design→Implement→Testing→Evolution. Therefore, it is important to apply security considerations to the main components that make up the generative AI system.

Table 2. The Main Components of a Generative AI System.

<table>
<tr><th colspan="3" rowspan="2">Lifecycle of traditional AI systems</th><th rowspan="2">Lifecycle of Generative AI systems</th><th rowspan="2"></th><th colspan="3">Component</th></tr>
<tr><th>OECD[16][17]</th><th>NIST (AI RMF 1.0)</th><th>EU ENISA[18]</th><th>Technical</th><th>Manage</th></tr>
<tr><td>Planning and Design</td><td rowspan="4">Design, Data and Model</td><td>Planning and Design</td><td>Business Goals</td><td>Planning</td><td rowspan="6">→</td><td rowspan="6">Data</td><td rowspan="11">System</td><td rowspan="11">Understanding of business, Risk Management and Gorvernance, Compliance</td></tr>
<tr><td rowspan="2">Data Collection and Processing</td><td rowspan="2">Data Collection and Processing</td><td>Data Collection</td><td rowspan="2">Data Collection and preparation</td></tr>
<tr><td>Data Exploration</td></tr>
<tr><td>Data Preprocessing</td><td rowspan="2">Data management and operations</td></tr>
<tr><td>Feature Selection</td></tr>
<tr><td>Model Building and Use</td><td>Model Building and Use</td><td>Model Selection and Build</td><td rowspan="3">Design and development</td><td>Model selection</td><td rowspan="5">Model</td></tr>
<tr><td colspan="2">Verification and Validation</td><td>Verification and Validation</td><td>Model Training</td><td></td></tr>
<tr><td colspan="2" rowspan="2"></td><td rowspan="2">Model Optimization</td><td>Training and fine-tuning</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>Model Application</td><td>Inference and evaluation</td><td>→</td></tr>
<tr><td colspan="2">Deployment</td><td>Deployment and Use</td><td>Model Deployment</td><td rowspan="3">Operation and management</td></tr>
<tr><td colspan="2" rowspan="2">Operation and Monitoring</td><td>Operation and Monitoring</td><td>Model Maintenance</td></tr>
<tr><td>Use or Impact</td><td>Business Understanding</td></tr>
</table>

3.2 Supply Chain Security Framework of Generative AI

Supply chain security of generative AI can be defined as security involved in the process of supplying, designing, developing, modifying, deploying, and operating components, with data and models learned from data forming the generative AI system as core components. By considering the four security issues derived for generative AI supply chain security and analyzing the components of generative AI supply chain security, we will apply the generative AI supply chain security framework as follows.

First, in order to construct a supply chain security model for generative AI, some of the framework that applies the concept of supply chain elements such as systems and components and process steps for cybersecurity management are borrowed according to the definition of the ICT supply chain, but the supply chain of generative AI is Reflects security specialties.

The components of the generative AI supply chain are derived from mapping the generative AI lifecycle stages to the main component dimensions of the AI system, as

shown in Table 2. The core components are technical (data, models, systems) and managerial (business understanding, risk management and governance, compliance) elements.

In the complex generative AI supply chain, security considerations are important to secure visibility based on the characteristics of the supply chain components, and as in ICT supply chain security, source and processing for each component that makes up the generative AI system for the entire AI system life cycle. Risks occurring at each stage of the process must be considered. For example, as the sharing and distribution of generative AI models becomes common, attackers can maliciously distribute or change AI models, and if developed using a modified model, it is possible to cause serious security problems for the entire AI system. Additionally, because AI systems evolve through data-based training, changes to AI models over time may require frequent maintenance and response to new risks. Accordingly, from the perspective of the operation principle of generative AI, technical security matters and generative AI security risk response measures according to the training process are reflected.

Therefore, the supply chain security model for generative AI should be created to include the process from source and design/development to distribution/operation management for the components of generative AI. Finally, we will present specific security considerations throughout the entire supply chain of generative AI through this model.

3.3 Supply chain security model of generative AI

The supply chain security model of generative AI consists of the processes of source, design and development, deployment, and operation management by component in Table 3, and proposes security considerations to protect the entire life cycle of generative AI systems. Through this model, we aim to protect generative AI-related components from potential vulnerabilities, threats, and attacks through security control that considers the overall ripple effect from the perspective of the technical operation principle of data learning and the model improvement process such as fine-tuning of generative AI for the complex generative AI supply chain.

Through this model, we aim to protect the components of generative AI systems from potential vulnerabilities, threats, and attacks through security control that considers the overall ripple effect from the perspective of the technical operation principle of data learning and the model improvement process such as fine-tuning of generative AI, as well as security management for complex generative AI supply chains.

Security considerations through this model include documents from standards and standards organizations on the security of foreign generative AI on supply chain security issues of generative AI [15][19][20], and domestic guidelines for the introduction and use of generative AI[21][22], security guidelines for generative AI from foreign government agencies [23][24][25][26], and security and governance reports for generative AI from other major organizations[27][28]. It was analyzed and presented thoroughly. If security considerations overlap for each component process step, they will be presented first in the steps that must take precedence or be important.

Table 3. Supply chain security model of generative AI.

Process / component		Source	Design and Development	Deployment and Operations Management
Technical	Data	• Data source and information [20][24][25][27] ※ Example of a data card template : Data subject, source, version, privacy and security handling, collection method, appropriate/inappropriate use cases, etc. • Privacy and security [27] • Maintaining supply chain integrity [15][19][27]	•Design of collection pipeline architecture [24][27] •Data minimization, personal and sensitive information processing [21][24][26][27] •Data preprocessing, removal of duplicate/similar data [21][22][23][24][28]	• Data information documentation and management [24][27] • AI data privacy and security management [19][21][22][27] • Manage service-generated data [19][21] • Establish data leak prevention and security measures [22]
	Model	• Model source and information [25][28] ※ Example of a model card template : Model version, training data and methodology, performance metrics, privacy and bias or restrictions, etc. •Supply chain AI deployment assurance practices [15][28] ※ Supply chain assurance practices : Adoption of secure model persistence formats, cryptographic hashing as integrity check for web-scale data, and immunization of images to prevent manipulation by models.	• Model design and architecture decision [22][24][25][27] • SbD (Security by Design), PbD (Privacy by Design) design [25][26] • Review of model security, such as model cards and risk cards, and supply chain security evaluation [15][24][28] • Development of training and fine-tuning processes[22][24][27] • Fine-tuning and ensuring model safety through human feedback such as RLHF (Reinforcement Learning from Human Feedback)[3][15][26] •Prompt Engineering [15][24][27] • Perform model and prompt documentation [25][27] • Improved learning efficiency and performance, output optimization, and learning pipeline security [27][28] • Model bias, fairness, and safety [23][27]	•Strengthening security and responsible AI launch through AI Red Team [15][25][28] • Deploy DevOps, CI/CD, utilize release assurance practices, manage model versions and information, deploy and maintain [14][22][27][28] • Manage model and input/output security, secure AI plugins [21][22][25][27] • Evaluate and validate prompts, evaluate and monitor model performance, and maintain quality standards [22][24][27] • Maintain appropriate human participation in automated processes (Human-In-The Loop) and strengthen supervision [24]

	System		• Check infrastructure and quality requirements [25][27] • System Security measures [23] •Clear interface design and user education about risks [24][25]	• Manage identity and access [21][27] • Manage system security and performance [21][22] • System operation, input monitoring [21][24][25]
Manage	Under-standing of busi-ness		•Establishing a business case for AI adoption [28] •Supply chain security assessment, requiring suppliers to comply with standards [25] • AI operation continuity management, accident prevention and response procedure development [23][25][26][28] • Technical Debt Management [25][28]	
	Risk Manage-ment and Gorvern ance		• Identifying and listing AI assets [20][24][25][28] • Performing risk management of risk assessment and selection/implementation of protective measures [20] • Establishing governance, assigning organizational roles and responsibilities [20][28]	• RMF Operation, TEVV Verification [20][28] ※Test, Evaluation, Verification, and Validation • Security and Privacy Protection Mechanism Operation [20][24] • Third party, supply chain monitoring [20][25] • AI Security / Privacy Protection Education, Awareness Raising [20][28]
	Compli-ance		• Reviewing compliance with legal requirements and regulatory compliance [20][21][23][28] • Verifying SLAs, reviewing contracts and insurance [20][23][28]	

4 Conclusion

Supply chain attacks, which are easy to attack but have high impact, are expanding to repository platforms used for developing and sharing generative AI. Given the characteristics of generative AI's open-source model and high dependency on data, supply chain security management for generative AI is an important issue. In addition, generative AI starts from training data and goes through a model improvement process such as fine tuning, so this process can lead to overall system errors and serious security risks, such as unintended or biased results. Accordingly, a plan to strengthen risk management is needed to eliminate these risks in advance and secure reliability.

In this paper, we conceptualize how important supply chain security is and how to define generative AI supply chain security through structural analysis of generative AI systems, and generative AI for supply chain security management through analysis of major security standards and guidelines related to generative AI. A security model was designed and proposed.

Generative AI's supply chain security model was proposed by analyzing the structural characteristics and security risks of generative AI and approaching it from a new perspective of supply chain security by reflecting the special characteristics of the process for data and models, which are core components of generative AI. In addition, the significance of this study lies in the inclusion of specific security considerations by approaching it from a multi-faceted perspective by responding to each process of the generative AI system components through the model. We believe that this will be meaningful research that will help implement a practical and effective risk management and governance system for generative AI in the future.

References

1. AI TIMES, “Generative AI is a game changer, necessary for societal level”, January 10, 2023.
2. Daily Secu, “Beware of critical security vulnerabilities in open source AI platforms such as MLflow and ClearML”, January 22, 2024.
3. OECD, on, “AI Language Models Technological, Socio-Economic and Policy Considerations”, OECD Digital Economy Papers, No.352, April 2023.
4. Center for Research on Foundation Models (CRFM), Stanford Institute for Human-Centered Artificial Intelligence (HAI), Stanford University, “On the Opportunities and Risks of Foundation Models”, arXiv:2108.07258, August 16, 2021.
5. NIST, “Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations”, NIST SP 800-161, r1, May 2022.
6. National Telecommunications and Information Administration, “Framing Software Component Transparency:Establish a Common Software Bill of Materials(SBOM), October 2021.
7. OWASP, “OWASP TOP 10 for LLM Application, Ver 1.1, October 16, 2023.
8. Federal Office for Information Security, “Generative AI Models, Opportunities and Risk for Industry and Authorities”, Ver.1.1, April 4, 2024.
9. Ian J, et al., “Explaining and Harnessing Adversarial Examples”, aRxIV:1412.6572, 2015.5.
10. Aniban Charkraborty, et al., “Adversarial Attacks and Defenses: A Survey”, aRxiv:1810.00069, Sep 2018.

11. ENISA, “Securing Machine Learning Algorithms”, Dec 2021.
12. Sohee Park and Dae-seon Choi, “Artificial Intelligence Security Issues”, Review of KIISC, 27(3), June 2017.
13. Myeong-Ju Kim and Su-Kyung Lee, “Cyber threats and countermeasures due to the development of artificial intelligence technology,” Information and Communications Magazine, 39(12), Nov 2022.
14. Florian Tramer et.al, “Stealing machine learning models via prediction APIs”, aRxIV:1609.02943, Sep 2016.
15. NIST, “Adversarial Machine Learning, Adversarial Machine Learning, A Taxonomy and Technology of Attack and Mitigations”, AI 100-2e2023, January 2024.
16. OECD, “Scoping the OECD AI principles”, OECD Digital Economy papers, No.291, November 2019.
17. OECD, “Framework for the Classification of AI Systems", Digital Economy Papers, No, 323, February 2022.
18. ENISA, “AI Cybersecurity Challenges-Threat Landscape for Artificial Intelligence”, December 2020.
19. ETSI, “Securing Artificial Intelligence (SAI); Data Supply Chain Security”, Group Report SAI 002 V1.1.1, August 2021.
20. NIST, “Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile” NIST AI 600-1 Initial Public Draft, April 2024.
21. National Intelligence Service, National Security Technology Research Institute, “Security guidelines for using generative AI such as Chat GPT”, June 2023.
22. Digital Platform Government Committee, NIA, “Guidelines for introducing and utilizing large-scale AI in the public sector”, April 2023.
23. Japan Ministry of Internal Affairs and Communications Ministry of Economy, Trade and Industry, “AI Guidelines for Business” ver 1.0, April 19, 2024.
24. Central Digital and Data Office, “Generative AI framework for HM Government”, ver.1.0, January 18, 2024.
25. US CISA, UK NCSC, “Guidelines for Secure AI System Development”, October 2024.
26. Singapore Infocomm Media Development Authority, AI Verify Foundation, “PROPOSED MODEL AI GOVERNANCE FRAMEWORK FOR GENERATIVE AI”, January 16, 2024.
27. EY, “Generative AI Risk and Governance, Way towards a responsible and trusted AI”, January 3, 2024.
28. OWASP, “LLM AI Cybersecurity & Governance Checklist”, February 19, 2024.

Privacy Enhanced P2P Power Trading Using DID-based Identity Verification in a Microgrid

Taeyang Lee and Jong-Hyouk Lee

Dept. of Computer and Information Security & Convergence Engineering for Intelligent Drone, Sejong University, Seoul, Republic of Korea
taeyang@pel.sejong.ac.kr, jonghyouk@sejong.ac.kr

Abstract. Smart meters play a crucial role in accurately measuring power usage and distribution data in microgrids. However, fine-grained data collection raises significant privacy concerns. To address this, it is essential to develop a distributed power trading system that is both trustworthy and privacy-preserving. In this paper, we propose a peer-to-peer power trading mechanism that ensures privacy through identity verification based on decentralized identifiers, zero-knowledge proofs, and pseudonyms. Besides, we introduce an efficient claims verification scheme with Boneh-Lynn-Shacham aggregate signature. We also validate the security of our proposed mechanism through the Real-or-Random model.

Keywords: DID· ZKP· Privacy Enhancing· Smart Metering.

1 Introduction

Conventional grids suffer from environmental pollution, resource shortages, and regional imbalances in power distribution, degrading end-user Quality of Service (QoS) and Quality of Experience (QoE) [1]. Distributed Energy Resource (DER) systems, which are small-scale generators, typically ranging from 3kW to 10,000kW, such as solar panels and small wind turbines, are being used to overcome these limitations [2]. It has led to the emergence of microgrids, which group power generation, storage, and load locally and generate power from such renewable energy sources as solar, wind, and biomass. Therefore, they can help address environmental pollution and resource shortages. They also resolve the regional power distribution imbalances by allowing end-users to generate and use their own energy and distribute surplus power. This is where the concept of prosumers, end consumers who can produce, consume, and sell power, in microgrids emerged. They are capable of distributed power trading with end consumers. Particularly through Information and Communications Technology (ICT), it is possible to measure and monitor data in real-time in microgrids. This enables Demand-Side Management (DSM), which controls power production and distribution to more accurately reflect the energy supply and demand in a region, enabling efficient power distribution control and dynamic pricing [3]. Smart meters play a major role in enabling distributed power trading by precisely measuring the energy supply and demand status at the end-users side.

However, fine-grained data collection in real-time raises privacy concerns. Suppose the data is collected and analyzed over a long period of time. In that case, it is possible to identify the behavior patterns of end-users, like when the users get up, when they come back from work, their power purchase/sale volume and price, etc. The disclosure of such information could have adverse consequences for users, potentially leading to a breach of privacy [4].

In this paper, we design a microgrid based on a permissioned blockchain to prevent service interruption caused by a Single Point Of Failure (SPOF). The proposal also aims to protect privacy by controlling the participation of nodes in the network and regulating access to data based on their authorities. Additionally, we utilize pseudonyms in communications so that the real identity of the other party cannot revealed even if data is exchanged between the parties for a long period of time. A Self-Sovereign Identity (SSI) model is designed based on a Decentralized Identifier (DID) which allows data subjects to control the scope of information required for identity verification. In the SSI model, we utilize Boneh-Lynn-Shacham (BLS) aggregate signature to improve verification efficiency for multiple claims. Privacy is enhanced by using Verifiable Presentation (VP) with Zero-Knowledge Proof (ZKP) in the model.

This paper is organized as follows. Section 2 discusses related works and briefly describes a signature algorithm and ZKP as preliminaries to the proposal. Section 3 presents a microgrid system based on a permissioned blockchain and describes an identity verification protocol for auction-based Peer-to-Peer (P2P) power trading. Section 4 shows the results of the security analysis of the proposed protocol. Finally, Section 5 concludes this paper.

2 Preliminaries

2.1 Existing Works for Enhancing Privacy in Power Trading

Diao et al. [4] proposed a Privacy-preserving Smart Metering scheme using Linkable Anonymous Credential (PSMLAC), which is an anonymous credential protocol based on the Camenisch-Lysyanskaya (CL) signature. It can provide identity anonymity, message authentication, and traceability to failed smart meters. However, it only addresses privacy issues in smart metering and not in P2P power trading. Xie et al. [5] proposed a privacy-preserving distributed power trading framework, namely the Private Energy Market (PEM). Under the theory of secure Multi-Party Computation (MPC), it is possible to allocate energy transactions and calculate prices without revealing private data by performing all computations using a novel cryptographic protocol based on homomorphic ciphers. However, they do not consider identity verification and authentication mechanisms. Hassan et al. [3] proposed a blockchain-based power auction method, namely the Differentially private Energy Auction for bLockchain-based microgrid systems (DEAL), which utilizes a consortium blockchain for the efficiency of the system. Differential privacy technique was utilized to ensure privacy in this system. However, it does not consider mutual authentication in the process of P2P power trading.

2.2 BLS Aggregate Signature

The Boneh-Lynn-Shacham (BLS) signature algorithm based on bilinear mapping is one of the digital signature algorithms. Let G_1, G_2, and G_T each be a multiplicative cyclic group composed of a large prime number p. Then, $e : G_1 \times G_2 \rightarrow G_T$ satisfies the following three properties.

- Bilinear: $e(aU, bV) = e(U, V)^{ab}$ for all $U \in G_1$, $V \in G_2$, and $a, b \in Z_p^*$.
- Non-degeneracy: There exists $U \in G_1$ and $V \in G_2$, such that $e(U, V) \neq 1$. That is the map does not send all pairs into the identity in G_T.
- Computability: For all $U \in G_1$ and $V \in G_2$, the polynomial-time algorithm calculation can be found.

The BLS signature algorithm operates in four steps: initialization, key pair generation, aggregated signature generation, and verification. In the initialization step, system parameters $(G_1, G_2, G_T, e, g_1, g_2, p, H)$ are set, where g_1 and g_2 are generators, and $H : 0, 1^* \rightarrow G_2$ is a hash function. A random number $x \in Z_p$ is chosen, and $v = g_1^x \in G_1$ is computed to generate a key pair. Let $sk = x$, which is a private key, and $pk = v$, which is a public key. A message m and sk are given as inputs to generate a BLS signature. A hash value h for m can be obtained through a hash function, and then $h^{sk} \in G_2$ is calculated to generate a signature σ. The inputs used are m, pk, and σ to verify the signature. After hashing m as h, $c = e(pk, h)$ and $c' = (g_1, \sigma)$ are computed. If $c' = c$, *Accept* is returned. Otherwise, *Reject* is returned as a verification result.

This algorithm can aggregate signatures into a single signature. When the number of users $u_i \in U$ is k, each u_i generates an individual signature σ_i for a different message $m_i \in \{0, 1\}^*$. The multiple σ_i $(i = 1, 2, 3, \ldots, k)$ are used as inputs to generate an aggregated signature $\sigma_{agg} \in G_2$ by computing $\sigma_{agg} = \prod_{i=1}^{k} \sigma_i$. When verifying σ_{agg}, it requires multiple pk_i, m_i, and σ_i $(i = 1, 2, 3, \ldots, k)$ as inputs to calculate $c = \prod_{i=1}^{k} e(pk_i, H(m_i))$ and $c' = e(g_1, \sigma_{agg})$. If $c' = c$, *Accept* is returned. Otherwise, *Reject* is returned as a verification result [6].

BLS signatures can increase the efficiency of signature verification by aggregating multiple signatures into a single signature. Taking advantage of this feature, this paper aims to improve the efficiency of identity verification by generating aggregated signatures for multiple claims contained in VCs.

2.3 Non-Interactive Zero-Knowledge Proofs

ZKP is a cryptographic technique used to prove that a statement is true without revealing any additional information beyond the validity of the statement itself. As such, ZKP can be used where user privacy is important, as they allow the prover to prove his claim without revealing information relevant to his claim. Therefore, in this paper, we aim to enhance privacy by applying ZKP to the claims contained in VPs in the SSI model.

There are two types of ZKP: interactive and non-interactive. In this paper, we use a non-interactive ZKP called Zero-Knowledge Succinct and Non-Interactive

Arguments of Knowledge (zk-SNARK), which can verify proofs within a few milliseconds. zk-SNARK works through the *Setup*, *Prove*, and *Verify* algorithms. To describe the algorithms, let $u \in \mathcal{L}$ be a statement known to both a prover and verifier. Then, for any u, if there exists a witness w known only to the prover, then (u, w) belongs to a polynomial-time decidable binary relation $\mathcal{R}$ (i.e., $(u, w) \in \mathcal{R}$) [7].

- $pk, vk \leftarrow Setup(\mathcal{R})$: A trusted party generates a proving key pk and verifying key vk with a given $\mathcal{R}$.
- $\pi \leftarrow Prove(\mathcal{R}, pk, u, w)$: A prover generates a proof π by multiplying u and w by some polynomials.
- $0/1 \leftarrow Verify(\mathcal{R}, vk, u, \pi)$: A verifier utilizes vk to verify π when u is given. The verifier approves the proof (1), if the equation of the three pairings holds. Otherwise, the proof is rejected (0).

3 Privacy Enhanced P2P Power Trading

3.1 System Model

The proposed decentralized microgrid consists of three layers: physical layer, blockchain layer, and application layer.

Physical Layer. This layer may consist of Utility Service Providers (USPs), DERs, power management equipment, transmission systems, and end-consumers. The elements that are relevant to P2P power trading are the USP, smart meters, and end-consumers. The USPs provide P2P power trading services. Smart meters measure power-related data on the end-user side. There are two types of end-consumers: prosumers and general consumers. Prosumers can generate and consume power themselves, as well as sell surplus energy. Typical consumers do not have the ability to generate power, but they can purchase energy from prosumers through auctions and consume it.

Blockchain Layer. This layer is made up of blockchain functions. P2P communication takes place over the P2P network. Consensus algorithms are utilized to process transactions and agree on block generation. Smart contracts support auction automation. Decentralized ledger records various data and is also used as a Verifiable Data Registry (VDR) to manage data related to identity verification.

Application Layer. This layer consists of supporting services for P2P power trading. An automated auction-based P2P power trading service is provided to assist users in selecting a suitable counterparty using smart contracts. An identity verification service based on an SSI model is provided to ensure reliable and secure transactions. In the SSI model, a USP acts as an issuer and a Certificate Authority (CA) issuing Verifiable Credentials (VCs) to holders. A holder requests the issuer to issue a VC and submits a VP containing the necessary claims to prove its identity to a verifier, who verifies the VP. In the proposed model, each smart meter acts as both a holder and a verifier, i.e., it acts as a

holder when it needs to prove its identity and as a verifier when it needs to verify the identity of the other party. The identity verification data is recorded in the VDR. Fig. 1 shows the proposed SSI model for P2P power trading.

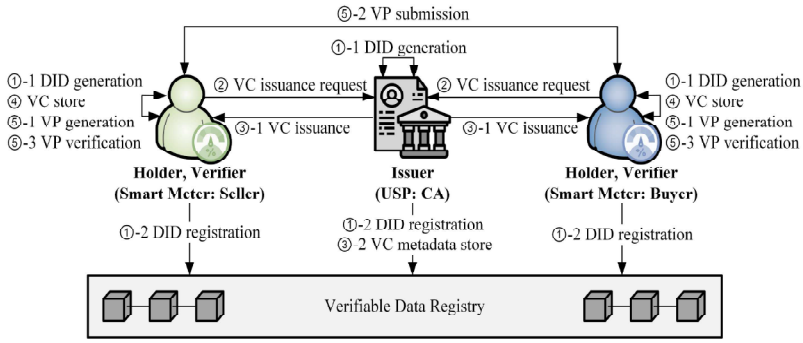


Fig. 1. Proposed SSI Model for P2P Power Trading

3.2 Assumptions

This section establishes the following assumptions for the supported SSI model to perform secure and reliable P2P power trading in a decentralized microgrid.

- During the registration, each party communicates through a secure channel.
- All parties fully trust USP and adversarial intervention by the USP is impossible.
- Verifiers must be able to identify USP through a trust issuer list.
- As an issuer, USP must define and register VC schemas in VDR, which verifiers can use to verify the holder's VC.
- USP installs smart meters in each household in advance and manages the relevant information (e.g., owner, installed location, etc.). Therefore, the USP can identify the smart meters and issue VCs for each smart meter.
- Smart meters must be mutually authenticated each time to participate in the auction, even if authentication has previously taken place between the same smart meters.

3.3 Proposed P2P Power Trading Mechanism

In the proposed system, P2P power trading is performed in three phases: registration, auction initiation, and bid submission. Before executing the three phases, the proposed system needs to set the system parameters. The USP and each smart meter should then generate their DIDs for identification and key pairs for message signing. The DIDs are resolved through DID documents containing public keys, metadata for identifiers, etc. The DID documents are recorded in the VDR. The key pairs are generated using the BLS key pair generation algorithm. The notations used to describe the specific operations of the proposed system are shown in Table 1.

Table 1. Notations

Notations	Descriptions	Notations	Descriptions
DID_A	DID	$VCproof_A$	VC proof field
PID_A	Pseudonym	VC_A^{meta}	VC metadata
T_A	Timestamp	VR_A	Verification result
Req_{VC}	VC issuance request	auc	A set of auction conditions
$Ddoc_A$	DID document	bid_A	Bidding
$IPFS$	Interplanetary file system	ZVP_A	ZKP VP
K_A^s	Private key	$VPID_A$	VP identifier
K_A^p	Public key	$VPproof_A$	VP proof field
VC_A	VC	SK_{A-B}	Session key
$VCID_A$	VC identifier	EM_A	Encrypted message

Registration Phase In this phase, smart meters are registered in the system. This is done by issuing VCs for the smart meters. For example, if a VC is issued for one smart meter SM_i, the interaction between SM_i and USP is as follows, where SM_i functions as both a buyer and a seller of electricity. Fig.2 presents the detailed process.

1. SM_i constructs a message M_{SM_i} with Req_{VC}, DID_{SM_i}, PID_{SM_i}, and T_1, and sends $M_1 = \{M_{SM_i}, \sigma_{M_{SM_i}}\}$ to USP.
2. USP first checks $|T_1 - T_1^*| \leq \triangle T$ and then queries $Ddoc_{SM_i}$ in VDR using DID_{SM_i}. VDR returns $Ddoc_{SM_i}$ by sending M_2. USP then checks $\sigma_{M_{SM_i}}$ with M_{SM_i} and $K_{SM_i}^s$. If $\sigma_{M_{SM_i}}$ is valid, USP sends M_3 to store DID_{SM_i} and PID_{SM_i} in $IPFS$. Next, USP queries SM_i's claims in $IPFS$, and $IPFS$ returns a set of claims $C_{SM_i} = \{claim_1, claim_2, claim_3, \ldots, claim_n\}$ via M_4. Afterward, USP generates a set of signatures S for $\{claim_i\}_{i=1}^n$ using K_{USP}^s with the BLS signature algorithm. It also generates a signature $\sigma_{C'_{SM_i}}$ for C'_{SM_i}, derived by concatenating $\{claim_i\}_{i=1}^n$, and constructs $VCproof_{SM_i} = \{S, \sigma_{C'_{SM_i}}\}$ for a VC. Finally, USP can obtain $VC_{SM_i} = \{VCID_{SM_i}, DID_{USP}, DID_{SM_i}, C_{SM_i}, VCproof_{SM_i}\}$ of SM_i. USP and then generates pk_{SM_i} and vk_{SM_i} and constructs $VC_{SM_i}^{meta} = \{VCID_{SM_i}, status, vk_{SM_i}\}$ and stores it in VDR via M_5. USP also generates T_2, and then signs $M_{USP} = \{VC_{SM_i}, pk_{SM_i}, T_2\}$ with K_{USP}^s. $M_6 = \{M_{USP}, \sigma_{M_{USP}}\}$ is then sent to SM_i.
3. When SM_i receives M_6 from USP, it checks $|T_2 - T_2^*| \leq \triangle T$ and queries $Ddoc_{USP}$ in $IPFS$ with DID_{USP}. $IPFS$ then returns $Ddoc_{USP}$ via M_7. Afterward, SM_i checks $\sigma_{M_{USP}}$ using K_{USP}^p obtained in $Ddoc_{USP}$ and M_{USP}. If $\sigma_{M_{USP}}$ is valid, it generates σ_{agg} with $VCproof_{SM_i}$ for verifying VC_{SM_i}. SM_i then verifies σ_{agg} using K_{USP}^p, C_{SM_i}, and C'_{SM_i}, if the result is $True$, it stores VC_{SM_i} and pk_{SM_i}.

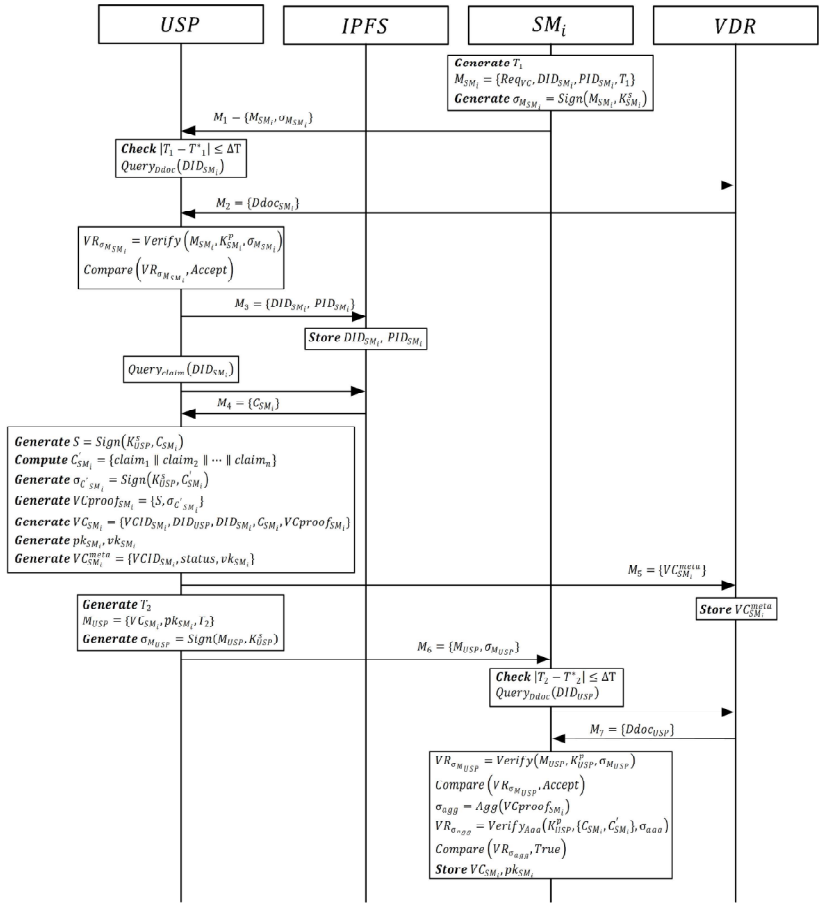


Fig. 2. Sequence Diagram of the Registration Phase

Auction Initiation Phase. This is the phase where SM_i^p, a party that wants to sell power, starts the auction. The auction starts with SM_i^p setting auction conditions and propagating them to nearby smart meters. SM_i^p and SM_j^b that is willing to participate in the auction to purchase power must verify the legitimacy of each other through identity verification. For example, the process of one SM_i^p interacting with one or more SM_j^b for an auction is as follows. Fig.3 illustrates the detailed process.

1. SM_i^p selects claims $C'_{SM_i^p} = \{claim_j\}_{j=1}^m$, which is required to prove its identity from $C_{SM_i^p} = \{claim_j\}_{j=1}^n$ in its VC. SM_i^p then generates $W_{SM_i^p} = \{C'_{SM_i^p}, S'_{SM_i^p}\}$ with $C'_{SM_i^p}$ and $S'_{SM_i^p} = \{\sigma_{claim_j}\}_{j=1}^m$, a set of signatures for selected each $\{claim_j\}_{j=1}^m$. $W_{SM_i^p}$ corresponds to a set of witnesses that

only SM_i^p knows for the ZKP. SM_i^p then generates $VC'_{SM_i^p} = \{VCID_{SM_i^p}, DID_{USP}, U_{SM_i^p}\}$, where $U_{SM_i^p}$ is a set of statements to be proven. It also generates $VPproof_{SM_i^p}$ using inputs $\mathcal{R}$, $pk_{SM_i^p}$, $U_{SM_i^p}$, and $W_{SM_i^p}$ and creates $ZVP_{SM_i^p} = \{VPID_{SM_i^p}, VC'_{SM_i^p}, VPproof_{SM_i^p}\}$, random value p_i to compute secret value $P_i = p_i \times G$, and *auc* including power volume, minimum bid, auction time, etc. Afterward, SM_i^p starts auction by propagating $M_8 = \{PID_{SM_i^p}, auc, ZVP_{SM_i^p}, P_i, T_3\}$ to near one or more SM_j^b.

2. Each SM_j^b checks $|T_3 - T_3^*| \leq \Delta\ T$ and *auc* and retrieves the $VC_{SM_i^p}^{meta}$ from VDR. VDR sends M_9 to return $VC_{SM_i^p}^{meta}$. Afterward, each SM_j^b verifies the status value *status* contained in $VC_{SM_i^p}^{meta}$ to check the status of $VC'_{SM_i^p}$ included in $ZVP_{SM_i^p}$. If the status is valid, each SM_j^b verifies $ZVP_{SM_i^p}$ using inputs $\mathcal{R}$, $vk_{SM_i^p}$, $U_{SM_i^p}$, and $VPproof_{SM_i^p}$. When the result is returned as *Accept*, it means SM_i^p is legitimate.

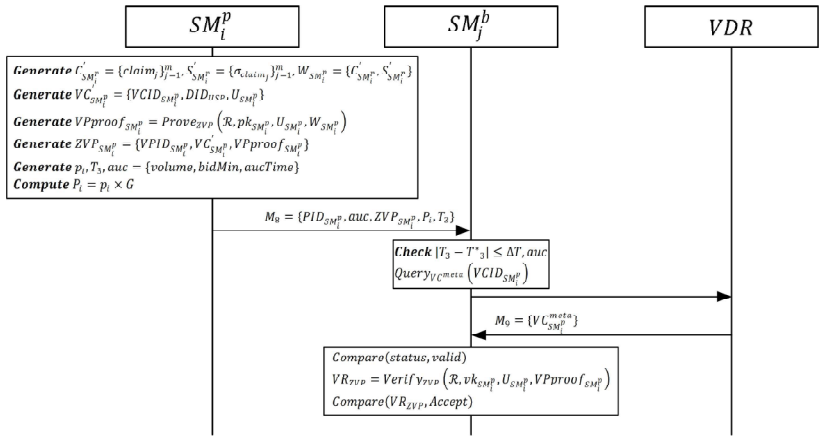


Fig. 3. Sequence Diagram of the Auction Initiation

Bid Submission Phase. This is the phase where each SM_j^b places bids. Each SM_j^b interacts with SM_i^p by presenting its ZKP-based VP required to prove its identity and bid. For example, the process of one SM_j^b presenting its VP and bidding to SM_i^p is as follows. Fig. 4 depicts the detailed process.

1. SM_j^b generates $ZVP_{SM_j^b}$ to participate in auction by proving its identity. The $ZVP_{SM_j^b}$ generation process is the similar to that of SM_i^p. After creating $ZVP_{SM_j^b}$, SM_j^b generates random number b_i and T_4 and then computes secret values $B_i = b_i \times G$ and $SK = P_i \times b_i$. Afterward, it generates $SK_{SM_i^p - SM_j^b} = H\left(P_i \parallel B_i \parallel SK \parallel VPID_{SM_j^b}\right)$. It sets $bid_{SM_j^b}$ and

encrypts $PID_{SM_j^b}$, $bid_{SM_j^b}$, and $ZVP_{SM_j^b}$ with SK to form $EM_{SM_j^b}$. After that, SM_j^b then places a bid for power purchase by sending $M_{10} = \{EM_{SM_j^b}, B_i, T_4\}$ to SM_i^p.

2. When SM_i^p receives M_{10}, it checks $|T_4 - T_4^*| \leq_\triangle T$ and then derives $SK' = p_i \times B_i$. Then, SM_i^p decrypts $EM_{SM_j^b}$ using derived SK' and computes $SK'_{SM_i^p - SM_j^b} = H\left(P_i \parallel B_i \parallel SK' \parallel VPID_{SM_j^b}\right)$. After that, SM_i^p checks whether the SM_j^b's $bid_{SM_j^b}$ is at least $bidMin$, a crucial threshold in the matching counterparty process, and verifies $ZVP_{SM_j^b}$ if the $bid_{SM_j^b}$ meets auc. The verification process for SM_j^b is the same as for SM_j^b verifying $ZVP_{SM_i^p}$. If the result of $ZVP_{SM_j^b}$ verification is $Accept$, it means that SM_j^b is legitimate.

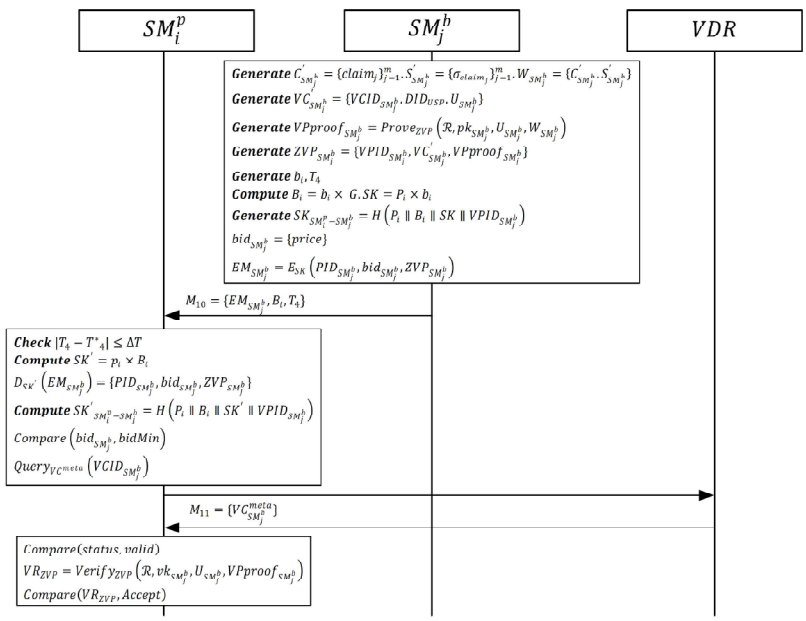


Fig. 4. Sequence Diagram of the Bidding

SM_i^p can perform identity verification for multiple SM_j^b by receiving VPs and bids from them. In this scenario, SM_i^p can determine the best trading partner by considering the bids submitted by one or more SM_j^b to perform P2P power trading. Sein et al. [8] proposed the power trading algorithm. The specific process of determining an auction destination based on bids can be adopted from [8].

4 Security Analysis

In this section, the security of the session key in the proposed mechanism is demonstrated through the Real-or-Random (RoR) model. The RoR model, based on probabilistic game theory, is widely used to prove the semantic security of session key agreements [9]. In the model, an adversary $\mathcal{A}$ interacts with the t-th instance of the participants in execution. Specifically, the instances of SM_i^p and SM_j^b participating in P2P power trading are defined as $\mathcal{P}_{SM_i^p}^{t1}$ and $\mathcal{P}_{SM_j^b}^{t1}$, respectively. Additionally, the RoR model considers four queries that assume the actual attacks of the adversary: $Execute\left(\mathcal{P}_{SM_i^p}^t, \mathcal{P}_{SM_j^b}^t\right)$, $Send\left(\mathcal{P}^t, M\right)$, $Reveal\left(\mathcal{P}^t\right)$, $Test\left(\mathcal{P}^t\right)$.

- $Execute\left(\mathcal{P}_{SM_i^p}^t, \mathcal{P}_{SM_j^b}^t\right)$: $\mathcal{A}$ can eavesdrop on the messages exchanged between $\mathcal{P}_{SM_i^p}^{t1}$ and $\mathcal{P}_{SM_j^b}^{t1}$.
- $Send\left(\mathcal{P}^t, M\right)$: $\mathcal{A}$ can send a message M to the instance $\mathcal{P}^t$ and receive a response message from $\mathcal{P}^t$.
- $Reveal\left(\mathcal{P}^t\right)$: $\mathcal{A}$ can obtain the current session key between $\mathcal{P}^t$ and its communication partner.
- $Test\left(\mathcal{P}^t\right)$: $\mathcal{A}$ can request the session key from $\mathcal{P}^t$ and obtain it, with the result being a probabilistic response based on the outcome of an unbiased coin flip c. If $c = 1$, $\mathcal{P}^t$ returns the session key, and if $c = 0$, returns a random number. Otherwise, null ($\perp$) is returned.

Additionally, the collision-resistant one-way hash function $Hash$ is utilized as a random oracle, allowing access to all parties and $\mathcal{A}$.

Theorem 1. *When $\mathcal{A}$ attempts to break the semantic security of the proposed protocol, $\mathcal{A}$ can query $Hash$ up to q_{hash} times. The length of the hash output is denoted as $|Hash|$. When $\mathcal{A}$ attempts to break the Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP), $\mathcal{A}$'s advantage is denoted as $Adv_{\mathcal{A}}^{ECCDHP}(t)$. Thus, the advantage of $\mathcal{A}$ in breaking the session key security in the proposed protocol can be estimated as $Adv_{\mathcal{A}}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2Adv_{\mathcal{A}}^{ECCDHP}(t)$.*

Proof. Following games G_i $(i = [0, 3))$ are considered to prove the semantic security of the session key in the proposed protocol. The advantage of $\mathcal{A}$ winning in each G_i is denoted as $Pr\left[Succ_{G_i}\right]$. The process of each game is as follows.

- G_0: In this game, the actual attack by $\mathcal{A}$ is modeled while the proposed protocol is in operation. At this point, $\mathcal{A}$ does not perform any queries and cannot obtain any information. Therefore, since c is chosen randomly, the advantage of $\mathcal{A}$ follows the following Equation (1).

$$Adv_{\mathcal{A}}(t) = \left|2Pr\left[Succ_{G_0}\right] - 1\right| \tag{1}$$

– G_1: In this game, $\mathcal{A}$ is modeled as performing an eavesdropping attack. $\mathcal{A}$ $Execute\left(\mathcal{P}^t_{SM^p_i}, \mathcal{P}^t_{SM^b_j}\right)$ to obtain $M_8 = \{PID_{SM^p_i}, auc, ZVP_{SM^p_i}, P_i, T_3\}$ and $M_{10} = \{EM_{SM^b_j}, B_i, T_4\}$. Afterward, to ascertain whether the derived $SK_{SM^p_i - SM^b_j}$ is the actual key or a random key, $\mathcal{A}$ invokes $Reveal\left(\mathcal{P}^t\right)$ and $Test\left(\mathcal{P}^t\right)$. For $\mathcal{A}$ to correctly derive $SK_{SM^p_i - SM^b_j}$, the short-term secret values p_i and b_i are required. However, $\mathcal{A}$ cannot determine these values from the obtained messages. Therefore, since M_8 and M_{10} do not help $\mathcal{A}$ increase the probability of a successful attack, $\mathcal{A}$'s advantage in G_1 is similar to that in G_0 and is derived as shown in Equation (2).

$$Pr\left[Succ_{G_1}\right] = Pr\left[Succ_{G_0}\right] \tag{2}$$

– G_2: In this game, $\mathcal{A}$ uses M_8 and M_{10} to compute $SK_{SM^p_i - SM^b_j}$ by executing $Hash$ and $Send$ queries. However, the eavesdropped M_8 and M_{10} are protected by a one-way hash function. Additionally, $\mathcal{A}$ attempts to derive $SK_{SM^p_i - SM^b_j}$, which requires solving the ECCDHP within polynomial time t to find $SK = b_i \times P_i = b_i \times (p_i \times G) = p_i \times (b_i \times G) = p_i \times B_i$. However, this is computationally infeasible due to the birthday paradox and the ECCDHP. Therefore, according to the birthday paradox and the hardness of the ECCDHP, the attacker's advantage in G_2 is determined as shown in Equation (3).

$$\left|Pr\left[Succ_{G_1}\right] - Pr\left[Succ_{G_2}\right]\right| \leq \frac{q^2_{hash}}{2\left|Hash\right|} + Adv^{ECCDHP}_{\mathcal{A}}\left(t\right) \tag{3}$$

After all the games are done, $\mathcal{A}$ guesses a correct c, leading to Equation (4).

$$Pr\left[Succ_{G_2}\right] = \frac{1}{2} \tag{4}$$

By using Equations (1) and (2), we can derive Equation (5).

$$\frac{1}{2}Adv_{\mathcal{A}}\left(t\right) = \left|Pr\left[Succ_{G_0}\right] - \frac{1}{2}\right| = \left|Pr\left[Succ_{G_1}\right] - \frac{1}{2}\right| \tag{5}$$

Next, using Equations (3), (4), and (5), we can derive Equation (6).

$$\frac{1}{2}Adv_{\mathcal{A}}\left(t\right) = \left|Pr\left[Succ_{G_1}\right] - Pr\left[Succ_{G_2}\right]\right| \leq \frac{q^2_{hash}}{2\left|Hash\right|} + Adv^{ECCDHP}_{\mathcal{A}}\left(t\right) \tag{6}$$

Finally, by multiplying both sides of Equation (6), we obtain the predetermined $Adv_{\mathcal{A}}\left(t\right)$ as shown in Equation (7). This corresponds to a negligible value.

$$Adv_{\mathcal{A}}\left(t\right) \leq \frac{q^2_{hash}}{\left|Hash\right|} + 2Adv^{ECCDHP}_{\mathcal{A}}\left(t\right) \tag{7}$$

5 Conclusion

This paper addresses privacy issues in microgrids during decentralized power trading, where smart meters collect fine-grained real-time data. We proposed a decentralized microgrid system using a permissioned blockchain to protect privacy and introduced an identity verification mechanism based on DID-based SSI, which enhances privacy through VP using ZKP. The P2P power trading mechanism utilizes pseudonyms during message exchanges to protect participants' identities. Additionally, leveraging VC based on the BLS signature algorithm enables efficient aggregate signature verification. However, BLS signatures are less efficient for individual signature tasks compared to the elliptic curve digital signature algorithm, indicating the need for further research. The security of the proposed mechanism was validated using the RoR model, ensuring the semantic security of session keys.

Acknowledgments. This work was supported by Korea Internet & Security Agency (KISA) grant funded by the Korea government(PIPC) *(No.RS-2023-00253686, Development of Standards for Personal Information Protection in the Blockchain Environment)*

References

1. Aggarwal, S., Kumar, N.: PETS: P2P Energy Trading Scheduling Scheme for Electric Vehicles in Smart Grid Systems. IEEE Transactions on Intelligent Transportation Systems **23**(9), 14361–14374 (2022)
2. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid—The new and improved power grid: A survey. IEEE communications surveys & tutorials **14**(4), 944–980 (2011)
3. Hassan, M. U., Rehmani, M. H., Chen, J.: DEAL: Differentially private auction for blockchain-based microgrids energy trading. IEEE Transactions on Services Computing **13**(2), 263–275 (2019)
4. Diao, F., Zhang, F., Cheng, X.: A privacy-preserving smart metering scheme using linkable anonymous credential. IEEE Transactions on Smart Grid **6**(1), 461–467 (2014)
5. Xie, S., Wang, H., Hong, Y., Thai, M.: Privacy preserving distributed energy trading. In: 2020 IEEE 40th International Conference on Distributed Computing Systems, ICDCS, pp. 322-332 (2020)
6. Liu, X., Feng, J.: Trusted blockchain oracle scheme based on aggregate signature. Journal of Computer and Communications **9**(3), 95–109 (2021)
7. Salleras, X., Daza, V.: ZPiE: Zero-knowledge proofs in embedded systems. Mathematics **9**(20), 2569 (2021)
8. Myung, S., Lee, J. H.: Ethereum smart contract-based automated power trading algorithm in a microgrid environment. The Journal of Supercomputing **76**(7), 4904–4914 (2020)
9. Thakur, G., Kumar, P., Jangirala, S., Das, A. K., Park, Y.: An effective privacy-preserving blockchain-assisted security protocol for cloud-based digital twin environment. IEEE Access **11**, 26877–26892 (2023)

Quantum Implementation of LSH

Yujin Oh, Kyungbae Jang, and Hwajeong Seo

Hansung University, Seoul (02876), South Korea

oyj0922@gmail.com, starj1023@gmail.com, hwajeong84@gmail.com

Abstract. As quantum computing progresses, the assessment of cryptographic algorithm resilience against quantum attack gains significance interests in the field of cryptanalysis. Consequently, this paper proposes the depth-optimized quantum circuit of Korean hash function (i.e., LSH) and estimates its quantum attack cost in quantum circuits. By utilizing an optimized quantum adder and employing parallelization techniques, the proposed quantum circuit achieves a 78.8% improvement in full depth and a 79.1% improvement in Toffoli depth compared to previous the-state-of art works. In conclusion, based on the proposed quantum circuit, we estimate the resources required for a Grover collision attack and evaluate the post-quantum security of LSH algorithms.

Keywords: Quantum circuit · Quantum collision attack · LSH.

1 Introduction

The advancement of quantum computing presents new challenges and opportunities in cryptography. The parallel processing capability of quantum computers can exponentially enhance the speed of breaking many traditional cryptographic systems. Particularly, it enables finding solutions to classical hard problems such as large-scale factorization and discrete logarithm problems at a much faster rate.

Moreover, the development in quantum computing allows for the discovery and application of new quantum algorithms. These algorithms can be employed to uncover vulnerabilities in currently used classical cryptographic systems. For instance, Shor's algorithm [16] demonstrated the ability to factorize large numbers efficiently, leading to the collapse of security in public-key cryptography schemes such as RSA. Grover's algorithm [7], on the other hand, reduces the time complexity of cryptographic functions such as symmetric key encryption and hash functions.

As quantum computing technology progresses further, there is numerous research underway to assess the security of current cryptographic systems. These studies involve implementing cryptographic systems as quantum circuits and estimating quantum attack costs to assess security strength. Accordingly, the National Institute of Standards and Technology (NIST) is actively hosting a competition for post-quantum cryptography (PQC) and has established quantum security levels that span from 1 to 5 [13,14]. Levels 1, 3, and 5 correspond

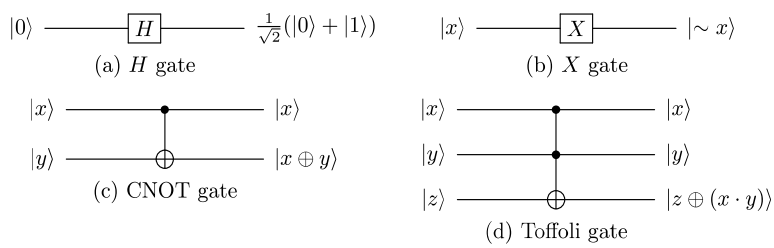
to the cost of a Grover's key attack on AES-128, AES-192, and AES-256 quantum circuits, respectively. Levels 2 and 4 pertain to the collision attack cost on SHA-2 and SHA-3 hash function quantum circuits, which have not yet been defined (with only classical costs reported). In the future, all post-quantum hash functions will be required to estimate the cost of collision attacks to align with Levels 2 and 4.

In this paper, we propose a depth-optimized quantum circuit of Lightweight Secure Hash (LSH) [11], which is a hash function included as validation subjects in the Korean Cryptographic Module Validation Program (KCMVP). Additionally, based on the quantum circuit, we estimate the cost of collision attack using Grover algorithm for LSH.

2 Background

2.1 Quantum Gates

In this section, we explain the quantum gates to implement our quantum circuit. The Hadamard gate creates superposition states of qubits. The X gate, also known as the Pauli-X gate, operates on a single qubit. It can invert the state of a qubit, transforming $|0\rangle$ state to $|1\rangle$ and $|1\rangle$ state to $|0\rangle$. This operation is similar to the classical NOT gate in traditional computing. The CNOT (Controlled-NOT) gate uses two qubits and performs a NOT operation on the target qubit if the control qubit is in the state $|1\rangle$. If the control qubit is in the $|0\rangle$ state, the target qubit remains unchanged. The Toffoli gate, also known as the CCNOT gate (Controlled-Controlled-NOT), uses two control qubits and one target qubit. The Toffoli gate performs a NOT operation on the target qubit only if both control qubits are in the state $|1\rangle$. Thus, it is similar to the classical AND operation. The Toffoli gate can be decomposed into a combination of gates such as H, CNOT and T gates.



(a) H gate (b) X gate (c) CNOT gate (d) Toffoli gate

Fig. 1: Quantum gates

2.2 The Grover algorithm

The Grover algorithm can find a solution for the n-qubit data (in a superposition state) with a complexity of $O(\sqrt{2^n})$ (i.e., a square root speedup compared to classical search of $O(2^n)$). As such, Grover's algorithm has an advantage in solving problems with high search complexity. Notably, extensive research has been conducted on Grover's algorithm for block ciphers and hash functions [8,10,15]. We explain Grover's algorithm through its use in pre-image attacks for hash functions. Grover's algorithm consists of three major processes; *Input setting, Oracle, Diffusion operator.*

Input setting. H gates are used to prepare an n-qubit in a superposition state ($|\psi\rangle$). As a result, an n-qubit input with a superposition state can represents 2^n cases as probabilities.

$$H^{\otimes n}|0\rangle^{\otimes n} = |\psi\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle \tag{1}$$

Oracle The target function (e.g., block ciphers or hash functions) is placed in the oracle and returns the solution using the superposition state of input. To accomplish this, the target function should be implemented using quantum gates (i.e., a quantum circuit). If the quantum circuit finds a solution for the target function (i.e., if $f(x) = 1$), the amplitude of the specific input in a superposition state changes negatively (see Equation 3).

$$f(x) = \begin{cases} 1 \text{ if Hash}(x) = \text{target output} \\ 0 \text{ if Hash}(x) \neq \text{target output} \end{cases} \tag{2}$$

$$U_f(|\psi\rangle|-\rangle) = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle|-\rangle \tag{3}$$

Diffusion operator The diffusion operator enhance the probability for measuring the solution returned by the oracle. Due to the fixed design of the diffusion operator and relatively low complexity compared to the oracle, it is often neglected for the cost estimation in Grover's search [6,10,8,12].

2.3 Quantum Collision Search

Grover's search for an n-bit key of block ciphers or a pre-image of an n-bit hash output for hash functions can be approached straightforwardly, as the complexity of $O(n)$ in classical computing is reduced to $O(\sqrt{n})$ in quantum computing. However, quantum collision search for hash functions is more complicated and can be approached in various ways.

There are various quantum collision attack algorithms using Grover's algorithm. Among them, the BHT algorithm [2] has a query complexity of $O(2^{n/3})$.

However, this algorithm demands a notably large quantum memory, $O(2^{2n/3})$. Also, Bernstein pointed out in [1] that this algorithm is controversial. Considering these aspects, we employ the CNS algorithm [3], which has a query complexity of $O(2^{2n/5})$ and requires only $O(2^{n/5})$ classical memory. Note that the CNS algorithm can be parallelized to reduce the search complexity of $O(2^{n/5})$. By utilizing 2^s quantum instances in parallel, the search complexity for finding collisions is reduced to $O(2^{\frac{2n}{5}-\frac{3s}{5}})$, with $s \leq \frac{n}{4}$. In [9], the authors defined a parallelization strength of $s = n/6$ to estimate the required quantum resources for finding a collision in the SHA-2 and SHA-3 hash functions. Following this approach, we also define a parallelization strength of $s = n/6$ for finding a collision in the LSH hash functions.

2.4 Description of LSH Hash Function

LSH is a Korean cryptographic hash algorithm included among the validation subjects of the KCMVP. LSH consists of LSH-256-224, LSH-256-256, LSH-512-224, LSH-512-256. LSH-512-384 and LSH-512-512. LSH-256-n operates based on a 32-bit word and LSH-512-n operates based on a 64-bit word. LSH operates in three stages: *Initialization*, *Compression*, and *Finalization*.

Initaillization During the *initialization* process, a given input message undergoes one-zero padding. Following this, the padded input message is divided into 32-bit word array messages. Additionally, in the initialization, the 16-bit word array hash chaining variables ($CV^{(i)}$) are set as an initialization vector.

Compression The *compression* function consists of MsgExp and Step (MsgAdd, Mix, and WordPerm) functions. The MsgExp function converts a 32-bit word array message into a 16-bit word array. The MsgExp function process is as follows in Equation 4. The permutation function τ in Equation 4 is defined in Table 1.

$$\begin{aligned} &\mathbf{M}_0^{(i)} \leftarrow (M^{(i)}[0], ..., M^{(i)}[15]),\ \mathbf{M}_1^{(i)} \leftarrow (M^{(i)}[16], ..., M^{(i)}[31]) \\ &\mathbf{M}_j^{(i)} \leftarrow (M_j^{(i)}[0], ..., M_j^{(i)}[15])_{j=2}^{N_s} \\ &M_j^{(i)}[l] \leftarrow M_{j-1}^{(i)}[l] \boxplus M_{j-2}^{(i)}[\tau(l)]\ for\ 0 \leq l \leq 16 \end{aligned} \tag{4}$$

***Step* function** The *step* function is composed of MsgAdd, Mix, and WordPerm functions. The MsgAdd inputs are $CV^{(i)} = T[0], ..., T[15]$ and $M_j^{(i)} = (M_j^{(i)}[0], ..., M_j^{(i)}[15])_{j=2}^{N_s}$. The MsgAdd process is MSGADD(T,M) $\leftarrow (T[0] \oplus M[0], ..., T[15] \oplus M[15])$. The Mix function updates the 16-bit word $T = T[0], ..., T[15]$. In this function, the 16-bit word array T is split into upper eight words and lower eight words, which are then used as input. The operation of the Mix function involves modular addition, XOR, and left rotation. The process of the Mix function is shown in Figure 2 and the the bit rotational amounts using in Mix function

Table 1: The permutation τ and σ

l	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\tau(l)$	3	2	0	1	7	4	5	6	11	10	8	9	15	12	13	14
$\sigma(l)$	6	4	5	7	12	15	14	13	2	0	1	3	8	11	10	9

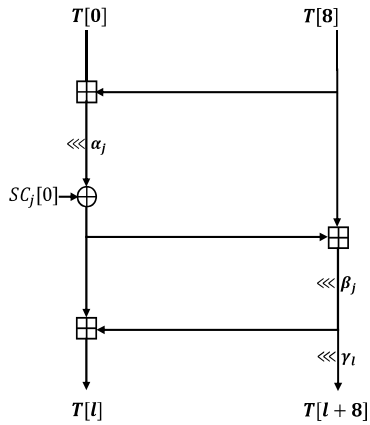


Fig. 2: Mix function

(α_j, β_j and γ_l) are shown in Table 2. The WordPerm function is defined as follows : $X = (X[0], ...X[15]) \leftarrow (X[\sigma(0)], ...X[\sigma(15)])$, where σ is defined by Table 1.

Finallization The finalization function produces an n-bit hash value, denoted as h, obtained from the final chaining variable. The finalization process is as follows:

$$\begin{aligned} \mathbf{h} &\leftarrow (CV^t[0] \oplus CV^t[8], ..., CV^t[7] \oplus CV^t[15]) \\ \mathbf{h} &= (h[0] \,||\, ... \,||\, h[w-1]) \\ h &\leftarrow (h[0] \,||\, ... \,||\, h[w-1])_{[0:n-1]} \end{aligned} \tag{5}$$

3 Quantum Circuit Implementation of LSH

This section describe our quantum circuit implementation of LSH. Our main focus is to optimize the circuit depth for the efficiency of the Grover collision attack. For the sake of simplicity, we primarily focus on explaining LSH-256-256. We set the input length to be equal to the hash length for implementation.

Table 2: Bit rotation amounts: α_j, β_j and γ_l

Algorithm	j	α_j	β_j	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7
LSH-256-n	even	29	1	0	8	16	24	24	16	8	0
	odd	5	17								
LSH-512-n	even	23	59	0	16	32	48	8	24	40	56
	odd	7	3								

3.1 Quantum adder for Optimizing the Depth

To implement the MsgExp function and Mix function, we use a quantum adder. Quantum adders can indeed be designed in various ways, and the choice depends on optimization techniques. Commonly used types of quantum adders include the ripple-carry adder (RCA) and the carry-lookahead adder (CLA).

The RCA adder operates in a sequential manner, where it calculates the carry-out from the previous stage before proceeding with the addition in the next stage. This sequential operation leads to a high depth of the adder, as each stage depends on the carry-out from the previous stage.

On the contrary, the CLA adder accelerates addition by pre-computing carry values for each stage. It adds extra circuits to calculate carry values in advance, determining whether a carry will occur at each stage. This pre-calculation is processed in parallel, speeding up the overall addition process and reducing the depth of the quantum circuit.

A previous work used a Cuccaro adder [4], an improved ripple-carry adder. This adder is implemented in-place operation and requires only one ancilla qubit, $(2n-3)$ Toffoli gates, $(5n-7)$ CNOT gates, and achieves a circuit depth of $(2n+2)$.

In our case, we utilize a Draper adder [5], which is a carry-lookahead adder. This adder can be implemented both in-place and out-of-place. Table 3 compares the resource estimation for 32-bit adders used in LSH-256-n application. Table 3 shows that the out-of-place Draper adder has about half the depth compared to the in-place adder but requires 32-bit output qubits for each adder. With a total of 1024 adders, 32,768 (1024×32) qubits are garbage qubits. Hence, to avoid this inefficiency, we opt for the in-place adder, which slightly increases the depth but significantly reduces the number of qubits required. By using Draper in-place adders, we can reuse all ancilla qubits (53 qubits) except for the input and output qubits in other operations. Although it involves a higher depth than the out-of-place adder, we compensate for this drawback by using adders in parallel within each function(described in Section 3.2 and Section 3.3). This allows us to conserve 32,768 qubits instead of allowing for a higher depth of about 4,134.

3.2 Parallel addition of MsgExp and Mix Functions

In the MsgExp function, 16 adders are needed to update $\mathbf{M}_j^{(i)}$. According to Section 3.1, we can initially allocate 53 ancilla qubits and reuse them throughout.

Table 3: Comparison of quantum resources required for adder (32-bit).

Adder	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit (reuse)	Depth
Cuccaro [4]	in-place	153	61	61	65 (1)	66
Draper [5]	in-place	123	254	22	117 (53)	28
	out-of-place	94	127	11	118 (22)	14

※: Estimation of undecomposed resources

Table 4: Comparison of quantum resources required for each component.

Function	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth
MsgExp	Sequential	1,968	4,064	352	1,077	433
	Parallel	1,968	4,064	**22**	1,872	**28**
Mix	Sequential	2,952	6,096	528	565	649
	Parallel	2,952	6,096	**66**	936	**84**

※: Estimation of undecomposed resources

However, in this scenario, the adders are executed sequentially, increasing the depth of the circuit. To optimize the circuit depth which is our purpose, we employ addition in parallel by allocating more ancilla qubits. To process 16 adders in parallel in the MsgExp function, 848 (16×53) ancilla qubits are required.

Similarly, in the Mix function, 24 (8×3) adders are used and 8 out of the 24 adders can be operated simultaneously. In other words, 8 adders can be processed in parallel, and this parallel addition is repeated a total of 3 times. In this scenario, the ancilla qubits used in the MsgExp function can be reused. Therefore, there is no need to allocate additional ancilla qubits for the adders in the Mix function. As a result, 848 ancilla qubits are initially allocated at once. However, due to the reuse of qubits, the depth may increase (the description continues in Section 3.3).

Table 4 shows the comparison of quantum resources required for MsgExp and Mix function. The parallel operations greatly reduce the toffoli depth and full detph compared to the sequential operations.

3.3 Combined Architecture of Compress Function

Within the Compression function, the MsgExp function, and the Mix function can operate independently. However, due to the ancilla qubit reuse in the Mix function, these functions cannot operate independently. While this architecture can reduce the number of qubits, it increases the circuit depth due to the sequential operations of high complexity. To optimize the circuit depth, we execute the MsgExp function and Mix function in parallel by allocating additional ancilla qubits as shown in Figure 3. This parallel execution method allows us to effectively reduce the overall circuit depth, improving efficiency.

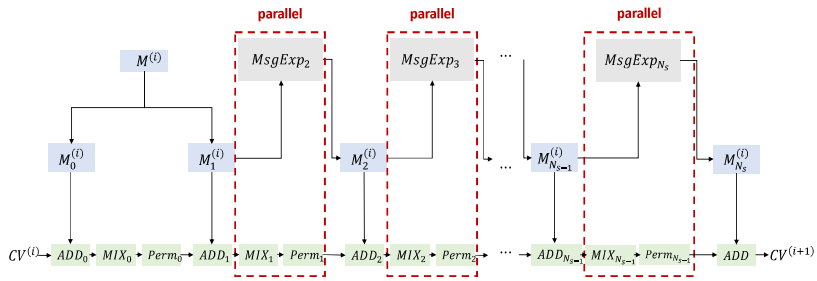


Fig. 3: Parallel process of Compression function

In previous work [17], Song et al. conducted sequential operations in the Compression function as shown in Figure 4. In contrast, our proposed circuit reduces the depth compared to previous work by allocating additional ancilla qubits and implementing the Mix and MsgExp functions in parallel. Specifically, the i-th Mix function and the $i+1$-th MsgExp function can execute in parallel, effectively reducing the circuit depth. Figure 5 shows our proposed Compression function. To enable this parallel process, we additionally allocate 424 (8×53) ancilla qubits for Mix function. Thus, we initially allocate 1,272 ancilla qubits at once and reuse them each round. Algorithm 1 represents the overall process of Compress function. By allocating two sets of ancilla qubits, we can parallelize the even-round Mix function with the odd-round MsgExp function, and the odd-round Mix function with the even-round MsgExp function.

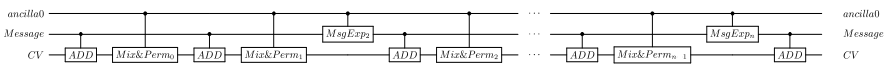


Fig. 4: Compression function in [17] using a sequential process

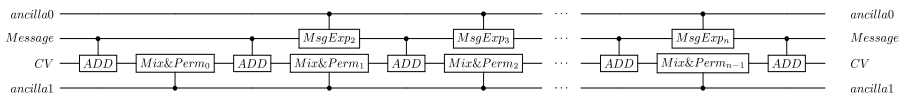


Fig. 5: Proposed parallel Compression function architecture

Table 5 shows the comparision of quantum resources required for Compression function. In parallel process, only the depths of the Mix functions are estimated because they have a higher depth compared to the MsgExp functions. Consequently, the process of the Compression and the Mix functions in parallel demonstrates lower depth compared to processing them sequentially.

Algorithm 1: Quantum circuit implementation of Compress function.

Input: M_{even}, M_{odd} CV, α, β, SC, $ancilla_0$, $ancilla_1$
Output: M_{even}, M_{odd}, CV, 424 qubit array-$ancilla_0$, 848 qubit array-$ancilla_1$

1: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$
2: $CV \leftarrow \text{Mix}(CV, \alpha_{even}, \beta_{even}, SC, ancilla_0)$
3: $CV \leftarrow \text{WordPerm}(CV)$

4: $CV \leftarrow \text{MsgAdd}(M_{odd}, CV)$
5: $CV \leftarrow \text{Mix}(CV, \alpha_{odd}, \beta_{odd}, SC, ancilla_0)$ ▷ Parallelization 1
6: $CV \leftarrow \text{WordPerm}(CV)$

7: **for** $1 \leq i \leq 13$ **do**
8: $M_{even} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$ ▷ Parallelization 1
9: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$
10: $CV \leftarrow \text{Mix}(CV, \alpha_{even}, \beta_{even}, SC, ancilla_0)$ ▷ Parallelization 2
11: $CV \leftarrow \text{WordPerm}(CV)$

12: $M_{odd} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$ ▷ Parallelization 2
13: $CV \leftarrow \text{MsgAdd}(M_{odd}, CV)$
14: $CV \leftarrow \text{Mix}(CV, \alpha_{odd}, \beta_{odd}, SC, ancilla_0)$ ▷ Parallelization 1
15: $CV \leftarrow \text{WordPerm}(CV)$
16: **end for**

17: $M_{even} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$ ▷ Parallelization 1
18: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$

19: return CV

4 Performance & Evaluation

In this section, we provide an estimated quantum resources and the costs of Grover collision attack of our LSH quantum circuit implementation comparing the previous work. We utilized ProjectQ as our quantum programming tool for circuit implementation and simulation. Our implementation was verified using the `ClassicalSimulator` library, and we analyzed the quantum resource using the `ResourceCounter`. For LSH-256-n, the only differences lie in the constant value and the hash length, while the overall operation remains identical. Therefore, all estimated resources, excluding X gates, remain the identical. Similarly, the same applies to LSH-512-n. Thus, we will only compare LSH-256-256 and LSH-512-512.

Table 6 shows the comparison of the decomposed quantum resources required for implementations of LSH. In [17], the decomposed quantum resources were not provided, so we estimate the quantum resources based on the undecomposed resources provided in the paper.

As shown in Table 6, we can observe that our implementation, which applies Cuccaro adders (the same adder as [17]) and parallelization method utilizes 8

Table 5: Comparison of quantum resources required for the Compression function.

Function	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth
Compression	Sequential	139,776	260,096	2,266	2,384	2,873
	Parallel	139,776	260,096	**1,716**	2,808	**2,198**

※: Estimation of undecomposed resources

more qubits compared to [17]. However, it reduces the full depth by approximately 12,000. Additionally, applying the Draper adder further increases the qubit usage, but it significantly reduces the full depth. To assess the trade-off between qubits and depth, we provide metrics TD-M, FD-M, TD^2-M and FD^2-M. As a result, our proposed quantum circuit achieves the optimized performance across all trade-off metrics.

Based on the estimated resources of the LSH quantum circuit, we can estimate the cost of collision attacks on LSH. To estimate the collision attack cost for LSH, we adopt the CNS algorithm described in Section 2.3. The CNS algorithm has the complexity of $O(2^{\frac{2n}{5}-\frac{3s}{5}})$ $(s \leq \frac{n}{4})$. According to [9], they set $s = \frac{n}{6}$ to define suitable criteria for NIST post-quantum security levels, and we follow that approach. Furthermore, since most of the quantum resources are used in implementing the target cipher in the quantum circuit, the overhead of the diffusion operator can be considered negligible compared to the oracle. Additionally, the Grover oracle consists of LSH quantum circuit twice consecutively. The first constructs the encryption circuit, and the second operates the encryption circuit in reverse to return to the state before encryption. As a result, the oracle necessitates twice the cost of implementing the quantum circuit, excluding qubits. Consequently, the cost of Grover's search for LSH is approximately $2 \times 2^{(\frac{2n}{5}-\frac{3s}{5})}$ $\times$ Table 6, as shown in Table 7. Since the cost of collision attacks varies depending on the input and output lengths, we present the resource costs for all LSH parameters.

Table 6: Quantum resources required for implementations of LSH.

Cipher	Source	#CNOT	#1qCliff	#T	Toffoli depth (TD)	#Qubit (M)	Full depth (FD)	TD-M	FD-M	TD^2-M	FD^2-M
LSH-256-256	[17]	545,536	187,813	437,248	6,283	1,552	50,758	$1.16 \cdot 2^{23}$	$1.17 \cdot 2^{26}$	$1.78 \cdot 2^{35}$	$1.82 \cdot 2^{41}$
	Ours-CDKM	545,536	187813	437,248	**4,758**	1,560	**38,483**	$\mathbf{1.77 \cdot 2^{22}}$	$\mathbf{1.79 \cdot 2^{25}}$	$\mathbf{1.03 \cdot 2^{35}}$	$\mathbf{1.05 \cdot 2^{41}}$
	Ours-Draper	1,700,608	306,947	1,820,672	**1,716**	2,808	**13,647**	$\mathbf{1.15 \cdot 2^{22}}$	$\mathbf{1.14 \cdot 2^{25}}$	$\mathbf{1.93 \cdot 2^{32}}$	$\mathbf{1.90 \cdot 2^{38}}$
LSH-512-512	[17]	1,203,760	418,369	966,000	13,875	3,088	111,532	$1.28 \cdot 2^{25}$	$1.28 \cdot 2^{28}$	$1.08 \cdot 2^{39}$	$1.09 \cdot 2^{45}$
	Ours-CDKM	1,203,760	418,369	966,000	**10,500**	3,096	**84,451**	$\mathbf{1.94 \cdot 2^{24}}$	$\mathbf{1.95 \cdot 2^{27}}$	$\mathbf{1.24 \cdot 2^{38}}$	$\mathbf{1.26 \cdot 2^{44}}$
	Ours-Draper	4,030,000	736,569	2,614,473	**2,028**	5,832	**17,385**	$\mathbf{1.41 \cdot 2^{23}}$	$\mathbf{1.51 \cdot 2^{26}}$	$\mathbf{1.40 \cdot 2^{34}}$	$\mathbf{1.60 \cdot 2^{40}}$

Table 7: Costs of the Grover's collision search for LSH.

Cipher	#Gate (G)	Full depth (FD)	T-depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
LSH-256-224	$1.65 \cdot 2^{89}$	$1.5 \cdot 2^{81}$	$1.51 \cdot 2^{80}$	$1.72 \cdot 2^{48}$	$\mathbf{1.23 \cdot 2^{171}}$	$1.29 \cdot 2^{130}$	$1.3 \cdot 2^{129}$	$1.95 \cdot 2^{211}$	$1.97 \cdot 2^{209}$
LSH-256-256	$1.25 \cdot 2^{99}$	$1.13 \cdot 2^{91}$	$1.14 \cdot 2^{90}$	$1.08 \cdot 2^{54}$	$\mathbf{1.42 \cdot 2^{190}}$	$1.23 \cdot 2^{145}$	$1.24 \cdot 2^{144}$	$1.41 \cdot 2^{236}$	$1.42 \cdot 2^{234}$
LSH-512-224	$1.96 \cdot 2^{90}$	$1.91 \cdot 2^{81}$	$1.78 \cdot 2^{80}$	$1.79 \cdot 2^{49}$	$\mathbf{1.87 \cdot 2^{172}}$	$1.71 \cdot 2^{131}$	$1.6 \cdot 2^{130}$	$1.64 \cdot 2^{213}$	$1.43 \cdot 2^{211}$
LSH-512-256	$1.49 \cdot 2^{100}$	$1.45 \cdot 2^{91}$	$1.35 \cdot 2^{90}$	$1.13 \cdot 2^{55}$	$\mathbf{1.07 \cdot 2^{192}}$	$1.64 \cdot 2^{146}$	$1.53 \cdot 2^{145}$	$1.18 \cdot 2^{238}$	$1.03 \cdot 2^{236}$
LSH-512-384	$1.96 \cdot 2^{138}$	$1.91 \cdot 2^{129}$	$1.78 \cdot 2^{128}$	$1.42 \cdot 2^{76}$	$\mathbf{1.87 \cdot 2^{268}}$	$1.36 \cdot 2^{206}$	$1.27 \cdot 2^{205}$	$1.3 \cdot 2^{336}$	$1.13 \cdot 2^{334}$
LSH-512-512	$1.29 \cdot 2^{177}$	$1.26 \cdot 2^{168}$	$1.17 \cdot 2^{167}$	$1.79 \cdot 2^{97}$	$\mathbf{1.63 \cdot 2^{345}}$	$1.13 \cdot 2^{266}$	$1.05 \cdot 2^{265}$	$1.43 \cdot 2^{434}$	$1.24 \cdot 2^{432}$

5 Conclusion

In this work, we focused on optimizing the depth of quantum circuits for the Korean cryptographic hash function LSH. To optimize the depth, we use optimized quantum adders and parallelization. Our quantum circuit implementation of LSH achieves a significant improvement in depth over 78.8% compared to the approach presented in [17]. Additionally, the Toffoli depth sees an enhancement of more than 79.1%.

Through the depth-optimized implementation, we also obtain the optimized quantum resources of Grover collision attack for LSH. Although NIST provide the post-quantum security level and quantum attack costs for symmetric key ciphers, they do not provide the specific quantum cost for hash functions. If NIST defines criteria for hash functions, we will compare our results with those criteria.

6 Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT).(No. RS-2023-00277994, Quantum Circuit Depth Optimization for ARIA, SEED, LEA, HIGHT, and LSH of KCMVP Domestic Cryptographic Algorithms, 90%) and this work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (<Q|Crypton>, No.2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 10%)

References

1. Bernstein, D.J.: Cost analysis of hash collisions: Will quantum computers make sharcs obsolete. SHARCS **9**, 105 (2009) 4
2. Brassard, G., Hoyer, P., Tapp, A.: Quantum algorithm for the collision problem. arXiv preprint quant-ph/9705002 (1997) 3
3. Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An efficient quantum collision search algorithm and implications on symmetric cryptography. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory

and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23. pp. 211–240. Springer (2017) 4
4. Cuccaro, S., Draper, T., Kutin, S., Moulton, D.: A new quantum ripple-carry addition circuit. arXiv (2008), https://arxiv.org/pdf/quant-ph/0410184.pdf 6, 7
5. Draper, T.G., Kutin, S.A., Rains, E.M., Svore, K.M.: A logarithmic-depth quantum carry-lookahead adder. arXiv preprint quant-ph/0406142 (2004) 6, 7
6. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's algorithm to AES: Quantum resource estimates. In: Takagi, T. (ed.) Post-Quantum Cryptography. pp. 29–43. Springer International Publishing, Cham (2016) 3
7. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996) 1
8. Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., Chattopadhyay, A.: Quantum analysis of AES. Cryptology ePrint Archive, Paper 2022/683 (2022), https://eprint.iacr.org/2022/683, https://eprint.iacr.org/2022/683 3
9. Jang, K., Lim, S., Oh, Y., Kim, H., Baksi, A., Chakraborty, S., Seo, H.: Quantum implementation and analysis of sha-2 and sha-3. Cryptology ePrint Archive (2024) 4, 10
10. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing Grover Oracles for quantum key search on AES and LowMC. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 280–310. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_10, https://doi.org/10.1007/978-3-030-45724-2_10 3
11. Kim, D.C., Hong, D., Lee, J.K., Kim, W.H., Kwon, D.: Lsh: A new fast secure hash function family. In: Information Security and Cryptology-ICISC 2014: 17th International Conference, Seoul, South Korea, December 3-5, 2014, Revised Selected Papers 17. pp. 286–313. Springer (2015) 2
12. Liu, Q., Preneel, B., Zhao, Z., Wang, M.: Improved quantum circuits for AES: Reducing the depth and the number of qubits. Cryptology ePrint Archive, Paper 2023/1417 (2023), https://eprint.iacr.org/2023/1417, https://eprint.iacr.org/2023/1417 3
13. NIST.: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf 1
14. NIST.: Call for additional digital signature schemes for the post-quantum cryptography standardization process (2022), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf 1
15. Rahman, M., Paul, G.: Grover on katan: Quantum resource estimation. IEEE Transactions on Quantum Engineering **3**, 1–9 (2022) 3
16. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. IEEE (1994) 1
17. Song, G., Jang, K., Kim, H., Seo, H.: A parallel quantum circuit implementations of lsh hash function for use with grover's algorithm. Applied Sciences **12**(21), 10891 (2022) 8, 9, 10, 11

Who ruins the game?: unveiling cheating players in the “Battlefield" game

Dong Young Kim and Huy Kang Kim

School of Cybersecurity, Korea University, Republic of Korea
{klgh1256, cenda}@korea.ac.kr

Abstract. The “Battlefield" online game is well-known for its large-scale multiplayer capabilities and unique gaming features, including various vehicle controls. However, these features make the game a major target for cheating, significantly detracting from the gaming experience.
This study analyzes user behavior in cheating play in the popular online game, the “Battlefield", using statistical methods. We aim to provide comprehensive insights into cheating players through an extensive analysis of over 44,000 reported cheating incidents collected via the “Gametools API". Our methodology includes detailed statistical analyses such as calculating basic statistics of key variables, correlation analysis, and visualizations using histograms, box plots, and scatter plots. Our findings emphasize the importance of adaptive, data-driven approaches to prevent cheating plays in online games.

Keywords: Cheating Play Detection · Online Game· The Battlefield· Statistical Method · Correlation Analysis · Visualization

1 Introduction

The Battlefield game [3] (operated by Electronic Arts Inc.) is a large-scale team multiplayer online game. It is also well-known for its unique gaming features, such as enabling various vehicle controls (*e.g.*, Tank, Armored Personnel Carrier, Infantry Fighting Vehicle, Fighter Jet, and Attack Helicopter) by game players. Such an aspect distinguishes it from other First-Person Shooters (FPS) games, attracting gamers worldwide. As the same as the real-world battlefield, the vehicle’s ability is too strong enough to eliminate the opponent player’s infantry troops. To leverage vehicles’ ability, some players use game cheating programs that are regarded as unfair gameplay.

Cheating play is an axis of evil in online gaming. If game companies fail to detect cheating play in time, the game users quit and leave the game because they can feel unfair. Consequently, cheating play detection is a critical success factor in preventing a game company’s financial damage and managing the user’s loyalty.

Not only in the Battlefield game, but cheating programs such as Aimbot, Wallhack, Speedhack, Weaponhack, and EXP hack have also been widely used in FPS games for several decades [5].



Fig. 1. The BattleField in-game cheating play example[1]

Thus, like the other FPS games, the Battlefield game has many users complaining about the cheating play. Fig. 1 shows a cheating player use the Wallhack and ESP cheating tool during the battle. In response, the latest version, the Battlefield 2042, deploys an Anti-Cheat program on the client-side. However, detecting cheating on the client-side may diminish user experience due to program errors and vulnerabilities. Detecting cheating over the network is practically impossible, as it requires real-time monitoring of all users' communications, which could degrade the overall gaming experience due to the real-time nature of FPS games. Server-side detection demands significant resources from game servers. It is practically impossible for game companies to monitor all users' logs, which could increase the game's latency, which is critical in FPS games. Thus, the existing cheating detection methods used in other game genres (*e.g.*, frequently used data mining or machine learning methods in MMORPG genre [4].) are not easily applicable in FPS games.

Therefore, this paper aims to provide insights for discovering users who use cheating tools without consuming resources from the game servers and clients, based on statistical analysis and visualization.

Our research method employs fundamental statistical analysis, which can effectively distinguish cheating players' behaviors. The main contributions of this study can be summarized as follows.

- Unlike existing cheating detection methods in FPS games, our approach does not need to be deployed in-game client-side. The current countermeasures deployed in the game client-side can harm the overall gaming experience because of the overhead (*i.e.*, typical usability vs. security problem)
- To the best of our knowledge, it is the first approach for cheating play user detection by lightweight statistical analysis of kill-log in the Battlefield game.

We discovered features that can distinguish cheating players' behaviors successfully.

2 Background

2.1 How to play the Battlefield Games?

The Battlefield game provides specialized game modes. For instance, some modes focus on the number of infantry kills to win, while others modes focus on the score based on the result of capturing specific areas during the combat. Among various modes, one of the most representative modes is the 'Conquest mode'. Conquest mode involves 64 or 128 players, where participants are divided into several classes, such as Assault, Support, Ammunition, and Recon. Each class has specialized equipment. For example, the Support class users, one of the infantry solder classes in the game, carry a medical kit to heal their wounded teammates, or they carry a defibrillator to revive soldiers who have been downed.

Depending on the map size and scale, these soldiers can control various vehicles. These vehicles include Tanks, Armored Personnel Carriers, Infantry Fighting Vehicles, Anti-Aircraft Vehicles, Fighter Jets, and Attack Helicopters. There are land, water, and air vehicles that players can use from the initial starting point or specific points on the map. The amount of damage dealt by the weapons on these vehicles is significantly higher than that of infantry weapons. Consequently, vehicles can massacre infantry, or infantry may find themselves hiding from the vehicles.

Conquest mode involves infantry using vehicles to capture specific points. Capturing a point occurs when there are more allied soldiers than enemies at a location for a certain period, and it happens faster with more allies present.

The Conquest mode is the most preferable play mode, but cheating play exists in all modes, not only the Conquest mode.

2.2 Definition and Types of Cheating in the Game

The well-known cheating plays in FPS are Aimbot, Damage Modify, Gadget Modify, Wallhack, Stealth, and Magical Bullet. 'Aimbot' is a program that artificially enhances a player's shooting accuracy by automatically targeting opponents' critical areas (*e.g.*, headshot), enabling precise shooting. This type of cheating elevates a player's shooting abilities, and eventually disrupting the game's balance seriously. 'Damage Modify' is cheating that manipulates a player's weapon to cause more damage than usual, allowing players to overpower enemies with fewer bullets. 'Gadget Modify' alters or enhances the functions of the equipment or gadgets players use. For example, it could increase the blast radius of explosives or reduce the cooldown period of gadgets. 'Wallhack' makes walls or other obstacles transparent or allows players to see the outlines of enemies through them. It provides a significant tactical advantage by predicting the location and movements of concealed enemies. 'Stealth' renders players invisible to enemy sights or detection systems, granting the ability to move covertly

and disrupt game balance. 'Magical Bullet' includes features that allow players' bullets to penetrate obstacles or automatically target enemies at unrealistically long distances, ignoring ballistics and physical laws, resulting in highly unfair gameplay.

2.3 Why Cheating Detection in FPS is Important?

Why is it necessary to prevent cheating in FPS games? First, it is essential to maintain fairness. Ensuring the game's fairness allows all players to compete under equal conditions, making the gaming experience more enjoyable and satisfying. Second, it is to ensure that all users can enjoy the game. One of the main purpose of playing a game is to have fun and feel a sense of accomplishment through challenge. Fun and challenge are key rewards in a game system. If the fun and challenge based reward system is disrubted by some cheating players, then many normal players will quit and leave the game. Third, it disturbs the growth and development of players. Honest players can develop their skills and learn strategies, whereas cheating players skip this developmental process, negatively impacting other players (*i.e.*, contagion) [14]. Thus, if cheating players gain an advantage in the game, honest players are disadvantaged, leading to many players leaving the game. For these reasons, cheating can ruin the game's overall fair-play culture and lead to user churn.

3 Related Work

Tekofsky *et al.* [12] analyzed play styles by age in Battlefield 3. Their study used data from 10,416 users to demonstrate that performance and speed of play decrease with age. May *et al.* [8] analyzed how the game Battlefield 2042 implements climate crises and environmental challenges. This study investigated how the game's settings and scenarios encourage players to think about ecological issues and reviewed how user-generated content and discussions about the game promote ecological engagement. They demonstrated through data analysis from 13,129 users how these elements stimulate ecological thinking among players.

Various studies have been conducted on detecting cheating in FPS games. Most of these studies use computer vision to detect changes in status or visually apparent cheating patterns. For example, Nie *et al.* [9] identified cheating patterns such as Aimbots in FPS games using a deep learning vision framework called VADNet. Jonnalagadda *et al.* [6] proposed a new vision-based approach that captures the final game screen state to detect illegal overlays. They collected data from two FPS games and several cheating software using deep learning models to verify cheat detection performance. Liu *et al.* [7] detected Aimbots using cosine similarity and statistical methods. Cosine similarity was used to measure the change in aim when players first spot an enemy, calculating the angle between vectors of the player's aiming direction to compare the automatic aiming patterns of Aimbots with those of regular players. Pinto *et al.* [10] proposed a novel approach using deep learning and multivariate time-series analysis to

detect cheats in video games. Instead of game data, they analyzed user behavior data such as key inputs and mouse movements, demonstrating high accuracy in cheat detection. Han *et al.* [5] discovered several features (*e.g.*, win-rate, headshot-ratio, play counts and play time, etc.) and detection rules to differentiate cheating players in Point Blank, one of the famous FPS games in Asian game market. In this work, the proposed method detected cheating plays based on the analysis of action logs, while our proposed model is based on the analysis of kill-log and stat data.

4 Methodology

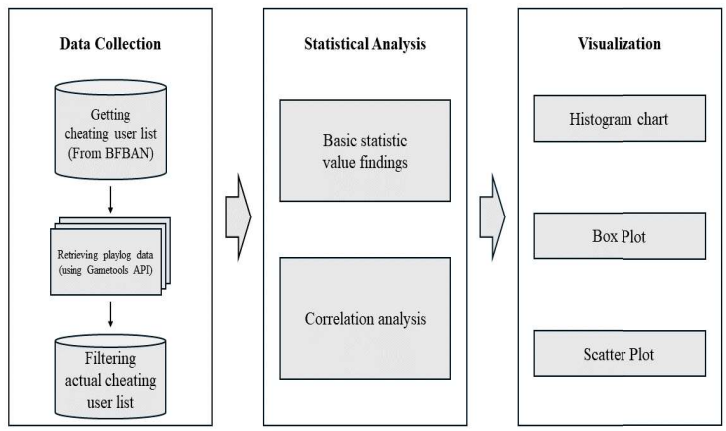


Fig. 2. Overall procedure of FPS cheating play detection

Fig. 2 shows the overall procedure of the proposed method. We aimed to conduct an in-depth analysis of cheating players using statistical analysis. In this process, we calculated basic statistics such as mean, standard deviation, and minimum/maximum values for each variable. Then we extracted meaningful variables based on these calculations. This approach allowed us to understand the overall characteristics of the data and identify key factors.

First, to understand the data distribution, we calculated basic statistics for each variable, including mean, standard deviation, minimum value, maximum value, median, and quartiles. This helped us grasp the central tendency and variability of the data and check for outliers or anomalies.

Second, we analyzed the correlations between variables. Then, we identified the relationships between variables through correlation analysis; and we determined which variables have influence value to others. This provided crucial information for understanding the interactions between variables. By measuring correlation coefficients, we identified variables that showed significant correlations.

Finally, for the data visualization, we created various graphs such as histograms, box plots, and scatter plots. Histograms allowed us to visually confirm the distribution of each variable, while box plots provided a clear view of the quartiles, median, and outliers. Scatter plots were used for visually expressing the relationships between two variables, making it easier to understand correlations or trends between variables.

By visually representing the results of complex statistical analyses, we could easily identify key characteristics and patterns in the data. Additionally, using various graphs to illuminate different aspects of the data enabled a more comprehensive and in-depth analysis.

5 Experiment Result

5.1 Dataset

Due to the absence of publicly available datasets for experimenting with the methodology of this paper, we collected the data directly. The dataset was gathered using the "Gametools API" [1] with the Requests [11] library. The "Gametools API" allows us to fetch the list and information of users identified as cheating players from "BFBAN" [2]. "BFBAN" is a site where users can report cheating players and check the status of their bans online. We specifically retrieved the list of users banned by the game company and analyzed their statistics.

The data was collected in two stages. First, we collected the list of cheating players from the "Gametools API". Second, we gathered detailed information about these users' cheating activities using the same API.

The total number of cheating players is 119,569. The data spans from the first reported cheating player on October 27, 2018, to the most recent report on April 29, 2024. However, due to some data being missing during the collection process from the "Gametools API", we were only able to collect data for 44,307 users.

5.2 Explanation of Key Variables

The variables *hack_score* and *hack_score_current* indicate the extent of a user's hacking activities. The *hack_score* represents the highest value of either *hack_score_current* or the player's highest historic *hack_score*, while the *hack_score_current* represents the *hack_score* of the player's current stats. Hack Score measures how abnormal and hacker-like a player's stats are. When the *hack_score* is more than 100, that means their stats are highly abnormal and can only be produced by hacks (except rare false positives. Refer to the dataset's description; the false positive ratio is 0.5% in total.). A *hack_Score* from 0-100 is calculated for each weapon the player has used. The *stats.kills* variable represents the number of opponents a user has eliminated in the game. The *stats.kpm* shows how many opponents the user kills per minute in the game.

The *stats.rank* indicates the user's level in the game, with higher values showing that the user has gained more experience points. The *unreleased_ weapons.kills* represents the number of kills a user has made using weapons that have not yet been officially released, indicating kills made with special or test weapons.

5.3 Basic Statistical Analysis

The dataset consists of 44,307 observations and includes various variables. Basic statistics, such as mean, standard deviation, minimum, maximum, median, and quartiles, were calculated for key variables (hack_score, hack_score_current, stats.kills, stats.kpm, stats.rank, unreleased_weapons.kills). These calculations allowed us to understand the central tendency and variability of the data and identify any outliers or anomalies. The whole result of the basic analysis is summarized in Table 1.

Column	Mean	Std	Min	Q1: 25%	Q2: 50%	Q3: 75%	Max
hack_score	477.45	736.96	0	41.75	143.0	528.25	16667.0
hack_score _current	473.89	732.86	0	41.75	143.0	528.25	16667.0
stats.kills	6937.79	14906.77	0	527.25	2958.5	7826.5	781157.0
stats.kpm	1.13	6.74	0	0.33	0.64	1.02	1488.56
stats.rank	65.45	46.87	0	22.0	61.0	109.0	154.0
unreleased _weapons. kills	6937.79	14906.77	0	527.25	2958.5	7826.5	781157.0

Table 1. Summary Statistics of Key Variables

5.4 Correlation Analysis Between Variables

The hack_score and hack_score_current showed a high correlation of 0.996, indicating that users who have cheated once tend to continue cheating. The stats.kills, and stats.rank showed a strong correlation of 0.794, meaning that recording a high number of kills increases the likelihood of reaching higher levels. The hack_score and stats.rank showed a moderate correlation of 0.449, suggesting that the relationship between cheating and level is not significant. This result indicates that low-level users are not necessarily more prone to cheating and high-level users are not guaranteed to refrain from cheating. The stats.kills and unreleased_weapons.kills showed a perfect positive correlation, indicating that users who use unreleased weapons (cheating players) have a 100% probability of recording a high number of kills. This suggests that some cheating players use these cheats to generate high number of kills, adversely affecting innocent opponents. The whole result of the correlation analysis is summarized in Table 2.

	hack_score	hack_score_current	stats.kills	stats.kpm	stats.rank	unreleased_weapons.kills
hack_score	1.000000	0.996151	0.385814	0.015135	0.449039	0.385814
hack_score_current	0.996151	1.000000	0.379819	0.015231	0.442342	0.379819
stats.kills	0.385814	0.379819	1.000000	0.004692	0.793550	1.000000
stats.kpm	0.015135	0.015231	0.004692	1.000000	-0.006504	0.004692
stats.rank	0.449039	0.442342	0.793550	-0.006504	1.000000	0.793550
unreleased_weapons.kills	0.385814	0.379819	1.000000	0.004692	0.793550	1.000000

Table 2. Correlation Matrix between key features

5.5 Data Visualization

We visualized the result using histograms, box plots, and scatter plots.

The histogram chart is used to visually confirm the distribution of variables and identify data concentrated in specific ranges and outliers. The Histogram chart is shown in Fig 3.

The box Plot is used to visually analyze the median, quartiles, and outliers of the variables to understand the overall data distribution. The Box Plot result is presented in Fig 4.

The scatter Plot is used to visually express the relationship between hack score and other variables, making it easy to understand correlations and trends between variables. The Scatter Plot result is presented in Fig 5.

The key findings from the visualization process are as follows.

- Most benign users have lower hack_score than the median value. It indicates that most users do not use cheating programs.
- We observe the hack_score and hack_score_current have a similar distribution. That means a user who does a cheating play tends to keep on doing cheating play continuously.
- We find the cheating player's records do not show a higher kill count (stats.kills). On the one hand, it means the cheating program is not always effective for all opponent users (some high-level human players can still overwhelm the cheating players.) On the other hand, it also mean the cheating tool can limit the kill ratio not to be detected by an in-game monitoring system. The lower kpm value (stats.kpm < 40) supports this point of view. Also, some cheating players use unknown weapons to kill the opponent players.

The proposed method does not require a high-performance computing environment. The simple statistical values can be estimated in near real-time by streaming in-game play logs. In particular, the continuous cheating behavior of users with high hack scores and the perfect correlation between stats.kills and unreleased_weapons.kills can serve as strong indicators for identifying cheating

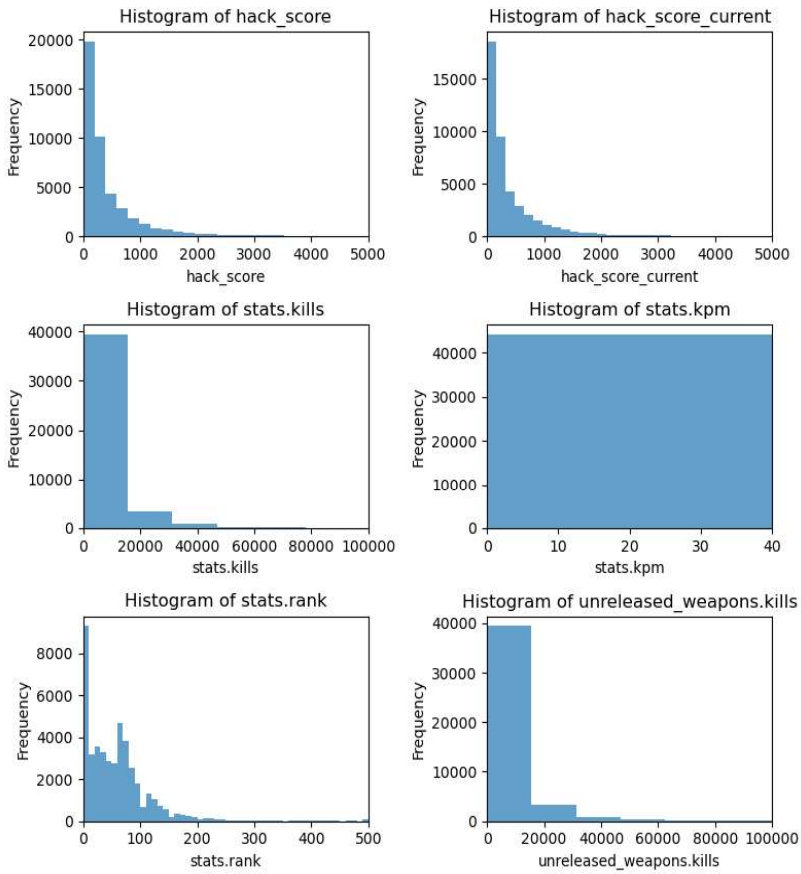


Fig. 3. Histogram of key features

players. These findings can be beneficial for detecting and responding to cheating behavior in real-time within the game.

6 Discussion

We gathered the dataset from the several public sites. and the banned user list can be regarded as reliable. The game company does not provide the official dataset, including banned users, to the public; thus, our approach is the best way so far. However, the dataset does not include a complete cheating player list (*i.e.*, it has a false-negative error) because it relies on the volunteer users' reporting. Also, the dataset can have a false-positive issue when the reporting is intentionally misreported (to dishonor a specific opponent user).

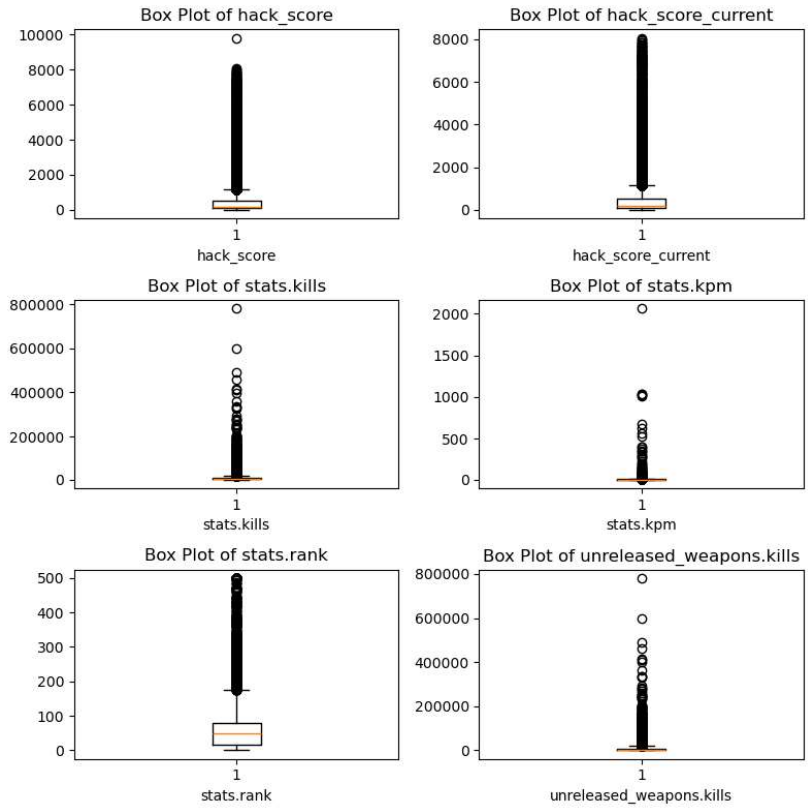


Fig. 4. Box Plot of key features

In this study, we adopt simple methods such as correlation analysis and statistical visualization to detect cheating play. As a result, we identified general patterns of cheating players through the correlation analysis between key variables. In the future, we will explore using more statistical techniques for robust and lightweight cheating play detection. We will consider testing other FPS games to get more generalizations of the proposed method. Also, we will research anomaly detection based on malicious and benign user data to overcome the drawbacks of our methods. Due to the difficulty in obtaining data from normal players, there are limitations in creating an anomaly detection model. We plan to implement an anomaly detection model based on the kill-log of abnormal players and explore this topic further.

7 Conclusion

This study proposed a data-driven approach to address the issue of cheating in the Battlefield. To conclude, the main contributions of this study are as follows:

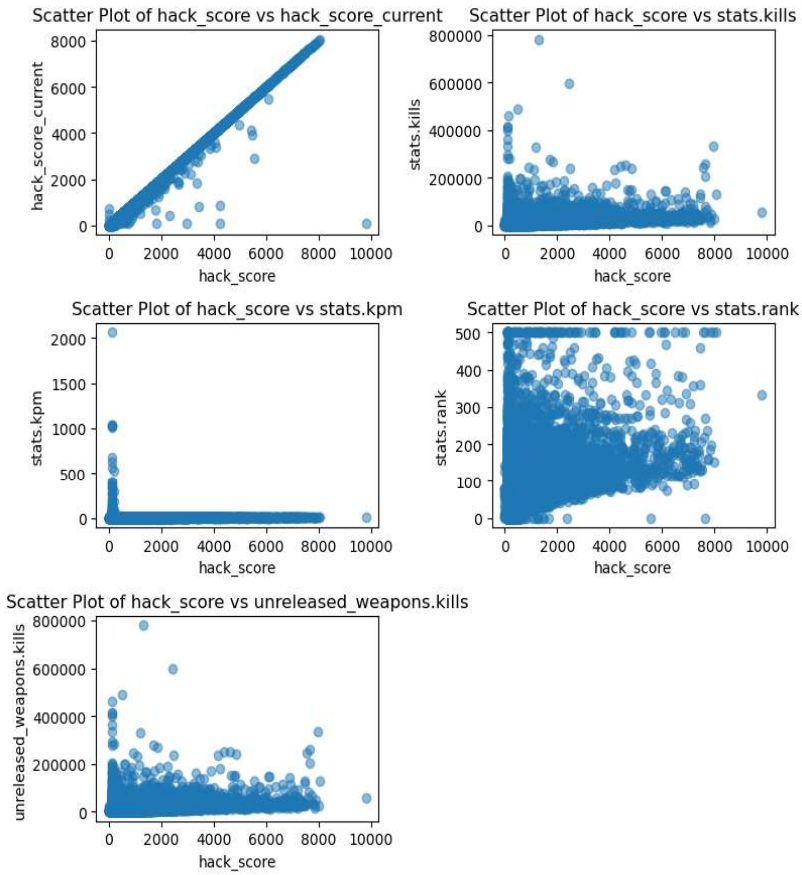


Fig. 5. Scatter between hack_score and other variables

First, we confirm the repeatability of cheating behavior. The high correlation (0.996) between hack_score and hack_score_current indicates that users who start cheating tend to continue cheating. This is a significant finding as it shows that cheating behavior is not a one-time occurrence but follows a repetitive pattern. These results emphasize the need for continuous monitoring and response to cheating players in cheating detection systems.

Second, we discovered that the use of specific weapons could be an indicator of cheating behavior. The perfect correlation (1.000) between stats.kills and unreleased_weapons.kills indicates that cheating players tend to record many kills using unreleased weapons. This suggests that monitoring the use of specific weapons can be an important indicator for detecting cheating. Game developers can use this pattern to detect cheating behavior more effectively.

Last, we showed that cheating behavior can occur at all player levels. The moderate correlation (0.449) between hack_score and stats.rank indicates that cheating players are not necessarily concentrated at high levels. This implies that cheating can frequently occur even at lower levels, highlighting the need for cheating detection across all player levels.

Acknowledgement. This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT). (Grant No. 2020-0-00374, Development of Security Primitives for Unmanned Vehicles).

References

1. API, G.: https://api.gametools.network/docs/, accessed: 2024-08-09
2. BFBAN: https://admin.bfban.com/player, accessed: 2024-08-09
3. EA: Battlefield games. https://www.ea.com/ko-kr/games/battlefield, accessed: 2024-08-09
4. Han, M.L., Kwak, B.I., Kim, H.K.: Cheating and detection method in massively multiplayer online role-playing game: systematic literature review. IEEE Access **10**, 49050–49063 (2022)
5. Han, M.L., Park, J.K., Kim, H.K.: Online game bot detection in fps game. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems-Volume 2. pp. 479–491. Springer (2015)
6. Jonnalagadda, A., Frosio, I., Schneider, S., McGuire, M., Kim, J.: Robust vision-based cheat detection in competitive gaming. Proceedings of the ACM on Computer Graphics and Interactive Techniques **4**(1), 1–18 (2021)
7. Liu, D., Gao, X., Zhang, M., Wang, H., Stavrou, A.: Detecting passive cheats in online games via performance-skillfulness inconsistency. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 615–626. IEEE (2017)
8. May, L., Hall, B.: Thinking ecologically with battlefield 2042. Game Studies **24**(1) (2024)
9. Nie, B., Ma, B.: Vadnet: Visual-based anti-cheating detection network in fps games. Traitement du Signal **41**(1) (2024)
10. Pinto, J.P., Pimenta, A., Novais, P.: Deep learning and multivariate time series for cheat detection in video games. Machine Learning **110**(11), 3037–3057 (2021)
11. Reitz, K.: Requests: Http for humans™. https://requests.readthedocs.io/en/latest/, accessed: 2024-08-09
12. Tekofsky, S., Spronck, P., Goudbeek, M., Plaat, A., van Den Herik, J.: Past our prime: A study of age and play style development in battlefield 3. IEEE Transactions on Computational Intelligence and AI in Games **7**(3), 292–303 (2015)
13. terDwas: Hurricane loader in the battlefield 2042. https://github.com/terDwas/Battlefield-2042-Hurricane-Cheat, accessed: 2024-08-09
14. Woo, J., Kang, S.W., Kim, H.K., Park, J.: Contagion of cheating behaviors in online social networks. IEEE Access **6**, 29098–29108 (2018)



The 25th World Conference on Information Security Applications

WISA 2024

Posters

CONTENTS

The 2024 WISA Poster Session

Poster Session 1 : 15:00 - 16:00, August 22 (Thursday), 2024
Session Chair: Jungsoo Park (Kangnam University)

Session1-1 Proposal of distributed network data management mechanism based on keyword extraction from text content
Jinsu Kim and Namje Park (Jeju National University)

Session1-2 Poster: Blockchain-Based ESG Reporting and Refurbishment Certification in the Cosmetics Industry
Cho Nwe Zin Latt, Muhammad Firdaus and Kyung-Hyune Rhee (Pukyong National University)

Session1-3 Random Public Codebook-Based Double XOR Cipher
Seung-Ha Ji, So-Eun Jeon and Il-Gu Lee (Sungshin Women's University)

Session1-4 Error Correction Code-Based Two-Layer Digital Watermarking Technique
Do-Eun Kim, So-Hyun Park and Il-Gu Lee (Sungshin Women's University)

Session1-5 A Study on Enhancing Reauthentication Methods Among Workloads in a ZTA Environment
Wonhwi Choi and Jungsoo Park (Kangnam University)

Session1-6 Exploring Hybrid Approach-Steganalysis in Digital Forensics Applications
Rini Wisnu Wardhani, Dedy Septono Catur Putranto, Thi-Thu-Huong Le, Agus Mahardika Ari Laksmono, Hwang Yeonjeong and Howon Kim (Pusan National University)

Session1-7 Ultra-high Reliable Multi-AP Cooperation Using Securely Compressed Channel-status Information
So-Eun Jeon and Il-Gu Lee (Sungshin Women's University)

Session1-8 Adversarial Attacks on Behavior Decisions and Countermeasures Using Dropout-Based Fake Node
Seungyeol Lee, Youngju Lee and Jaecheol Ha (Hoseo University)

Session1-9 Blockchain-based secure vehicle data trading model for vehicle accident reconstruction
Su Jin Shin, Youngho Park and Sang Uk Shin (Pukyong National University)

Session1-10 Secure and Low Latency Transmission Technique for Next Generation Wireless LANs
Sun-Jin Lee and Il-Gu Lee (Sungshin Women's University)

Session1-11 Data Preprocessing Technique for Privacy-Preserving Machine Learning
Ye-Rim Jeong, So-Yeon Kim and Il-Gu Lee (Sungshin Women's University)

Session1-12 Mission-centric Cyber Resilience Assessment using Transformer Models in Tactical Modeling and Simulation
Seungwoon Lee, Haerin Kim and Jaesik Kang (LIG Nex1)

Session1-13 Enhancing Cyber Warfare Training: A Cost-Effective Approach to Multi-Host Network Simulations with Container-Based Cyber Range
Haerin Kim and Seungwoon Lee (LIG Nex1)

Session1-14 Defense Mechanism Using Purifier and Detector Against Adversarial Patch Attacks for Unmanned Combat Systems
Dong-U Kang (Korea University), Jaesik Kang and Seungwoon Lee (LIG Nex1)

Session1-15 A Robust Counting Sketch for Data Plane Intrusion Detection
Sian Kim and DaeHun Nyang (Ewha Womans University)

Session1-16 Multi-Link Operation-Based Group Selection Technique for Enhancing Privacy and Efficiency in Federated Learning
So-Yeon Kim, Na-Eun Park, Yu-Ran Jeon and Il-Gu Lee (Sungshin Women's University)

Session1-17 COLA Sketch: Enhancing Intrusion Detection with Collision-Aware Noise Reduction
JiYeon Kim, Sian Kim and DaeHun Nyang (Ewha Womans University)

Session1-18 Clock Glitch-based Fault Injection Attack on Deep Neural Network
Hyoju Kang, Seongwoo Hong and Jaecheol Ha (Hoseo University)

Session1-19 Modular Design Approach in API Security
I Wayan Adi Juliawan Pawana, Youngjae Kim, Bonam Kim and Ilsun You (Kookmin University)

Session1-20 Poster: Impact Assessment of Cooperative Driving Performance because of I/O Latency
Juon Kim(Korea University), Sanghwa Jin, Jongin Lee (Samsung Electronics) and Wonsuk Choi (Korea University)

Session1-21 Proposal of eqSIM Security Toolkit for Private 5G Device Authentication
Hyejin Yoon, Yoonjin Jang, Seunghwan Yun and Okyeon Yi (Kookmin University)

Poster Session 2: 16:00 - 17:00, August 22 (Thursday), 2024
Session Chair: Dong-Chan Kim (Kookmin University)

Session2-1 Proposal of a client consensus mechanism in a blockchain proof environment based on token distribution

Jinsu Kim, Keunbae Choi and Namje Park (Jeju National University)

Session2-2 Better RDP-GAN with A Noisy Generator

JungHye Choi and DaeHun Nyang (Ewha Womans University)

Session2-3 Study on the possibility of hash collisions of OTP values

Tae-Gyeong NA, Hansaem Wi and Okyeon Yi (Kookmin University)

Session2-4 Template-free Jailbreak Red-Teaming for LLM: A Effort-Effective Approach

Donghyun Kim, Mingyu Lee, Yijun Jin, Sang Seo, Hyunpyo Choi and Juhyun Jang (NSHC Inc.)

Session2-5 Empirical Analysis of Cryptocurrency Payment Tracking through Money Laundering Brokers: Case Study of an Unexplored Scam

Mijin Shin, Minjung Yoo, Eunbeen Lee, Yiseul Choi, Yunyoung Jung and Seongmin Kim (Sungshin Women's University)

Session2-6 Proposal for a Small Quantum Encryptor for Application Data Security

Hyeongyeop Kim, Seowoo Jung, Changuk Jang and Okyeon Yi (Kookmin University)

Session2-7 A Study on the Processing of Contraband Evidence; Focused on Child Sexual Abuse Material

Ye-eun Jeon, Yeram Lim and Seongmin Kim (Sungshin Women's University)

Session2-8 Generative AI and Child Sexual Abuse Material: Enhancing Legal Frameworks through Child Recognition Model

Yeram Lim, Ye-eun Jeon and Seongmin Kim (Sungshin Women's University)

Session2-9 COPY&RETURN: Nullifying Suspicious Cryptographic Operations for Behavioral Analysis of Ransomware

Yohan Lee, Jiseok Bang and Seungkwang Lee (Dankook University)

Session2-10 Interoperable Alternative to Canary-based Protection of Stack Smashing

Kangmin Kim (Dankook University), Jeong-Nyeo Kim (ETRI) and Seungkwang Lee (Dankook University)

Session2-11 Early Detection of Vulnerabilities in Static Analysis with Terraform and Kubernetes: Methods and Strategies

Hyeyeon Hyeon, Gahyun Kim, Mijin Kim, Yerin Chun and Seongmin Kim (Sungshin Women's University)

Session2-12 Exploiting Standalone Access Vulnerabilities in Envoy Proxy for Restart Attacks
Jiwon Yoo, Seo-yeon Kang and Seongmin Kim (Sungshin Women's University)

Session2-13 Interference Avoidance and Mitigation Technique Utilizing Cooperative Characterization of Interferers in Dense Wireless Networks
Jin-Min Lee, Hye-Yeon Shim and Il-Gu Lee (Sungshin Women's University)

Session2-14 An Effective Labeling Model for Minimal Labeled Encrypted Network Traffic Data
Daeho Choi, Minjeong Cho and Kiwook Shon (Seoul National University of Science and Technology)

Session2-15 5G Private Networks: A Survey on Architecture, Threats, Security Requirements, and Research Challenges
Changhyeon Woo, Jieon Oh, Yongho Ko and Ilsun You (Kookmin University)

Session2-16 Reliable Practical Byzantine Fault Tolerance Mechanism for High Throughput and Low Latency Blockchain Consensus
Namryeong Kim and Il-Gu Lee (Sungshin Women's University)

Session2-17 Cryptojacking Container Detection Based on Network Flow Analysis in Kubernetes
Sumin Kim, Soomin Lee, Jeongeun Ryu, Ri-Yeong Kim and Seongmin Kim (Sungshin Women's University)

Session2-18 Autoencoder-Based Face De-identification and Swapping for Enhanced Privacy Protection
So Yeon Kim, Seong Hyun Min, Ma Ii Jo and Won Gyum Kim (AIDEEP Inc.)

Session2-19 One-Pixel Attack Mitigation Technique
Yeon-Ji Lee and Il-Gu Lee (Sungshin Women's University)

Session2-20 Enhancing Alarm Systems Utilizing In-Network Computing for Mitigating Tampered Alarms
Junghyun Moon, Eunji Chang and Kookheui Kwon (KINAC)

Poster Session 3: 17:00 - 18:00, August 22 (Thursday), 2024
Session Chair: Il-Gu Lee (Sungshin Women's University)

Session3-1 Multi-Channel-Based Physical Layer Security for Dynamic Key Update
Chae-Yeon Park, Sun-Jin Lee and Il-Gu Lee (Sungshin Women's University)

Session3-2 The Paradox of Security Controls: Do Stricter Organizational Measures Improve Security Levels?
Jungjoo Oh and Il-Gu Lee (Sungshin Women's University)

Session3-3 Optimized Implementation of SABER on resource-constrained 16-bit MSP430 MCU
MinGi Kim, DongHyun Shin, YoungBeom Kim and Seog Chung Seo (Kookmin University)

Session3-4 TrustMark, can I trust you?
Su-Kyoung Kim, Yu-Ran Jeon, Jung-Hwa Ryu and Il-Gu Lee (Sungshin Women's University)

Session3-5 Partial Homomorphic Encryption for Privacy-Preserving Data Learning
Hye-Yeon Shim, Yu-Ran Jeon and Il-Gu Lee (Sungshin Women's University)

Session3-6 ML-DSA Implicit certificates: PQC Migration for IEEE 1609.2 Standard
YoonSun Han (Kookmin University), Dongmin Kim (Samsung Electronics) and Seog Chung Seo (Kookmin University)

Session3-7 Efficient Implementation of Falcon on Cortex-M4
Junhyeok Choi, Seung-Ju Wi and Seog Chung Seo (Kookmin University)

Session3-8 Repeated Embedding Techniques of Duplicated Least Significant Bits for Secure Digital Watermarks
Chae-Won Song, So-Hyun Park and Il-Gu Lee (Sungshin Women's University)

Session3-9 Practical Zero Trust Implementation for Low Latency and High Reliability Networks
Seo-Yi Kim and Il-Gu Lee (Sungshin Women's University)

Session3-10 A Study on the Strengthening of the Research Security System: Focusing on the Comparative Analysis of Laws and Systems in Major Countries
Hyun-Che Song, Hye-In Lee and Il-Gu Lee (Sungshin Women's University)

Session3-11 Forensic Analysis of Trust Wallet Artifacts in Browser Extension Environment
JiWon Bae and Kiwook Sohn (Seoul National University of Science and Technology)

Session3-12 Optimizing HQC on Cortex-M4
DongCheon Kim and Seog Chung Seo (Kookmin University)

Session3-13 A Study of V2V Certificate Management for PQC Migration
Jung-Hun Kang, Soojin Lee and Seung-Hyun Seo (Hanyang University)

Session3-14 Security Evaluation on the KpqC Round 2 Lattice-based Candidates
Sumin Park, Eunmin Lee, Hyun-A Noh, Hyojung Yoon and Joohee Lee (Sungshin Women's University)

Session3-15 Comparison of Voice Feature Extraction Techniques for Deepfake Detection
Seong-Bean Park, Jeong-Hyun Sim and Hyun-Min Song (Dankook University)

Session3-16 The architectural difference of Consumer, IoT and M2M eSIM RSP security
Hoseok Kwon, Taeho Won, Bonam Kim and Ilsun You (Kookmin University)

Session3-17 Analysis of Security Threats and Countermeasures of Artificial Intelligence-Based Digital Education
Hye-In Lee, Hyun-Che Song and Il-Gu Lee (Sungshin Women's University)

Session3-18 Hardware-based NTT accelerator design for ML-DSA
Byeonguk Jeong, Eunse Kang and Howon Kim (Pusan National University)

Session3-19 Efficient Fuzzing Technique for Lightweight Firmware-Over-The-Air
Na-Hyun Kim, Jin-Min Lee and Il-Gu Lee (Sungshin Women's University)

Session3-20 Enhancing Operational Technology Cybersecurity using Security Levels, Differential Privacy, and Post-Quantum Cryptography
Junghyun Moon, Eunji Chang and Kookheui Kwon (KINAC)

Session3-21 Evaluation and Analysis of Adversarial Attacks to Digital Watermarks
A-Young Jeon, Yeon-Ji Lee and Il-Gu Lee (Sungshin Women's University)

Proposal of distributed network data management mechanism based on keyword extraction from text content

Jinsu Kim[1][0000-0003-1009-3928] and Namje Park[2][0000-0003-4434-8933]

[1] Department of Convergence Information Security, Graduate School, Jeju National University, Jeju-si 63294, Korea

[2] Dept. of Computer Education, Teachers College, Jeju National University, 61 Iljudong-ro, Jeju-si, Jeju Special Self-Governing Province, 63294, Korea

namjepark@jejunu.ac.kr

Abstract. The current web environment is called Web 2.0, which means an environment in which data exchange is performed between service providers with servers and users who generate data. At this time, the service provider performs reading and writing by receiving content generated by users in addition to its own content and providing it to other users. Therefore, economic benefits are obtained using data by temporarily providing the right to use content to a service provider. However, as the importance of data is emphasized, Web 3.0, which provides data rights to users, is emerging. Web 3.0, which is currently being presented, requires learning to provide optimized data by learning user experience. In this paper, in order to perform optimized content through user experience in the Web 3.0 environment, major keywords are extracted from text-based content and the authority of content including keywords is proven on a block network to provide learning data that can be used for user experience. We propose a mechanism for recording and managing data in the block network.

Keywords: Keyword extraction, blockchain, web 3.0, content management, learning data

1 Introduction

The web has changed from Web 1.0, an environment in which data can be unilaterally transmitted by service providers, to Web 2.0, an environment in which content created by users is uploaded to a server and service providers provide the content in the server to users. In the Web 1.0 environment, content provision was performed by service providers, so each service provider that built the web environment had decentralization as an independent platform. However, in Web 1.0, data is unilaterally provided by service providers, and there are many difficulties in utilizing content generated by users. To overcome this limitation, Web 2.0 is an interactive web environment between service providers and users[1].

Web 2.0 provides a shared web environment where service providers, who own the platform, receive content from users and allow them to share it with other users. Thus, it overcomes the limitations of Web 1.0, which only provided data created by service providers. With the establishment of Web 2.0, data sharing among users has actively changed, significantly impacting the Internet. However, Web 2.0 forms a centralized structure as user data is concentrated on the platform service providers. This has the advantage of requiring users to have minimal network access since server management responsibilities lie with the platform. Nonetheless, it also contains the issue of creating a hierarchy as the authority over data is temporarily vested in the platform[2].
In response to the discussion on the importance of user data, the concept of Web 3.0 has been proposed, aiming to provide authority over data to users and creating a decentralized network environment. Web 3.0 refers to a distributed network form that utilizes various technologies such as blockchain, artificial intelligence, and the metaverse to offer optimized web services based on user experience. However, it is a concept that encompasses a comprehensive meaning of the futuristic web environment rather than a clearly defined concept[3].
Therefore, this paper defines the Web 3.0 environment as one where users create content based on a distributed network and directly provide their content to other users. In this paper, we propose a mechanism that uses textrank to extract key keywords from the generated text-based content, performs proof of authority and records the key keywords using blockchain, thereby providing an optimized experience for users based on the keywords in the blockchain network.

2 Data management mechanism based on text content keyword extraction

In this paper, we propose a mechanism that extracts key keywords from text content based on TextRank, records the extracted keywords along with content proof in the block network, thereby proving user authority over the content, and provides an optimized service based on user experience using the keywords. The proposed mechanism can be divided into the keyword extraction process, where key keywords are extracted from text content by the user; the content proof process, where the extracted keywords and content are recorded in the block network to prove data authority; and the data provision process, where data is provided to the user based on the keywords.
To upload content to a distributed network, N or more keywords are first extracted by TextRank. The extracted keywords(K_N) are used for learning content associations and searching related data. The content is then distributed into M or more data blocks to make it unidentifiable. Unified identification information(ID_B) is generated for the distributed data blocks(C_M), and the distributed data blocks, keyword and identification information are divided and stored among the participants in the distributed network participating in the blockchain. [Equation 1] shows the structure of the content data distributed and stored among users participating in the block network.

$$Content\ data = [ID_S, K_N, C_M] \quad (1)$$

When a request for content occurs, the user with the authority over the content conducts a transaction with the requester. If the transaction is successfully conducted and the content is to be provided to the requester, the content provider generates a dataset (D_T) that includes the requester's identification information(ID_R), content identification information (ID_B), and transaction-related metadata (M_D). [Equation 2] shows the structure of the dataset for transaction proof.

$$D_T = [ID_R, ID_B, M_D] \quad (2)$$

The dataset is then encrypted using the user's private key to prove that it has been verified by the user with the content authority and is broadcasted on the block network. Therefore, clients holding the distributed content data decrypt the dataset using the content user's public key to verify it as a proven dataset. They then use the requester information and content identification information within the dataset to transmit the corresponding content data to the requester. At this time, the keywords included in the content allow the user's learning model to perform learning with a certain weight so that the content requested by the user can be updated.

3 Conclusion

In this paper, we proposed a mechanism for providing content data based on user experience in Web 3.0 by extracting key keywords from text content, distributing and storing the extracted keywords along with the distributed content information on the block network, and performing reliable data transactions when content requests occur. In the future, research on artificial intelligence learning modules for optimized content provision through keyword learning is required.

Acknowledgement

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2022S1A5C2A04092269). And, this research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2023RIS-009) [Development of a Smart Tourism Content Service Platform Based on Artificial Intelligence through Digital Transformation of National Heritage Data]

References

1. Xu, Z., Geyong, M., Tong, L., Zhan, M., Xun, C., Shaowei, W.,; AI and Blockchain Empowered Metaverse for Web 3.0: Vision, Architecture, and Future Directions. IEEE Communications Magazine. 60-66, Jun. 2023.
2. Namje, P., Jin, K., Seungjoo, K., Dongho, W., Howon, K.,: WIPI mobile platform with secure service for mobile RFID network environment. Advanced Web and Network Technologies, and Applications: APWeb 2006 International Workshops. 741-748 (2006).
3. Sarika , Z., Deep, P., Abhishek, G., Sanket, G., Akash T,.; Text Summarization using TF-IDF and Textrank algorithm. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). Jun. 2021.

Poster: Blockchain-Based ESG Reporting and Refurbishment Certification in the Cosmetics Industry *

Cho Nwe Zin Latt[1][0000−0002−4258−9509]
Muhammad Firdaus[2][0000−0003−0104−848X], and
Kyung-Hyune Rhee[3][0000−0003−0466−8254]

[1] Department of Information Security, Graduate School,
Pukyong National University, South Korea
chocho1612@pukyong.ac.kr
[2] Industrial Science Technology Research Center,
Pukyong National University, South Korea
mfirdaus@pukyong.ac.kr
[3] Division of Computer Engineering and AI,
Pukyong National University, South Korea
khrhee@pknu.ac.kr

Abstract. The cosmetics industry faces increasing pressure to adopt and demonstrate adherence to Environmental, Social, and Governance (ESG) principles. Traditional ESG reporting mechanisms often struggle with transparency and data privacy issues. This paper proposes a framework that integrates Hyperledger Fabric, a permissioned blockchain platform, to enhance ESG reporting and refurbishment certification in the cosmetics industry. The framework ensures data privacy while maintaining transparency and utilizes Non-Fungible Tokens (NFTs) for product traceability and refurbishment certification, promoting circular economy principles.

Keywords: Blockchain, ESG Reporting, Circular Economy, Refurbishment Certification, Non-Fungible Tokens (NFTs).

1 Introduction

Due to increasing consumer awareness and stringent regulatory requirements, the cosmetics industry is pressured to adopt sustainable practices and maintain transparent operations. Environmental, Social, and Governance (ESG) criteria have become pivotal in assessing businesses' ethical impact and sustainability practices. Companies are required to ensure transparency in their operations,

* This research was supported as a 'Technology Commercialization Collaboration Platform Construction' project of the INNOPOLIS FOUNDATION (Project Number: 1711202494)

reduce their environmental footprint, and address significant environmental challenges, such as waste disposal and the use of controversial ingredients like marijuana [3].

However, traditional ESG reporting mechanisms face several challenges related to data integrity, transparency, and protection of sensitive information, further compounded by diverse regulatory requirements across regions. To address these challenges, this paper leverages blockchain, specifically Hyperledger Fabric (HLF), which can be used to issue digital refurbishment certificates that are cryptographically secured and easily verifiable by consumers and regulators [5]. With its advantages, HLF is well-suited for complex and large-scale ESG reporting needs. Additionally, Non-Fungible Tokens (NFTs) can improve transparency and traceability by creating unique digital identifiers for products, enabling detailed tracking throughout their lifecycle. This technology helps enhance consumer trust by providing verifiable proof of sustainability practices and combating counterfeit products.

2 Background

2.1 ESG Principles in the Cosmetics Industry

ESG principles are critical for promoting sustainable and ethical practices in the cosmetics industry. Key components of ESG include environmental responsibility, which involves reducing environmental impacts through sustainable production, minimizing waste, Eco-friendly packaging, and responsible sourcing of raw materials. Efforts also focus on reducing carbon emissions and efficient water management. Social responsibility emphasizes ensuring fair labor practices, promoting diversity and inclusion, and engaging with local communities. Ethical sourcing and avoiding harmful substances are paramount. Governance entails establishing a robust corporate governance framework, transparent reporting, and regulatory compliance to build stakeholder trust and ensure accountability.

2.2 Hyperledger Fabric

Blockchain technology, introduced with the advent of Bitcoin by Satoshi Nakamoto in 2008 [4], has revolutionized the way digital transactions are conducted. At its core, blockchain is a decentralized and distributed ledger that records transactions across multiple entities, ensuring that the recorded data cannot be altered through its consensus mechanism. On the other hand, HLF is a permissioned blockchain framework tailored for enterprise applications, offering significant advantages for ESG reporting and refurbishment certification. A key feature of HLF is its ability to manage complex transactions and workflows. Androulaki et al. [1] emphasize that the framework's modularity allows for customizable implementations of various components, such as consensus mechanisms and membership services, making it highly adaptable to diverse enterprise requirements. Additionally, HLF supports private channels, enabling sensitive information to be

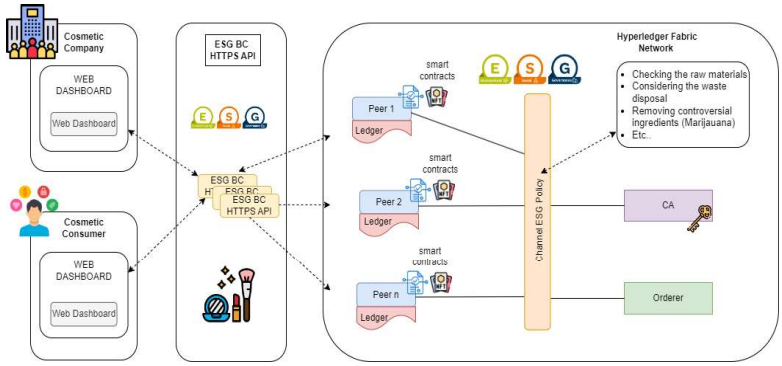


Fig. 1. Proposed Framework

shared only among authorized participants. This capability enhances privacy while preserving the transparency and immutability inherent in blockchain technology.

3 Proposed Framework

Figure 1 represents the proposed framework that leverages Hyperledger Fabric (HLF) in the cosmetic industry to ensure a secure, transparent, and efficient ESG reporting system. The framework comprises three main components: stakeholders (including cosmetic companies, customers, regulators, etc.), the ESG Blockchain (BC) HTTPS API as the intermediary interface, and the HLF network, which is set up with the necessary components such as peers, orderers, and Certificate Authorities (CAs).

Firstly, each participant, such as the cosmetic company, is issued a digital identity managed by a CA using Elliptic Curve Digital Signature Algorithm (ECDSA). Digital certification contains public keys, verifying participant identities and enabling secure transactions. The cosmetic company then collects ESG data, including reports on its raw materials, to avoid the use of controversial ingredients (such as marijuana) and to manage waste disposal through a web dashboard. This transaction is signed using the company's public ECDSA key, ensuring authenticity and integrity, and is then uploaded via the ESG HTTPS APT to the blockchain. To address privacy concerns, sensitive ESG data is encrypted using different ECC keys for transmission and storage [2]. This approach ensures that only authorized parties can access the data, safeguarding business information while maintaining transparency in ESG reporting.

The HLF network maintains peer nodes executing smart contracts [6]. NFTs provide unique product identifiers for detailed tracking and verification of sustainability practices, enhancing transparency and reducing counterfeit risks. Essential tracking information linked to NFTs ensures privacy. Smart contracts automate ESG compliance verification and data management.

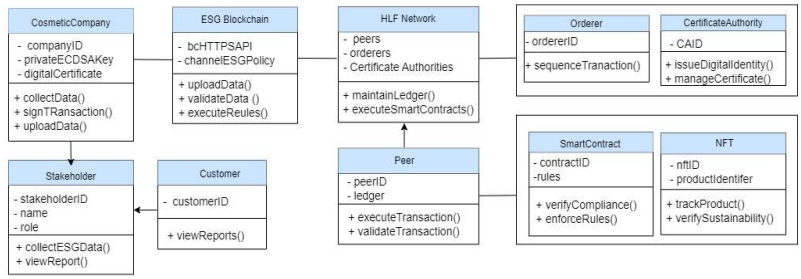


Fig. 2. ESG-BC data model

The channel ESG policy governs the network by defining data validation, access control, and transaction approval rules. It enforces ESG standards and coordinates stakeholders, ensuring data integrity and privacy. This framework supports ESG and circular economy principles in the cosmetics industry referred in Figure 2.

Framework Efficiency and Benefits Hyperledger Fabric's modular architecture allows customization for the cosmetics industry, with private channels ensuring data privacy and transparency. Its consensus mechanism ensures data integrity and immutability, crucial for ESG reporting. The framework's ability to handle large-scale ESG reporting efficiently makes it a robust solution.

Role of NFTs in the Framework NFTs provide unique digital identifiers, enhancing traceability and verification of sustainability practices. Unlike traditional identifiers, NFT are tamper-proof, reducing counterfeit risks. They failitate tracking a product's lifecycle and documenting refurbishment processes, supporting the circular economy.

4 Conclusion

This paper introduces a framework using Hyperledger Fabric for secure and transparent ESG reporting in the cosmetics industry, emphasizing data integrity, transparency, and privacy. Future research should focus on enhancing blockchain scalability and performance, integrating IoT for real-time data collection, and exploring advanced cryptographic methods for improved privacy. Key areas include ensuring compatibility with other blockchain platforms, developing user-friendly interfaces, adapting to regulatory changes, and assessing the environmental impact of blockchain technology. Real-world pilot studies within the cosmetics industry and exploring the framework's application in other sectors will help refine and expand its capabilities, further supporting sustainable development in the industry.

References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference. pp. 1–15 (2018)
2. Dwivedi, S.K., Amin, R., Vollala, S., Das, A.K.: Design of blockchain and ecc-based robust and efficient batch authentication protocol for vehicular ad-hoc networks. IEEE Transactions on Intelligent Transportation Systems (2023)
3. Gopalakrishnan, P.K., Hall, J., Behdad, S.: A blockchain-based traceability system for waste management in smart cities. In: International design engineering technical conferences and computers and information in engineering conference. vol. 83952, p. V006T06A015. American Society of Mechanical Engineers (2020)
4. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
5. Regueiro, C., Gómez-Goiri, A., Pedrosa, N., Semertzidis, C., Iturbe, E., Mansell, J.: Blockchain-based refurbishment certification system for enhancing the circular economy. Blockchain: Research and Applications **5**(1), 100172 (2024)
6. Wang, Q., Li, R., Wang, Q., Chen, S.: Non-fungible token (nft): Overview, evaluation, opportunities and challenges. arXiv preprint arXiv:2105.07447 (2021)

Random Public Codebook-Based Double XOR Cipher

Seung-Ha Ji[1][0009-0003-2030-7747], So-Eun Jeon[2][0000-0003-3700-8885]
and Il-Gu Lee[3][0000-0002-5777-4029]

[1] Dept. of Convergence Security Engineering, Sungshin Women's University, Korea
[2] Dept. of Future Convergence Technology Engineering, Sungshin Women's University, Korea
[3] Dept. of Convergence Security Engineering & Future Convergence Technology Engineering, Sungshin Women's University, Korea

Abstract. Data security has become increasingly crucial as the volume of network traffic surges. Recent research has actively explored the application of Differential Privacy (DP) and encryption to protect data. However, DP techniques face a trade-off between security and usability, while conventional encryption methods have complexity issues. This paper proposes the Random Public Codebook-based Double XOR Cipher (RPC-DXC) for data protection. Experimental results indicate that the proposed method increases security capacity by up to 90.39% and reduces latency by 76.61% compared to conventional DP techniques.

Keywords: Lightweight Encryption, Security and Privacy, Exclusive-OR.

1 Introduction

The volume of data transmitted and received in network environments has exploded with the increasing utilization across various industries. Differential privacy (DP) has been studied to ensure secure data transmission in a network environment. DP is a technique of adding artificial noise to data to prevent attackers from inferring the original data [1,2,3]. Although DP has the advantage of preventing information leakage even if attackers compromise data, the noise in the data can degrade the utility for legitimate users. To overcome the limitations of DP, previous research has applied differential levels of artificial noise to data, enhancing both security and utility [4,5,6,7]. However, these methods increase computational complexity and overhead while adding noise for DP.

Therefore, this paper proposes the Random Public Codebook-based Double XOR Cipher (RPC-DXC) to address the trade-off between security and utility and the high complexity issues of conventional DP techniques.

2 Related Work

S. Kim [5] proposed a new distributed learning technique combining the Vickrey-Clarke-Groves (VCG) mechanism and DP to address system performance degradation

and privacy leakage in heterogeneous IoT environments. Maoguo Gong et al. [6] introduced an adaptive differential privacy-preserving learning method to resolve the trade-off between utility and security in DP. Agarwal et al. [7] suggested a lightweight DP mechanism. This multi-dimensional Skellam method analyzes the loss distribution of personal information to ensure privacy and improve efficiency by extracting only the Poisson distribution. These studies highlight the increase in computational complexity when solving the trade-off between security and utility.

3 Random Public Codebook-based Double XOR Cipher Technique

This section describes the Random Public Codebook-based Double XOR Cipher (RPC-DXC) method for secure data transmission. Fig. 1 illustrates the operational mechanism of RPC-DXC.

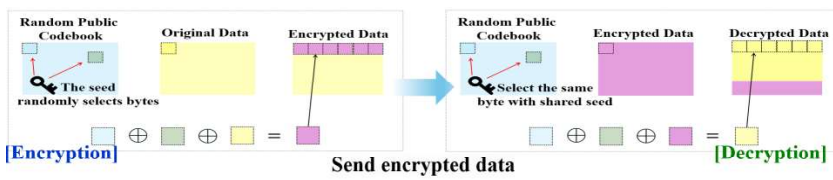


Fig. 1. Operation mechanism of RPC-DXC

First, the sender generates a random codebook consisting of random byte values and uses a pre-shared seed with the receiver to generate index positions randomly extracted from the byte table using a Random Number Generator (RNG). The key at the index generated by the RNG is extracted, and the data to be encrypted is XORed byte-by-byte. The sender transmits the RPC-DXC encrypted data to the receiver, who uses the pre-shared seed with the sender to extract the key index from the RNG to decrypt the data. RPC-DXC performs data encryption using XOR operations without encrypting the byte codebook by sharing the RNG in advance between users, ensuring secure and lightweight data transmission.

4 Performance Evaluation

4.1 Evaluation Environment

This section explains the evaluation environment used to verify the performance of the proposed RPC-DXC technique. The evaluation used the Bank Account Fraud Dataset Suite (NeurIPS 2022) [8] to detect fraud bank accounts for machine learning training. Legitimate users detect forged accounts using the received data with a machine learning model, while attackers extract financial personal information using the model.

The Decision Tree was used as the machine learning model, and the seed shared between the sender and receiver was generated using a Linear Congruential Generator (LCG).

For the performance evaluation of RPC-DXC, conventional models included the encoding-based DP technique for recovering DP data by legitimate users (Con-DP-wEn) [4] and the standard DP technique without encoding (Con-DP-woEn) [1,2,3]. Con-DP-wEn defines sensitive features required by the attacker to identify labels and user features needed by legitimate users to detect account forgery, applying DP to sensitive features and encoding to user features. The encoding was implemented with the Reedsolo 1.7.0 module, and the feature importance ranking for label identification was derived using the permutation importance model. Con-DP-woEn applies only the DP technique.

The performance of each method was evaluated using security capacity and latency as metrics. Security capacity refers to the performance difference between the learning accuracy of the legitimate user's model and the attacker's model, while latency refers to the time taken to decode the original data from data processing by the model.

4.2 Evaluation Results and Analysis

Fig. 2 compares the security capacity of the proposed RPC-DXC with the conventional models Con-DP-wEn and Con-DP-woEn.

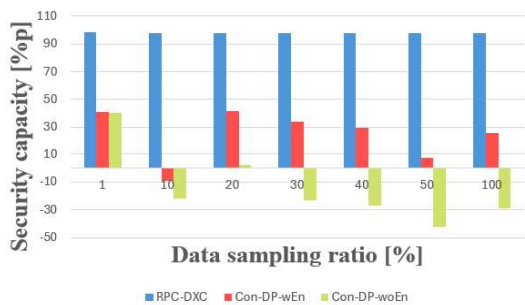


Fig. 2. Security Capacity Performance Evaluation Results for RPC-DXC

According to Fig. 2, the proposed RPC-DXC model had the highest security capacity since attackers could not decrypt the original values without knowing the seed required for decryption. Con-DP-woEn applied DP to all data without a separate original data recovery mechanism, resulting in inefficient learning performance for legitimate users and attackers. Con-DP-wEn, which allowed legitimate users to recover original data using the encoding mechanism effectively, performed better than Con-DP-woEn. However, the encoding pattern exposure could also allow attackers to recover the original data, resulting in lower security capacity than the proposed RPC-DXC.

Fig. 3 shows the latency performance for RPC-DXC.

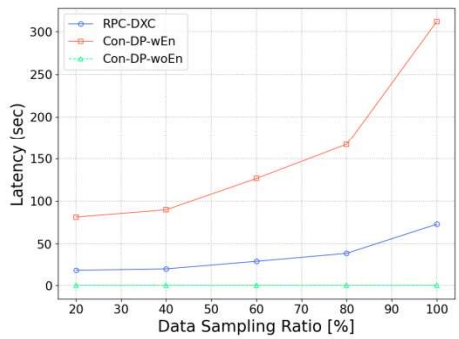


Fig. 3. Latency Performance Evaluation Results for RPC-DXC

As shown in Fig. 3, the latency increased for RPC-DXC, Con-DP-wEn, and Con-DP-woEn as the data sampling rate increased. RPC-DXC, which performs the simple XOR operation, reduced latency by up to 76.61% compared to the more complex encoding-based DP technique Con-DP-wEn. Con-DP-woEn required only operations to generate and insert artificial noise into the data, resulting in the lowest latency.

5 Conclusion

This study proposes a lightweight encryption-based data security technique to minimize the trade-off between security and utility and the complexity issues. Evaluation results show that RPC-DXC increased security capacity by an average of 90.39% and 139.97% compared to conventional techniques and reduced latency by 72.77% compared to Con-DP-wEn. Future research plans to optimize the size of the encryption keys used in terms of security and complexity. Additionally, the performance of the proposed RPC-DXC will be verified by comparing it with representative lightweight encryption techniques.

Acknowledgments. This work was supported by a 2024 Real Challenge program grant funded by Korea Institute of Human Resources Development in Science and Technology(KIRD), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Bo Ning, Yunhao Sun, Xiaoyu Tao, and Guanyu Li, “Differential privacy protection on weighted graph in wireless networks,” Ad Hoc Networks, vol. 110, 102303, 2021.

2. N. Wu, F. Farokhi, D. Smith and M. A. Kaafar, "The Value of Collaboration in Convex Machine Learning with Differential Privacy," 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 304-317, 2020.
3. Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor, "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3454-3469, 2020.
4. Seung-Ha Ji, So-Eun Jeon, Il-Gu Lee, "Highly Reliable Differential Privacy Technique Utilizing Error Correction Encoding," Annual Symposium of KIPS 2024, Pyeongchang, Korea, May. 2024.
5. S. Kim, "Incentive Design and Differential Privacy Based Federated Learning: A Mechanism Design Perspective," in IEEE Access, vol. 8, pp. 187317-187325, 2020.
6. Maoguo Gong, Ke Pan, Yu Xie, A.K. Qin, Zedong Tang, "Preserving differential privacy in deep neural networks with relevance-based adaptive noise imposition," Neural Networks, vol. 125, pp. 131-141, 2020.
7. Agarwal, Naman and Kairouz, Peter and Liu, Ziyu, "The Skellam Mechanism for Differentially Private Federated Learning," Advances in Neural Information Processing Systems, vol. 34, pp. 5052-5064, 2021.
8. Sérgio Jesus, José Pombal, Duarte Alves, André Cruz, Pedro Saleiro, Rita Ribeiro, João Gama, Pedro Bizarro, "Turning the Tables: Biased, Imbalanced, Dynamic Tabular Datasets for ML Evaluation," Advances in Neural Information Processing Systems, vol. 35, pp. 33563-33575, 2022.

Error Correction Code-Based Two-Layer Digital Watermarking Technique

Do-Eun Kim[1], So-Hyun Park[2] and Il-Gu Lee[3]

[1] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
20221080@sungshin.ac.kr
[2] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
220227022@sungshin.ac.kr
[2] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
iglee@sungshin.ac.kr

Abstract. Digital watermarking is a technology that inserts information into digital content. This paper proposes a hierarchical watermarking technique that can detect attacks, correct errors, identify owners, and analyze the quality and error correction capabilities.

Keywords: Digital Watermarking, Error Correction Code, Image Copyright, Noise Attack

1 Introduction

Recently, eradicating piracy and identifying content generated by artificial intelligence has become an important task in the digital content industry [1]. Digital watermarking is a technology that inserts information into digital content. It protects copyright and ownership by inserting copyright information, preventing illegal copying, and identifying content creators [2, 3]. However, digital watermarking is vulnerable to geometric or non-geometric attacks [4].

In this paper, we propose a hierarchical watermarking technique using error correction codes. To respond to attacks that infringe copyright and ownership rights of digital content, the digital ID of the information subject is encoded using an error correction code and repeatedly inserted into the signature image, hierarchically inserted with the watermark image, and then restored and extracted. The proposed method maintained invisibility and improved performance over the conventional method in capacity and robustness.

2 Proposed Watermarking Framework

The discrete wavelet transform (DWT) algorithm was used to insert and extract the proposed hierarchical watermarking.

2.1 Hierarchical Watermark Insertion Algorithm

The proposed hierarchical insertion algorithm is shown in Fig. 1.

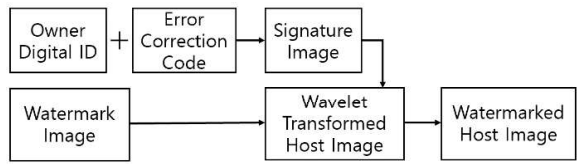


Fig. 1. Proposed Hierarchical Insertion Algorithm

First, an error correction code is generated using the digital ID of the information subject, and the owner is converted to binary numbers, and the binary sequence of the error correction code is replaced with the black/white pixels of the image and repeatedly encoded with padding to create a signature image for attack detection. In this study, an 8-digit decimal digital ID was used. The error correction code used is the BCH (Bose–Chaudhuri–Hocquenghem) error correction code. Afterwards, a signature image for forgery detection is inserted into the LL (low-low) sub-band, and a watermark logo image is inserted into the HL (high-low) sub-band into the host image that discrete wavelet transformed by applying Haar-wavelet.

2.2 Hierarchical Watermark Extraction Algorithm

The proposed hierarchical watermarking extraction algorithm is shown in Fig. 2.

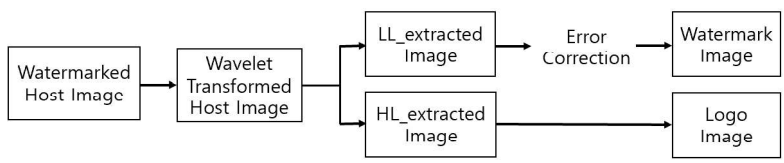


Fig. 2. Proposed Hierarchical Extraction Algorithm

The watermarked host image is subjected to discrete wavelet transformation to extract images from the HL sub-band and LL sub-band. In the HL sub-band, the extracted watermark image can extract a high-quality image with an excellent NCC (Normalized Cross-Correlation) value with the original image without additional processing. In the LL sub-band, error correction is performed to restore and extract the signature, and black/white pixel values are converted to binary 0/1 to extract the attack detection signature.

3 Performance evaluation

This experiment targeted 512x512 color scale images; Lenna was used as the host image, and Pepper image was used as the watermark image.

3.1 Hierarchical Watermarking Framework

Fig. 3 compares the invisibility according to the insertion strength of the conventional single image insertion watermarking and the proposed hierarchical watermarking with the cognitive similarity, the peak signal-to-noise ratio (PSNR), and the structural similarity index map (SSIM).

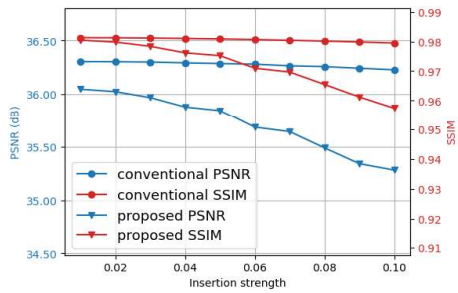


Fig. 3. SSIM and PSNR per Watermarking Type and Insertion Strength

In the case of conventional single-image watermarking, even if the insertion intensity is increased to 0.1, the PSNR of the host image is 36.23dB, and the SSIM is 0.9795, showing high invisibility. Inserting additional signature images according to the proposed hierarchical watermarking method reduces invisibility compared to the conventional single image watermarking method, but inserting a watermark by increasing the LL sub-band insertion strength to 0.1 in a situation where the HL sub-band insertion strength is 0.05 The PSNR of the host image is 35.28dB, and SSIM is 0.9573, showing excellent visual and structural invisibility.

3.2 Error Correction

After performing a Gaussian noise attack on the watermarked image implemented by the proposed method, the standard deviation of the Gaussian normal distribution was adjusted, and the quality and error correction performance of the signature image extracted from the LL sub-band were analyzed. Fig. 4 shows the change in the state of the signature image when a Gaussian noise attack with a standard deviation of 0.5 is performed.

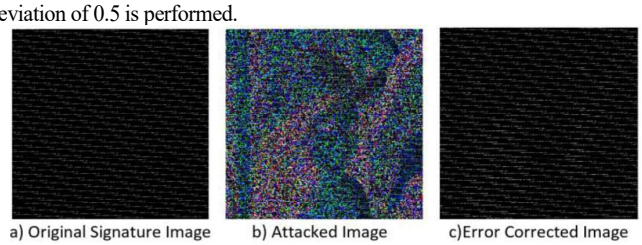


Fig. 4. a) Original Signature Image, b) Attacked Image, c) Error Corrected Image.

a) is a signature image in which the digital ID of the information subject is encoded based on BCH codes and repeatedly inserted 300 times, b) is a signature image extracted after performing an attack with the standard deviation of Gaussian noise set to 0.5, c) is the signature image on which BCH error correction was performed in b).

To evaluate the performance of error correction codes, uncoded cases without error correction codes were compared with BCH error correction codes (BCH codes, Bose–Chaudhuri–Hocquenghem codes) implemented to correct 5-bit errors. In addition, to analyze the performance according to the number of repeated signature insertions of the signature image, the user's digital ID encoded with an error correction code was repeatedly inserted 300, 600, and 900 times.

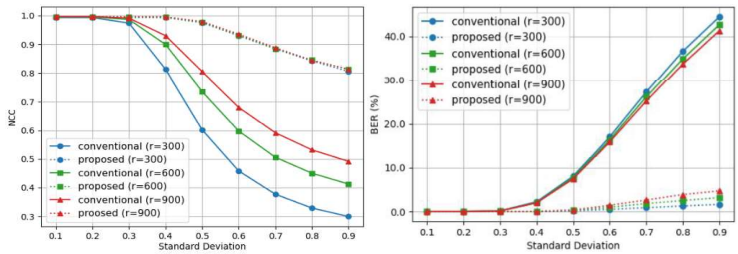


Fig. 5. NCC and BER from LL Layer Extracted Signature Image per Signature Repetition and Standard Deviation

Fig. 5 shows the signature image's quality and error correction performance according to the number of signature insertions and the presence or absence of an error correction function during a Gaussian noise attack. The signature image was encoded with error correction using BCH codes, and the signature image was used with r, the parameter for the number of repeated insertions, adjusted to 300, 600, and 900. After performing a Gaussian noise attack on the host image into which the watermark image and signature image were hierarchically inserted, the NCC and BER of the signature image extracted from the LL sub-band were measured. Conventional is a signature image extracted immediately after a Gaussian noise attack, and proposed is a proposed model that performed error correction in conventional. Conventional is in the same situation as b) in Fig. 4, and proposed is in the same situation as c) in Fig. 4.

As a result of comparing and analyzing the NCC and BER of the attacked image and the signature extracted by recovering the error using an error correction code, the NCC of the signature image decreases, and the BER increases as the standard deviation of the Gaussian noise attack increases, and as the number of insertions increases, Quality deteriorates accordingly. The damaged signature was restored using BCH error correction to obtain excellent NCC and BER. For example, in the case of conventional (r=300) with a standard deviation of 0.9 and signature insertion 300 times, the NCC of the signature image extracted from the LL sub-band immediately after the attack was severely damaged with an NCC of 0.2993 and a BER of 44.86%. On the other hand, the proposed (r=300), which restored the damaged signature image with BCH codes, increased NCC to 0.8051 and lowered BER to 3.18%.

4 Conclusion

This study proposed a hierarchical watermarking framework with digital watermark owner identification, forgery detection, and error correction functions. In the insertion algorithm, it is possible to identify the owner and detect forgery by repeatedly inserting the signature that encodes the digital ID with error correction along with the watermark image, and the signature can be restored with an error correction code through the proposed extraction algorithm even in attack and damage situations. According to the experimental results, the proposed method using an error correction code reduced the 44.23% BER of the conventional method to 2.64% by restoring the signature image through error correction, even in a Gaussian noise attack situation. The proposed method ensured robustness against watermark neutralization attacks while maintaining invisibility and increasing capacity efficiency through frequency division-based hierarchical watermarking.

Acknowledgments. This paper was conducted as a result of the research results of the 2024 Industrial Innovation Talent Growth Support Project (RS-2024-00415520) of the Ministry of Trade, Industry and Energy and the Korea Institute for Advancement of Technology (RS-2024-00415520) and the ICT Innovation Talent 4.0 Project of the Ministry of Science and ICT and the Korea IT Planning and Evaluation Institute (No. IITP-2022-RS-2022-00156310)

References

1. Mark, F., Paulius, J.: Originality and the future of copyright in an age of generative AI. Computer Law & Security Review, 51, 105892. (2023)
2. Evsutin, O., Dzhanashia, K.: Watermarking schemes for digital images: Robustness overview. Signal Processing: Image Communication, 100, 116523. (2022)
3. Do-Eun, K., So-Hyun, P., Il-Gu, L.: "디지털 워터마킹 공격 탐지를 위한 계층적 워터마킹 기법", Proceedings of the Annual Symposium of Korea Information Processing Society Conference (KIPS) 2024, Vol. 31, no. 1, pp. 283-284, 2024.
4. Begum, M., Uddin, M. S.: Digital Image Watermarking Techniques: A Review. Information. MDPI AG. (2020)

A Study on Enhancing Reauthentication Methods Among Workloads in a ZTA Environment

Wonhwi Choi[1] and Jungsoo Park[2]

[1] Kangnam University(Undergraduate Student), Republic of Korea
whc.ppp@gmail.com
[2] Kangnam University(Professor), Republic of Korea
jspark@kangnam.ac.kr

Abstract. Zero Trust Architecture (ZTA) is a security framework based on the principle of "never trust, always verify," continuously validating all access requests to protect organizational resources. While this enhances security by verifying every access attempt, constant reauthentication can disrupt workflows and reduce productivity. To address this, the proposed model uses Policy Information Points (PIPs) to collect real-time information on user behavior, device status, and network activity. This approach enhances security and improves work efficiency by allowing seamless transitions between resources with minimal reauthentication. The paper details the components of ZTA and Software Defined Perimeter (SDP), discusses the PIP-based model, and presents how it enhances work efficiency while maintaining security.

Keywords: Zero Trust Architecture(ZTA) · Software-Defined Perimeter(SDP) · Authentication and authorization

1 Introduction

Zero Trust Architecture (ZTA) is a security framework based on the principle of "never trust, always verify," requiring continuous verification of all access requests. It maintains a consistent security posture by not trusting any user or device, whether inside or outside the organization[1, 3]. Access permissions are managed by the Policy Enforcement Point (PEP) and Policy Decision Point (PDP), which enhance security but can disrupt workflow. To balance security and productivity, Policy Information Points (PIPs) are used to gather contextual information such as user behavior and device status. This dynamic approach minimizes repeated full authentication processes[2]. Section 2 discusses ZTA and Software-Defined Perimeter (SDP), with ZTA treating all network traffic as a potential threat and SDP defining the traditional physical perimeter in software. Section 3 proposes a model that adheres to ZTA and SDP policies while creating a seamless work environment using PIPs. Section 4 summarizes the study's main results and conclusions, aiming to maintain ZTA and SDP compliance while ensuring a smooth work environment.

2 Related Work

2.1 ZTA - Zero Trust Architecture

Zero Trust Architecture (ZTA) is a security framework based on the principle of "never trust, always verify." It employs methods like microsegmentation to enhance security by isolating network segments with distinct security policies. ZTA continuously verifies user identities and device statuses, ensuring strict authentication for all access requests and applying the Principle of Least Privilege to minimize the attack surface. Key components of ZTA include the Policy Decision Point (PDP), Policy Enforcement Point (PEP), and Policy Information Points (PIPs), which are essential for managing network security policies[5, 6].

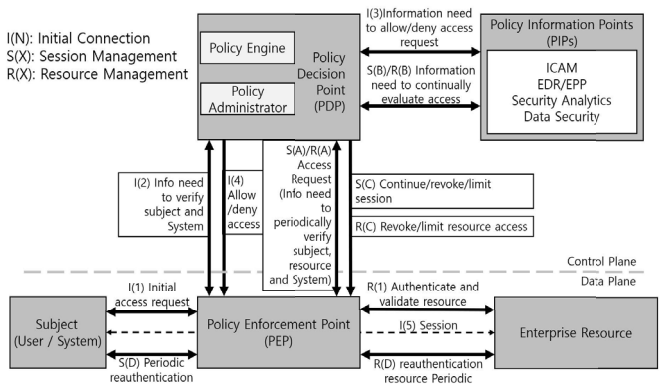


Fig. 1. Zero Trust Work Flow

2.2 Software-Defined Perimeter(SDP)

SDP v2.0 is the latest security specification released by the Cloud Security Alliance (CSA), designed to implement Zero Trust principles. Moving away from traditional network-based security boundaries, SDP focuses on securing users, assets, and resources. This approach allows for stricter access control and a more reliable security environment. SDP primarily consists of Controllers, Initiating Hosts (IH), and Accepting Hosts (AH), which form a dynamic, identity-centric perimeter to enhance security[4].

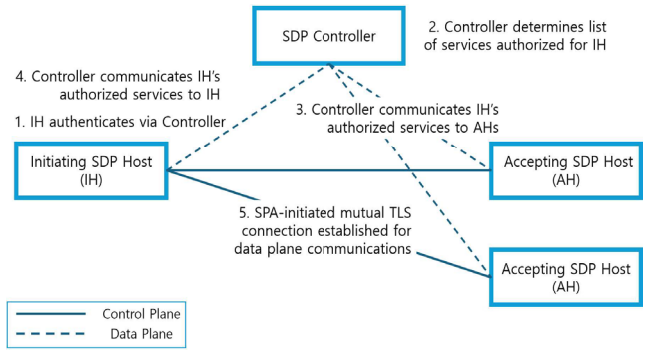


Fig. 2. Software-Defined Perimeter (SDP) Access Work Flow

3 Proposed Model

Zero Trust Architecture (ZTA) and Software-Defined Perimeter (SDP) require users to log out and re-authenticate for each new resource, which can disrupt workflow. To address this, we propose using Policy Information Points (PIPs) to regularly collect contextual information. This approach facilitates seamless access to resources while maintaining ZTA and SDP principles, reducing the need for repeated logins and re-authentication.

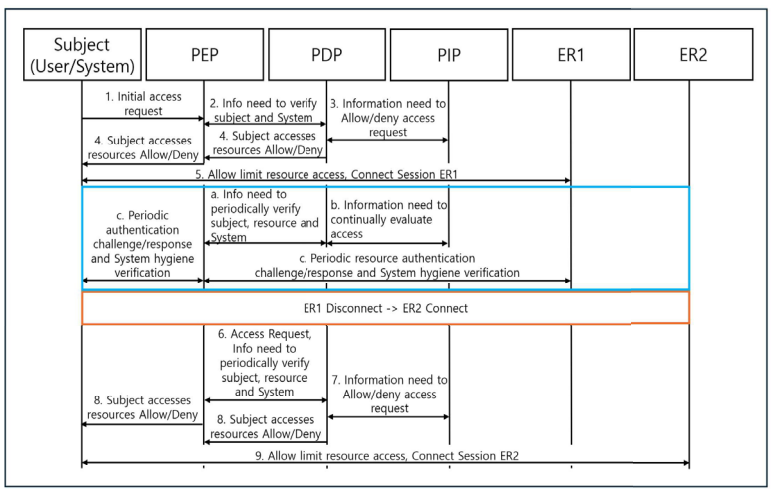


Fig. 3. Workflow for Reauthentication Methods in a Zero Trust Architecture Environment

In Figure 3, the workflow for improving work efficiency operates as follows:
1. The subject requests access to ER1 through the PEP.
2. The PEP asks the PDP for verification information.
3. The PDP gathers contextual information from PIPs to decide on access.
4. The PDP sends an Allow or Deny message to the PEP, which informs the subject.
5. If denied, re-authentication is needed. If allowed, the subject gets minimal access to ER1.
After the access decision and connection to a resource, continuous monitoring and collection of contextual information begin. User behavior, device status, and network activity are regularly monitored and sent to the PDP via the PEP. This information is provided to the PIPs for ongoing security verification and management. The session remains active until terminated.
a. The PEP periodically collects necessary information.
b. The PDP continuously evaluates the collected information.
c. Information is periodically collected to verify system integrity.
The section marked in orange indicates the termination of the ER1 connection and the initiation of a new connection to ER2.
6. After disconnecting from ER1, the subject uses periodically verified information to request access to ER2 from the PEP and PDP.
7. The PIPs provide contextual information collected during access to ER1 to the PDP for an access decision.
8. The PDP sends an Allow or Deny message to the PEP, which then informs the subject.
9. If access is denied, re-authentication is required. If allowed, the subject gains minimal access to ER2, and information collection begins immediately.

4 Conclusion

This study introduces an innovative login mechanism based on Zero Trust Architecture (ZTA) and Software-Defined Perimeter (SDP) to reduce repetitive re-authentication inefficiencies. Utilizing Policy Information Points (PIPs) to gather contextual data, the system maintains the user's authenticated state, enabling seamless access to multiple resources without repeated logins. If significant changes in authentication data occur, re-login is required to ensure security. This approach enhances work efficiency, balances security with operational efficiency, and minimizes workflow disruptions while providing strong threat protection.

Acknowledgments.
"This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-2020-0-01602) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)"

References

1. National Institute of Standards and Technology (NIST). "Zero Trust Architecture." NIST Special Publication 800-207, 2020. National Institute of Standards and Technology, U.S. Department of Commerce. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf.
2. National Institute of Standards and Technology (NIST). "Implementing a Zero Trust Architecture." Computer Security Resource Center (CSRC), NIST Special Publication 1800-35, National Institute of Standards and Technology, Sept. 12, 2023, https://csrc.nist.gov/publications/detail/sp/1800-35/draft.
3. Jungsoo Park et al.. "Analysis of the U.S. DoD Zero Trust Reference Model and Domestic Implementation Strategies." Proceedings of the Summer Academic Conference of the ICT Platform Society, vol. 10, no. 1, 2023, 44-49.
4. Cloud Security Alliance. Software-Defined Perimeter (SDP) Specification v2.0. Cloud Security Alliance, 10 Mar. 2022, https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-zero-trust-specification-v2.
5. Alevizos, L., et al. "Automation and Orchestration of Zero Trust Architecture: Potential Solutions and Challenges." Machine Intelligence Research. 2022. DOI: 10.1002/spy2.191.
6. Syed, N. F., et al. "Zero trust architecture (ZTA): A comprehensive survey." IEEE Access, vol. 10, 2022, pp. 57143-57179. DOI: 10.1109/ACCESS.2022.3174679.

Exploring Hybrid Approach-Steganalysis in Digital Forensics Applications

Rini Wisnu Wardhani[1[0000−0003−0565−6458]] *, Dedy Septono Catur Putranto[2,3[0000−0002−1246−7877]], Thi-Thu-Huong Le[2,3[0000−0002−8366−9396]], Agus Mahardika Ari Laksmono[1[0009−0007−6606−3716]], Hwang Yeonjeong[1[0009−0003−6906−7772]], and Howon Kim*[1[0000−0001−8475−7294]]

[1] School of Computer Science and Engineering, Pusan National University, Busan 609735, South Korea
[2] IoT Research Center, Pusan National University, Busan 609735, South Korea
[3] Blockchain Platform Research Center, Pusan National University, Busan 609735, South Korea
*howonkim@pusan.ac.kr

Abstract. Digital forensics (DF) plays a critical role in uncovering cybercrimes by analyzing vast amounts of digital evidence. However, traditional methods struggle with scalability and efficiency as datasets become increasingly complex. While machine learning offers a promising avenue, current approaches often rely on intricate algorithms with limited interpretability. This work explores a hybrid DF environment that merges classical and quantum techniques. We examine four hybrid approaches (i.e., CC, QC, CQ, and QQ) and evaluate a steganalysis CQ experiment within the accuracy results, compare prediction points versus the number of training data, and provide a histogram of ablation analysis of score predictions of the classical technique using various types of neural network. The hybrid approach in steganalysis, which combines the EfficientNet B1-B3 neural network with quantum approach processing, achieves a 97% efficiency compared to classical methods.

Keywords: Hybrid Quantum · Digital Forensic · Digital Evidence · Steganalysis

1 Introduction

The rapid evolution of digital technologies has revolutionized various aspects of modern life, but it has also brought new challenges, particularly in cybersecurity. With the proliferation of cybercrimes and sophisticated digital threats, the need for effective digital forensics (DF) methodologies has never been greater. DF systematically applies analytical and investigative techniques to identify, collect, preserve, analyze, and present digital evidence from various electronic devices, storage media, and computer networks [14]. Conventional methods used in DF are becoming less effective due to the increasing amount, complexity, and variety of digital evidence encountered in modern investigations. Human experts carry out forensic investigations using script-based tools and applications, which are laborious, require specialized knowledge, and are susceptible to mistakes [21] [17]. This has led to the investigating of novel methodologies utilizing artificial intelligence (AI) and machine learning (ML).

* This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business (Pusan National University) support program (IITP-2024-2022-0-01201) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)

Thagard's research [21] indicates that utilizing AI and machine learning ML to train datasets for DF investigators is a significant advancement in this field. However, current approaches often rely on complex algorithms that can be opaque and difficult to interpret. Balushi et al. research [1] emphasizes the importance of strengthening interpretability in these models for successful integration into practical applications. Advancements are strongly needed to improve DF investigations' efficiency, scalability, and accuracy while prioritizing interpretability to ensure trust and reliability in the forensic process. The Europol Innovation Lab's Observatory Report also underscores the potential of quantum technology for law enforcement [22], necessitating proactive efforts to monitor its implementation in DF developments.

This paper explores hybrid environments potential, with a specific focus on steganalysis as a practical application in DF, as noted in [13], [8], [5], and [15]. We examine hybrid categories such as Classical-Classical (CC), Classical-Quantum (CQ), Quantum-Classical (QC), and Quantum-Quantum (QQ), as studied in [9], providing an overview of research conducted in hybrid classical and quantum environments and the use of ML in quantum computing (QML). This study further elaborates experiments on image steganalysis [23], utilizing the EfficientNet B1-B3 convolutional neural network (CNN) architecture in the classical layer and Sim et al.'s circuits 5 and 6 for the quantum circuit [19], which reported a significant 97% improvement in efficiency compared to pure classical methods. Further, in this study, we analyze the accuracy of the results by comparing the prediction points with the number of training data. Additionally, it provides a histogram of the ablation analysis of score predictions for the classical technique using different types of EfficientNet. The results highlight the potential of hybrid CQ approaches to address big data challenges in steganalysis for the DF field. This research paves the way for further investigation into the applications of classical-quantum hybrid methods across various models and stages of DF investigations for practitioners in the field.

2 Digital Forensics

According to the DF roadmap outlined in the Digital Forensics Research Workshop 2021 and research by Reith et al., [18], DF has been defined as the use of scientifically derived and proven methods for the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources. DF is relevant for individuals aspiring to be part of the profession, particularly DF examiners. Notable investigative models include the Digital Forensics Research Workshop Model (DFRWS), the Abstract Digital Forensics Model (ADFM), the Integrated Digital Investigation Process Model (IDIP), and the End-to-End Digital Investigation Process Model (EEDIP) [1].

Steganography and steganalysis are relatively new scientific disciplines in security systems and DF, but they have matured significantly over the past two decades [8]. Steganography aims to hide messages in a not easily noticeable way, while steganalysis aims to identify their existence and recover hidden information. Both steganography and steganalysis have received significant attention, especially from law enforcement, as cybercriminals and terrorists often use steganography to evade arrest when possessing encrypted, incriminating material [13]. Understanding hidden messages in digital media, especially digital images, and mastering cutting-edge techniques to uncover hidden information is crucial for detecting criminal activities [13].

3 Hybrid Approaches

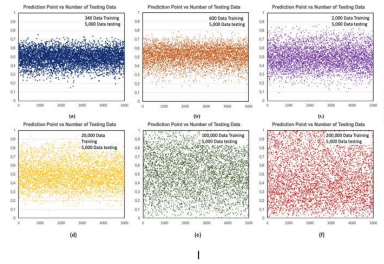
Hybrid approaches combine classical and quantum techniques to facilitate complex calculations or data handling. Research [9] [4] has identified four distinct approaches to integrate quantum computing and ML disciplines: CC, CQ, QC, and QQ. The classical DF approach involves human investigators employing various tools and script-based applications, which is time-consuming, requires expertise, and is prone to errors [21] [17]. The surge in electronic-based cybercrimes has highlighted the urgent need for digital solutions [17]. While the utilization of AI and ML in this domain is still in its early stages [10], some research, such as [1], has been able to identify notable ML works in DF in hybrid classical environments, combining one or more classical ML techniques, particularly categorized as CC, that utilize classical computation to assist the field of DF. Meanwhile, classical data is processed using QML algorithms categorized in the CQ method, exploring the potential of classical learning techniques to assist in various quantum tasks. The QC approach uses classical ML algorithms to process quantum data. The hybrid QQ approach leverages QML algorithms to analyze data directly on real quantum computers, offering significant future potential despite being in its early stages of development [16] [6].

In DF, AI and ML are integrated into practices, potentially enhancing the effectiveness and accuracy of investigations. Using AI and ML applications to train datasets for widespread use by DF investigators represents a significant progression. Garfinkel [11] emphasizes the growing importance of AI and ML in criminal investigations, particularly DF. Due to the rise of cyberattacks, DF necessitates acquiring, processing, and analyzing vast amounts of digital data [3]. Hybrid quantum-classical variations algorithms have dominated recent developments in QML [24]. Deep learning with quantum combined is now well suited to be impacted by classical deep neural networks, which are powerful learning techniques [2].

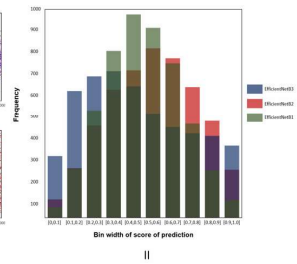
4 Hybrid Quantum-Classical Steganalysis for Digital Forensic

This paper further details Wardhani et al.'s steganalysis research [23], using the Qiskit framework to design and visualize results. The dataset from the Kaggle competition Alaska#2 contains 75,000 JPEG covers and stego-images concealed using three algorithms [12] [7]. The images were processed using a 3.8 GHz processor, 8 cores, and running Qiskit version 0.34.0 with Python 3.8.12. In the classical layer, the JPEG input images are transformed into the spatial domain and fed into the CNN via a set of fixed transform kernels of varying sizes. The feature vector is primarily fed to a fully connected layer, which performs CNN-based classification. The single CNN architecture Tan's EfficientNet model [20] was employed, demonstrating significantly higher classification accuracy and efficiency. Meanwhile, in the quantum layers, incorporating quantum circuits from Circuits 5 and 6, as proposed by Sim et al., [19], the classifier output from EfficientNet was encoded into quantum circuits using feature maps generated by ZZFeatureMap circuit libraries. This encoding enabled the construction of QNNs by integrating them into the alternating convolutional and pooling layers of the QNNs module.

Figure 1(I) visually presents the experimental results, specifically exploring the impact of varying the total amount of data. We compare the predicted outcomes against the number of training data points to assess prediction accuracy. Altering the total amount of data processed led to different findings derived from the 5,000 tests conducted in this study. The first graph, Figure 1(I)-(a), depicts results from training with 340 cover and stego data points. Figure 1(I)-(b) shows results for 600 data points and stego data, and Figure 1(I)-(c) for 2,000 data points. The distribution in



I



II

Model	Classical Layer	Quantum Layer			
	Feature Extraction	Encoding	Circuit		
			Basic	5	6
Hybrid	EfficientNet B1	Basic	93,9	94.7	93.9
		Amplitude	93,9	94.8	93,9
		Angle	94.4	94.2	93.9
Classical		–	93,8		
Hybrid	EfficientNet B2	Basic	95,1	95.5	95
		Amplitude	95,1	95,6	95
		Angle	95,2	95,8	95
Classical		–	94,9		
Hybrid	EfficientNet B3	Basic	95,8	96,04	95,9
		Amplitude	95,9	96.05	95,8
		Angle	95,8	97	96,03
Classical		–	95,6		

III

Fig. 1: Experiment results:(I) Comparison Prediction Point Result vs. the Number of Data Training to Evaluate Accuracy Result.(II) Histogram of Ablation Analysis of Score Predictions of the Classical Technique using various types of EfficientNet.(III) Ablation Study Results.

Figure 1(I)-(d) includes 20,000 data points, Figure 1(I)-(e) includes 100,000 data points, and Figure 1(I)-(f) illustrates the distribution with 200,000 cover and stego data points. In Figure 1(II), we present a histogram representing the ablation analysis of score predictions obtained using different variants of the EfficientNet model within the classical technique. As the amount of training data increases, the prediction accuracy during the evaluation stage also improves. This stage identifies the maximum predicted value of an image in the testing folder, indicating a higher likelihood that the image contains hidden text and thus is predicted as a stego image for DF examiners. Lastly, as depicted in Figure 1(III) for the ablation study results in comparison, we obtained data from blind steganalysis analysis using ablation, as described in Wardhani et al. ([23]). Utilizing EfficientNet B3 [20] as the CNN in the classical layer and incorporating circuit 5 quantum circuits [19] into its parameterized quantum layers, we achieved optimal effectiveness of 97%.

5 Conclusion

This paper explores the potential of hybrid approaches in digital forensics (DF) using four primary paradigms [9]. The study highlights a significant 97% improvement in efficiency when using the EfficientNet B3 CNN architecture compared to traditional methods [23]. This study elaborated more on the accuracy results, compared prediction points versus the number of training data, and provided a histogram of ablation analysis of score predictions of the classical technique using various types of EfficientNet. The findings suggest that classical-quantum hybrid approaches hold significant promise for enhancing various stages of DF investigations, paving the way for future research and development in this domain. This ultimately offers valuable tools for DF examiners.

References

1. Al Balushi, Y., Shaker, H., Kumar, B.: The use of machine learning in digital forensics. In: 1st International Conference on Innovation in Information Technology and Business (ICIITB 2022). pp. 96–113. Atlantis Press (2023)
2. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)

3. Center, I.I.T.R.: 2021 annual data breach report sets new record for number of compromises (2022), https://www.idtheftcenter.org/post/identity-theft-resource-center-2021-annual-data-breach-report-sets-newrecord-for-number-of-compromises, accessed on May 21, 2024
4. Chen, L., Li, T., Chen, Y., Chen, X., Wozniak, M., Xiong, N., Liang, W.: Design and analysis of quantum machine learning: a survey. Connection Science **36**(1), 2312121 (2024)
5. Chutani, S., Goyal, A.: A review of forensic approaches to digital image steganalysis. Multimedia Tools and Applications **78**(13), 18169–18204 (2019)
6. Closser, D., Bou-Harb, E.: A live digital forensics approach for quantum mechanical computers. Forensic Science International: Digital Investigation **40**, 301341 (2022)
7. Cogranne, R., Giboulot, Q., Bas, P.: Alaska# 2: Challenging academic research on steganalysis with realistic images. In: 2020 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–5. IEEE (2020)
8. Dalal, M., Juneja, M.: Steganography and steganalysis (in digital forensics): a cybersecurity guide. Multimedia Tools and Applications **80**(4), 5723–5771 (2021)
9. Dunjko, V., Taylor, J.M., Briegel, H.J.: Quantum-enhanced machine learning. Physical review letters **117**(13), 130501 (2016)
10. Dunsin, D., Ghanem, M.C., Ouazzane, K., Vassilev, V.: A comprehensive analysis of the role of artificial intelligence and machine learning in modern digital forensics and incident response. Forensic Science International: Digital Investigation **48**, 301675 (2024)
11. Garfinkel, S.L.: Digital forensics research: The next 10 years. digital investigation **7**, S64–S73 (2010)
12. Howard, A., Quentin, G., Patrick Frenchie, R.C., Cukierski, W.: Alaska2 image steganalysis (2020), https://kaggle.com/competitions/alaska2-image-steganalysis, [Online; accessed 19-Sep-2023]
13. Karampidis, K., Kavallieratou, E., Papadourakis, G.: A review of image steganalysis techniques for digital forensics. Journal of information security and applications **40**, 217–235 (2018)
14. Kävrestad, J.: Fundamentals of digital forensics. Springer (2020)
15. Michaylov, K.D., Sarmah, D.K.: Steganography and steganalysis for digital image enhanced forensic analysis and recommendations. Journal of Cyber Security Technology pp. 1–27 (2024)
16. Overill, R.E.: Digital quantum forensics: Challenges and responses. In: Future Information Technology: 6th International Conference, FutureTech 2011, Loutraki, Greece, June 28-30, 2011, Proceedings, Part II. pp. 110–114. Springer (2011)
17. Qadir, S., Noor, B.: Applications of machine learning in digital forensics. In: 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2). pp. 1–8. IEEE (2021)
18. Reith, M., Carr, C., Gunsch, G.: An examination of digital forensic models. International Journal of digital evidence **1**(3), 1–12 (2002)
19. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies **2**(12), 1900070 (2019)
20. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
21. Thagard, P.: Philosophy and machine learning. Canadian Journal of Philosophy **20**(2), 261–276 (1990)
22. The European Commission's Joint Research Centre (JRC), Europol's European Cybercrime Centre (EC3), T.E.I.L.: The second quantum revolution: The impact of quantum computing and quantum technologies on law enforcement (2023), https://www.europol.europa.eu/cms/sites/default/files/documents/Europol_Innovation_Lab_Observatory_Report%20-%20The%20Second%20Quantum%20Revolution.pdf, accessed on May 21, 2024
23. Wardhani, R.W., Putranto, D.S.C., Ji, J., Kim, H., et al.: Towards hybrid classical deep learning-quantum methods for steganalysis. IEEE Access (2024)
24. Yang, Y.F., Sun, M.: Semiconductor defect detection by hybrid classical-quantum deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2323–2332 (2022)

Ultra-high Reliable Multi-AP Cooperation Using Securely Compressed Channel-status Information

So-Eun Jeon[1][0000-0003-3700-8885] and Il-Gu Lee[1][0000-0002-5777-4029]

[1] Sungshin Women's University, 2 Bomun-ro 34da-gil, Seongbuk-gu, Seoul 02844, Republic of Korea
soeun727@naver.com, iglee@sungshin.ac.kr

Abstract. Multi-access point (AP) cooperation technique, wherein multiple APs work together to transmit data, has been actively discussed in Wi-Fi 8, which offers improved communication coverage through coordinated transmission using an optimized transmission technique wherein control information is exchanged via multiple APs. However, it has limitations such as high network overheads and security threats owing to the increased transmission and reception of control information between devices. Therefore, this study proposes a hash-based bitmap compression method, called the bitmap compression-based secure channel-management mechanism (BC-SCMM), to minimize the control-information overhead and mitigate attacks in intelligent communication environments. The experimental results showed that BC-SCMM can reduce latency and energy consumption by 74.12 and 105.19%, respectively, while ensuring the confidentiality of channel-status information compared with the conventional non-channel-management mechanism that communicates over a fixed subchannel.

Keywords: Dense Network, Multi-AP Cooperation, Channel Management.

1 Introduction

Wi-Fi 8 (802.11bn) study group, which is currently discussing the standardization of Wi-Fi 8, is conducting research and development with the goal of ultra-high reliability (UHR). A representative example of a UHR technology is multi-access point (AP) coordination, which was introduced with Wi-Fi 7 (802.11be) to improve spectrum efficiency. However, using these advanced cooperative communication technologies to improve network performance and efficiency leads to the issue of higher information-exchange traffic between wireless communication devices, resulting in interference. Existing studies have focused on improving network performance without considering the increased overheads and security threats [1-3].

Therefore, we propose the bitmap compression-based secure channel-management mechanism (BC-SCMM) that converts the channel state of each STA into a simple bitmap and compresses it using a hash function.

2 Related Work

Naderializadeh et al. [4] proposed a multiagent deep RL-based interference management mechanism for distributed resource management and mitigating interferences in wireless networks. This method incorporates an RL agent in each transmitter to learn and make decisions. However, the distributed learning mechanism, wherein all transmitters learn and make decisions, involves high computation complexity. Because conventional ML-based mechanisms still have limitations in terms of complexity, non-ML mechanisms have been actively discussed in recent standards for efficient channel management. Yu et al. [5] proposed a mechanism to manage channel interference by simply specifying and sharing punctured channel information as a bitmap, thereby mitigating the complexity associated with ML-based methods. However, longer bitmaps lead to

increased overheads in high-throughput environments, and the possibility of a malicious attacker manipulating the channel-state information was not considered.

3 BC-SCMM

The proposed BC-SCMM's operating mechanism for efficiently managing network interference, as shown in Fig. 1.

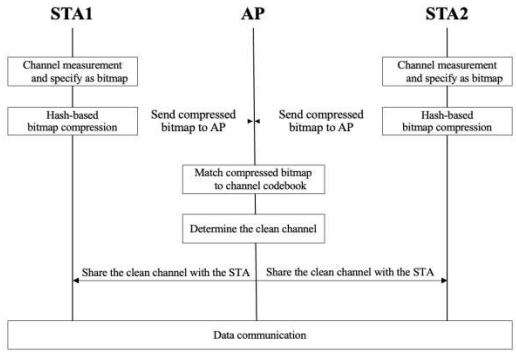


Fig. 1. Operational flowchart of BC-SCMM.

The BC-SCMM is a technique for transmitting channel-state information between STAs and APs in each basic service set (BSS) to ensure communications over an optimal clean channel. First, each STA measures the channel state of each subchannel using the signal-to-interference-plus-noise ratio (SINR) and converts it into a bitmap. When the SINR of a subchannel exceeds a certain threshold, the channel state with low interference is identified as 1. However, if it is below the threshold, a bitmap is generated by identifying it as a subchannel with high interference and specifying it as 0. Subsequently, each STA hashes the bitmap generated for compression and encryption and transmits it to the AP, which decrypts it by comparing it with a codebook wherein the hashed-channel and original bitmaps are matched, and a common clean subchannel identified by the STAs in the BSS is reshared and used to communicate with minimal interference. BC-SCMM proposes using the reserved bit as hashed channel-status information by defining a new encoding rule for the duration/ID field of the existing media access control (MAC) frame, as illustrated in Table 1.

Table 1. Encoding rules for the duration/ID field.

Existing Standard modes	**Bits 0-13**		**Bit 14**	**Bit 15**	**Usage**
	0–32767			0	Duration value applies to all frames except PS-Poll during the contention period and those transmitted during the contention free period using the hybrid coordination function.
	0		0	1	Fixed value based on the point coordination function
	1–16383		0	1	Reserved
	0		1	1	Reserved
	1–2007		1	1	AID in PS-Poll frames
	2008–16383		1	1	Reserved
Newly added mode	**Bits 0-13**		**Bit 14**	**Bit 15**	**Usage**
	Bits 0-12	**Bit 13**			
	1–8191	1	1	1	Hashed channel-status information

In the existing standard, the usage when Bits 14 and 15 of the duration/ID field have a value of 1 is defined in Table 2. In the new encoding rule proposed in this paper, when Bits 13, 14, and 15 of duration/ID have a value of 1, we employ a method that can be expanded while maintaining compatibility with existing standards by using Bits 0–12 as hashed channel-status information.

4 Performance evaluation and analysis

4.1 Evaluation Environment

We employed the conventional non-channel-management mechanism (NCMM) [6], which communicates over a fixed subchannel without measuring the channel status, and the simplified channel-management mechanism (SCMM) [7], which specifies channels using bitmaps but does not perform hash-based compression and encryption, to evaluate the performance improvements owing to BC-SCMM.

Each model was implemented in MATLAB and an orthogonal frequency-division multiplexing (OFDM) environment with a bandwidth of 160 MHz and eight subchannels. Additionally, we assumed that two STAs and APs in one BSS communicate and implemented an environment wherein each STA transmits four 10000-byte data frames per 1 ns. The latency, throughput, and energy consumption were used as the evaluation metrics. The latency was calculated as follows:

$$\text{Latency} = \mathrm{T_{Header}} + \mathrm{T_{Data}} + \mathrm{T_{SIFS}} + \mathrm{T_{ACK}} + \mathrm{T_{DIFS}} \quad (1)$$

T_{Header} and T_{data} represent the times required to transmit the header and data frame, respectively, whereas T_{SIFS}, T_{ACK}, and T_{DIFS} represent the times required to transmit the short inter-frame space (SIFS), acknowledgment (ACK), and distributed coordination function (DCF) inter-frame space (DIFS) frames, respectively. In this scenario, T_{data}, T_{SIFS}, T_{ACK}, and T_{DIFS} are implemented in an environment wherein their transmission times are 80000, 16, 0.1, and 34 ms, respectively. T_{Header} is the total time required to transmit the frame-control, duration, address, and sequence-control fields, as well as the reserved-bit field containing the channel-state information in the header frame. T_{Header} was implemented to vary the duration based on the size of the reserved bit ($T_{reserved\ bit}$) in each model. The throughput was calculated as follows:

$$\text{Throughput} = \frac{\text{Amount of data received successfully}}{\text{Transmission time}} \quad (2)$$

The STAs were implemented in an environment wherein 1 bit is transmitted in 1 ns and each STA transmits four 10000-byte data frames. Additionally, the channel state randomly changes whenever data are transmitted, and the transmission fails when the data are transmitted to a subchannel with an SINR equal to or lower than the threshold and retransmitted up to eight times. The energy consumption was calculated by defining it based on the state using the finite-state machine (FSM) model [8]. The model's state was defined as TX, RX, IDLE, or Collision, wherein 0.0001, 250.5, 0.5, and 1000 mW of energy were consumed per 1 bit, respectively. Additionally, data transmission fails in the Collision state owing to channel interference.

4.2 Evaluation Results and Analysis

Fig. 2 shows the performance of the proposed BC-SCMM and conventional SCMM and NCMM models according to the channel-interference rate.

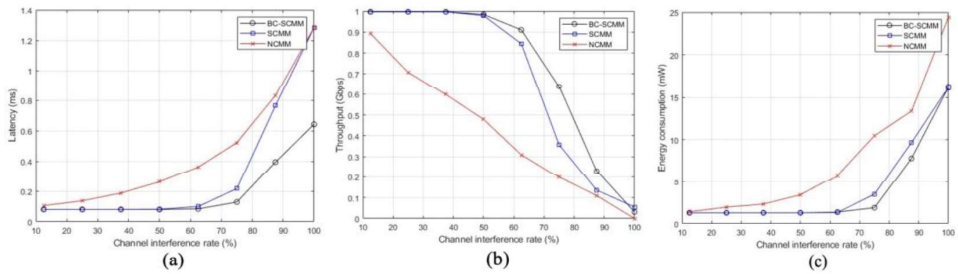


Fig. 2. Performance of proposed BC-SCMM and conventional SCMM and NCMM models : (a) Latency (b) Throughput (c) Energy consumption.

The latency increased linearly and proportionally as the channel-interference rate increased. NCMM model communicating over a fixed channel without measuring the channel state increased the probability of data-transmission failure; thus, more retransmissions resulted in higher latency. By contrast, the proposed BC-SCMM exhibits lower latencies than both models. The throughput is inversely proportional to the channel-interference rate. Because the NCMM is highly likely to communicate over a subchannel containing interference as it communicates over a fixed channel without measuring the channel state, the throughput decreases as the number of retransmissions increases. By contrast, the SCMM and BC-SCMM, which communicate over optimal clean subchannels, exhibit higher throughputs than the NCMM. The energy consumption of the proposed BC-SCMM and conventional SCMM and NCMM models. The energy consumption increased proportionally with the channel interference rate. The NCMM model with many TXs consumed more energy due to retransmissions, whereas the SCMM and BC-SCMM modes, which communicate over clean channels, consumed less energy. In particular, the energy consumption of the BC-SCMM was lower because the size of the header field transmitted by it was smaller.

4.3 Security Analysis

Table 2 presents a security analysis of BC-SCMM and SCMM, which employ channel-measurement information that can be targeted by an attacker.

Table 2. Security analysis of BC-SCMM and SCMM.

Model	Access feasibility	Exposure of channel state
BC-SCMM	O	X
SCMM	O	O

Because both BC-SCMM and SCMM store the channel information in the bitmap channel-measurement field of an unencrypted header, attackers can easily access the channel-state information. However, BC-SCMM performs hash-based de-identification and compression only on the corresponding field section; therefore, even if attackers can access the channel-state information, they cannot obtain the original channel-state information and modify it.

5 Conclusion

This paper proposes a hash-based bitmap compression method, called BC-SCMM, to minimize overheads and security threats in intelligent communication environments. The experimental results demonstrated that BC-SCMM improves efficiency in terms of latency, throughput, and energy consumption compared with conventional models. In a future study, the performance of the BC-SCMM will be verified in a multi-AP environment, wherein interference is higher owing to the control information.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP), and the 2024 Real Challenge program grant funded by Korea Institute of Human Resources Development in Science and Technology(KIRD).

Disclosure of Interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Lanante, L., Kim, J. I., Erkucuk, S., Zhang, J., Baykas, T. (Ofinno): "Cross interference during coordinated spatial reuse. IEEE 802.11 UHR SG (2023).
2. Ajami, A. K., Asterjadhi, A., Patil, A., Ho, D., Cherian, G., Naik, G., Sun, Y. (Qualcomm): Considerations for AP coordination in UHR: coordinated medium access. IEEE 802.11 UHR SG (2023).
3. Guo, J. Y., Li, Y., Huang, G., Gan, M., Yu, R. J. (Huawei): Coordinated spatial reuse for UHR. IEEE 802.11 UHR SG (2023).
4. Naderializadeh, N., Sydir, J.J., Simsek, M., Nikopour, H.: Resource management in wireless networks via multi-agent deep reinforcement learning. IEEE Trans. Wirel. Commun. 20, 3507–3523 (2021). https://doi.org/10.1109/TWC.2021.3051163.
5. Yu, R. J., Hu, M., Gan, M., Jiang, W. (Huawei): Preamble puncture bitmap indication for 20 and 40 MHz OFDMA transmission. IEEE 802.11 TGbe (2020).
6. Nunez, D., Smith, M., Bellalta, B.: Multi-AP coordinated spatial reuse for Wi-Fi 8: group creation and scheduling. In: 2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet), pp. 203–208. IEEE, Island of Ponza, Italy (2023). https://doi.org/10.1109/MedComNet58619.2023.10168857.
7. Jeon, S., Lee, I.: Simplified channel management mechanism for multi-AP cooperation in dense networks. In: Proceedings of the Korean Information Processing Society, pp. 137–140. KIPS, Pyeongchang, Korea (2024).
8. Wu, F., Yang, W., Ren, J., Lyu, F., Yang, P., Zhang, Y., Shen, X.: Named data networking enabled power saving mode design for WLAN. IEEE Trans. Veh. Technol. 69(1), 901–913 (2020). https://doi.org/10.1109/TVT.2019.2952665.

Adversarial Attacks on Behavior Decisions and Countermeasures Using Dropout-Based Fake Node

Seungyeol Lee[1][0009−0005−2954−1243], Youngju Lee[1][0009−0009−6480−8328], and Jaecheol Ha[1][0000−0003−3730−5580]

[1] Hoseo University, Asan-si, Chungcheongnam-do, 31499, South Korea
stl990726@naver.com, lyj04281@naver.com, jcha@hoseo.edu

Abstract. In artificial intelligence industries, DRL(Deep learning-based Reinforcement Learning) have been widely used to provide remarkable performance in terms of user convenience. Despite their widespread use in the autonomous vehicle and robotics industries, DRL model is vulnerable to several adversarial attacks. In this paper, we experimentally show that a behavior decision using a DRL failed during the execution of some adversarial attacks. To counter these attacks, we propose a method in which dropout-based fake nodes are added. The proposed method installs fake nodes in the output layer of the DRL and then randomly removes a part of them using dropout to obfuscate the output values required by the adversarial attack. As a result, we experimentally verified that we weakened the adversarial attack by preventing the attacker from generating adversarial perturbations

Keywords: Reinforcement Learning · Adversarial Attacks · Dropout-based Fake Node

1 Introduction

Recently DRL(Deep learning-based Reinforcement Learning) has been widely used in autonomous vehicles, smart factories, and IoT applications. However, the DRL model is vulnerable to many adversarial attacks where an attacker adds perturbations to the network input data, causing it to malfunction [1]. In particular, some researchers have remotely accessed Autopilot ECUs [2], demonstrating that adversarial attacks can be performed. These attacks can compromise behavior decision systems that use DRL. As a result, incorrect actions may lead to accidents in autonomous vehicles, home network systems, and so on.

To defend against these adversarial attacks, some useful countermeasures such as adversarial training or denoising by adding a new model [3, 4]. However, these methods require more cost and may compromise the original functionality of DRL.

In this paper, we first implement a function for behavior decisions, using HighwayEnv [5] for virtual simulation and validation in highway driving environments. We adopted the DQN(Deep Q-Network) as a deep reinforcement learning algorithm in simulation system. Then, assuming an attacker, we try adversarial attacks such as BIM (Basic Iterative Method) [6] and PGD (Projected

Gradient Descent) [7]. Consequently, these attacks cause the behavior decision algorithm to work abnormally. Furthermore, we propose a dropout-based fake node method to defend against these adversarial attacks [8,9]. In the DRL model using the proposed method, attackers cannot generate threatening adversarial perturbations due to obfuscated nodes.

2 Background

2.1 Adversarial Attacks

The DRL is used in various artificial intelligence fields but are vulnerable to adversarial evasion attacks, which add noise to input images of learning model, causing the networks to malfunction.

As an adversarial attack, BIM(Basic Iterative Method) [6] is a method that computes gradients of the original image to generate small perturbations. The generated noise is then added to the original image to create an adversarial image by continuously using the FGSM(Fast Gradient Sign Method) technique.

$$x^{'}_{t+1} = \text{Clip}\{x + \alpha \cdot \text{sign}(\nabla_x J(\theta, x^{'}, y))\} \tag{1}$$

Here, we compute the gradients of the initial image x using the loss function J, and then multiply these gradients by the intensity of the noise α to create the adversarial image $x^{'}_{t+1}$.

Another adversarial attack PGD [7] is also based on FGSM. In this attack, an attacker repeatedly calculates the gradients of the loss function from the original image x. And he modifies the image by adapting in the opposite direction of the gradients. Finally, the perturbation generated through this attack method is used to produce the adversarial sample x^{n+1}.

$$x^{n+1} = \mathit{\Pi}_{x+S}\left(x^t + \alpha \cdot \sin\left(\nabla_x L(\theta, x, y)\right)\right) \tag{2}$$

2.2 Defensive Dropout

The dropout technique in learning model was proposed to prevent overfitting by randomly dropping out features during model training. In 2018, Wang et al. developed a countermeasure to defend DNN against adversarial attacks using dropout [7]. They apply dropout in test phases to reduce the dependency on specific neurons in a DNN. Their technique was able to achieve the effect of reducing image noise by attenuating the dependency on specific neurons.

Consequently, dropout complicates the adversarial attack process by introducing randomness to the use of neurons during the DNN training and testing phases, and generates the DNN output non-deterministic, making it more robust to adversarial attacks

2.3 Fake Gradient

Feng et al. proposed a fake gradient in a DNN-based image classification system [8]. The fake gradient countermeasure is designed to hide real output information

by generating fake classes and gradients, and prevent adversarial attacks from being performed effectively.

This method adds fake output to the original output layer so that the gradient obtained by the attacker does not interfere with the operation of the original model. Furthermore, it employs a random key to shuffle real and fake nodes, making it difficult for the attacker to identify the real node result.

Fake gradient method demonstrates that it can effectively deceive the attacker and defend against adversarial attacks. The following Fig. 1 shows a fake gradient system architecture.

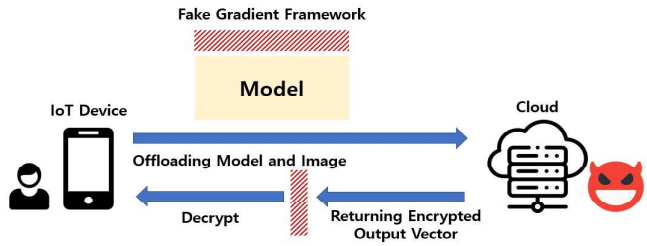


Fig. 1. Fake gradient system architecture

3 Proposed Dropout-based Fake Node Countermeasure

We firstly implement the original DQN model in HighwayEnv in highway driving environments. As a result, this model has an episode during, which is one of evaluation parameters, up to 96.92%. And another parameter, reward, is up to 95.22%.

And then we assume that an attacker launches adversarial attacks such as BIM and PGD with epsilon set to 0.005 to the original DQN. In these cases, an attacker can decrease the episode during and reward to 31.79% and 28.40% respectively. These results show that an attacker can deceive the behavior decision system, leading to incorrect driving actions.

Furthermore, we also implement a dropout-based fake node operation to defeat these adversarial attacks. First, we adopted several fake nodes to the output layer of DQN. And then we obfuscated the output of the fake node by applying randomly dropout and using node shuffling technique according to a random key. An attacker cannot generate malicious perturbations due to obfuscation of the output.

Examples of two perturbation generated by an attacker targeting DQN are in Fig. 2. The first column shows the original state data. The second shows the perturbations generated by proposed dropout-based fake node method. And the third column is perturbations generated by BIM adversarial attack, which does not adopt any countermeasure. We notice that the BIM perturbation attacks can easily trigger adversarial attacks by generating more perturbations than attack applied dropout-based fake node method.

As shown in Table 1, when an adversarial attack is performed, our dropout-based fake node method can achieve an episode during 94.42%. This is a performance improvement of about 62.63% over the BIM perturbation on DQN. We confirmed that the reward evaluation parameter was improved from 28.40% in the case of BIM perturbation attack to 89.38% after applying the countermeasure. The proposed dropout-based fake node method in case of PGD perturbation attack also showed similar performance to that when the BIM attack was used.



Fig. 2. Adversarial perturbations on DQN model

Table 1. Evaluation of DQN with dropout-based fake node

Model	Adversarial Attacks	During	Reward
DQN	N/A	96.92%	95.22%
	BIM [6]	31.79%	28.40%
	PGD [7]	27.06%	24.31%
Our DQN	N/A	96.92%	95.22%
	BIM [6]	94.42%	89.38%
	PGD [7]	93.94%	88.25%

4 Conclusion

Although the DRL for behavior decision has been widely used in autonomous vehicles, smart factories, and IoT applications, it is vulnerable to adversarial attacks. In this paper, we firstly implemented a behavior decision system using DQN and launched adversarial attacks using BIM and PGD perturbation methods. As a result of performing BIM adversarial attack, the episode duration and reward parameters were reduced by up to 31.79% and 28.40%, respectively.

Furthermore, we propose a dropout-based fake node method in a DQN, which is a countermeasure to defeat these adversarial attacks by providing fake output to the attacker. Our experiment on DQN confirmed that the proposed dropout-based fake node method was able to improve the episode during and reward of the DQN to more than 93.94% and 88.25% in the PGD attack. Therefore, the proposed countermeasure can be useful to defeat adversarial attacks in artificial intelligence fields or network systems that require behavior decisions.

Acknowledgments. This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-004)

References

1. Lin, Y., Hong. Z., Liao. Y., Shih. M., Liu. M., and Sun. M.: Tactics of adversarial attack on deep reinforcement learning agents. In: In International Joint Conference on Artificial Intelligence
2. Nie, S., Liu. L., Du. Y., and Zhang. W.: Over-the-air: How we remotely compromised the gateway, BCM, and autopilot ECUs of Tesla cars. In: International Joint Conference on Artificial Intelligence, Black Hat USA (2018)
3. Bai, T., Luo. J., Zhao. J., Wen. B., and Wang. Q.: Recent advances in adversarial training for adversarial robustness. In: IJCAI, pp. 4312—21, 2021
4. Bakhti, Y., Fezza. S. A., Hamidouche. W., and Deforges. O.: DDSA: A defense against adversarial attacks using deep denoising sparse autoencoder. In: IEEE, pp. 160397—407, 2019
5. Leurent, E.: An environment for autonomous driving decisionmaking,: https://github.com/ eleurent/highway-env, last accessed 2018
6. Kurakin, A., Goodfellow. I., and Bengio. S.: Adversarial examples in the physical world. In: Int. Conf. Learning Representations, pp. 1—14, 2018
7. Madry, A., Makelov. A.,Schmidt, L., Tsipras, D.,and Vladu. A.: Towards deep learning models resistant to adversarial attacks. In: ICLR, 2018
8. Wang, S., Wang. X., Zhao, P., Wen, W., Kaeli, D., Chin, P., and Lin. X.: Defensive dropout for hardening deep neural networks under adversarial attacks. In: ICCAD, pp. 1–8, 2018
9. Feng, X., Xie. Y., Ye, M., Tang, X., and Yuan. B.: Fake Gradient: A Security and Privacy Protection Framework for CNN-based Image Classification. In: ACM, pp. 5510–8

Blockchain-based secure vehicle data trading model for vehicle accident reconstruction

Su Jin Shin[1], Youngho Park[2], and Sang Uk Shin[3]

[1] Department of Information Security, Graduate School Pukyong National University, Republic of Korea
inuin9014@pukyong.ac.kr
[2] Electronics and Information Communications Research Center Pukyong National University, Republic of Korea
pyhoya@pknu.ac.kr
[3] Division of Computer Engineering and Artificial Intelligence, Pukyong National University Republic of Korea
shinsu@pknu.ac.kr

Abstract. We propose a secure data trading model that protects the input privacy of Event Data Recorder (EDR) data held by the data owner while providing the computational results required for traffic accident reconstruction using multi-party computation (MPC) techniques. The proposed model ensures fairness through punishment, reward, and reputation mechanisms, and prevents collusion attacks among the components by using a random leader selection technique to select computation nodes.

Keywords: Data trading · MPC · Blockchain · Fairness.

1 Introduction

When a car accident occurs, a car accident reconstruction is performed to determine the cause and fault of the accident, and the most important data is the data recorded in the Event Data Recorder (EDR). EDRs include vehicle trajectories, speed changes before and after the crash, seatbelt use, as well as sensitive information about the vehicle owner or the vehicle's surroundings [1]. EDR data is essential for organizations or individuals performing traffic accident reconstruction, but EDR data owners are reluctant to provide it due to the possibility of leaking their sensitive information. Therefore, in this paper, we propose a secure data trading model using multi-party computation (MPC) techniques to obtain the computational results required for traffic accident reconstruction while protecting the privacy of EDR data. The proposed model provides decentralization, input privacy, fairness, and resistance to collusion attack.

2 Main cryptographic primitive: MOZAIK Architecture

The main algorithms of MOZAIK Architecture are designed based on Secret Sharing technique and AEAD (Symmetric Authenticated Encryption with As-

sociated Data Algorithm) [2]. The main algorithmes of MOZAIK Architecture consist of the following $KeyGen, DistDec$ and $Infer$ algorithmes [3].

- $KeyGen$: Key generation algorithm is an algorithm that generates keys, specifically generating a symmetric key k. Then, using the share algorithm, it generates an appropriate number of key shares $Share(k) \rightarrow k_1, ..., k_n$ for the MPC computation nodes.
- $DistDec$: The distributed decryption algorithm takes a ciphertext c and authentication tag t as public input data, and the key shares $k_i, i \in \{1, ..., N\}$ as private input data. In this algorithm, it verifies the authentication tag t, decrypts the ciphertext c, and returns the plaintext shares $d_1, ..., d_n$. If the authentication tag t is not verified, return $\perp$. This algorithm can be expressed as $DecAndVerify(k_i, c, t) \rightarrow d_n$ based on the AEAD technique.
- $Infer$: This algorithm performs data analysis on input data $d_1, ..., d_n$ that has been secret shared. Then, based on the analyzed values, it generates shares of the inference result, $x_1, ..., x_n$ by performing the computation of certain function f.

3 Proposed Model

3.1 System Architecture

- Data Requester (DR) : For the reconstruction of traffic accidents, the DR deploys smart contracts on the blockchain network, and performs decryption to obtain the computation results of the function it requested.
- Data Owner (DO) : The DO owns the EDR recorded data for the sale and performs the encryption.
- Computation Nodes (CN_i) : The CN_i performs the actual MPC computation based on the function requested by the DR.
- Smart Contract (SC) : The SC randomly selects a CN_i, and determines who is cheating and award appropriate penalities and rewards accordingly.

3.2 Details of Proposed Model

The proposed model follows the following phase, and Fig.1 shows an overview of the proposed model.

1. Request & Response phase
   The DRs distribute requests to the blockchain with information such as the data and the function they want to compute. The DO verifies the request and delivers the appropriate response.
2. Computation node selection phase
   The SC uses the Algorand [4] technique to randomly select computation nodes to participate in the actual computation process. The SC selects N reputable nodes with a signature value less than or equal to P as computing nodes. Finally, the SC distributes a list containing the information of the selected nodes to the blockchain.

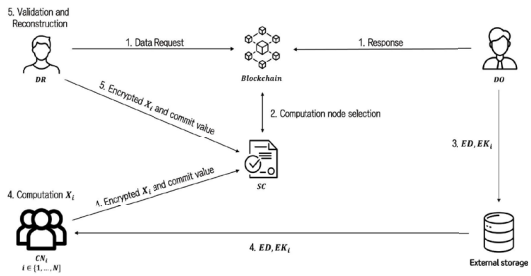


Fig. 1. Overview of the proposed model

3. Data encryption phase
   The DO performs $KeyGen()$ to generate a secret key K and N secret key shares K_i, $i \in \{1, ..., N\}$. The DO encrypts K_i using CN_i's public key and generates the encrypted data ED=$AEAD.Enc(K, D)$ and commit values for EK_i. And then ED and EK_i are stored in external storage. The DO distributes the commit values and the location URI stored in the external storage to the blockchain.
4. Verification of encrypted values and computation phase
   The CN_i obtains ED and EK_i via the URI. The CN_i decrypts EK_i using its own private key and verifies the commit value. If the validation fails, the SC finds the dishonest entity, punishes it, and terminates the trading. If it passes validation, through MPC processing, CN_i decrypts ED first to obtain the share D_i, and then obtains the share X_i of the computation result of the function requested by the DR. And the CN_i encrypts X_i using DR's public key and generates a commit value of the encrypted X_i. The CN_i delivers the encrypted value directly to the DR and the commit value through the blockchain.
5. Verification of computation results and reconstruction phase
   The DR decrypts the encrypted share passed by CN_i using its own private key and verifies the commit value. If there is a mismatch, the SC finds the dishonest entity, punishes it, and terminates the trading. If the comparison results in a match, the DR reconstructs the decrypted value to obtain the complete computation result. After that, the SC pays DO for the data, CN_i for the incentive, increases the reputation of each entity, and terminates the trading.
6. Dispute resolution phase
   The SC performs dispute resolution steps when it receives the following request: (1) a dispute resolution request from CN_i due to a verification failure for EK_i, or (2) a dispute resolution request from DR due to a verification failure for encrypted X_i. Upon receipt of the request, the SC performs the verification step to reveal the dishonest component. Afterward, the SC forfeits the deposit of the identified dishonest entity, reduces its reputation, and terminates the trading.

4 Analysis

The proposed model is based on a public blockchain environment and ensures full decentralization by performing the data trading process with SCs without the construction of trust elements. In addition, it also ensures the following features.

- Input privacy: The data trading process in the proposal model does not expose the original data D provided by the DO. Due to the security of the random selection technique of computation nodes and the security of the applied MPC scheme, the probability of exposing the original data D or the key K is negligible. In addition, the value CN_i obtains by decrypting D is the share of D, and the value DR finally obtains is the computation result of the function he requested. Therefore, input privacy is guaranteed in the proposed model.
- Resistance to collusion attack: In order to carry out collusion attack, it is necessary to select the computation nodes to collude before the trading protocol is executed. However, due to the use of cryptographically secure hash functions and signature techniques, and the security properties of Algorand, it is difficult to predict in advance which nodes are selected as computation nodes among the participating nodes in a blockchain network. Therefore, the probability of success of resistance to collusion attack among participating entities is negligible in the proposed model.
- Fairness: In the proposed model, the SC detects cheating behavior of each component through verification of the commit value. The SC then punishes the cheating component by confiscating its deposit and lowering its reputation. In this way, the proposed model ensures fairness by using punishment and reward mechanisms to encourage honest behavior among entities.

5 Conclusion

In this paper, we proposed a secure and fair data trading model to compute the values required for traffic accident reconstruction while protecting the privacy of EDR data. In addition, by using the MPC technique, we ensure input privacy by protecting the privacy of data owners contained in the EDR data. Computation nodes are selected based on a random leader selection technique of Algorand, to prevent collusion attacks among components. Finally, we ensure fairness by implementing punishment and reward mechanisms to encourage honest behavior of the components.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No.2022R1I1A3063257), and supported by the MSIT (Ministry of Science and ICT), Korea, under the Special RD Zone Development Project (RD) - Development of RD Innovation Valley support program (2023-DD-RD-0152) supervised by the Innovation Foundation.

References

1. Giok Park, Heejin Kang, Joonho Jun, Heejune Kim.: Study on the Improvement of Event Data Recorders through Accident Analysis. Journal Of Auto-Vehicle Safety Association, **13**(4), 66-72 (2021)
2. Rogaway, P.: Authenticated-encryption with associated-data. In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 98-107. (2002).
3. Marquet, E., Moeyersons, J., Pohle, E., Van Kenhove, M., Abidin, A., Volckaert, B.: Secure key management for multi-party computation in mozaik. In 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 133-140, IEEE (2023)
4. Chen, J., Micali, S.: Algorand. arXiv preprint arXiv:1607.01341, (2016)

Secure and Low Latency Transmission Technique for Next Generation Wireless LANs

Sun-Jin Lee[0000-0001-9114-2992] and Il-Gu Lee[0000-0002-5777-4029]

Sungshin Women's University, 02844, Seoul, Korea
iglee@sungshin.ac.kr

Abstract. The development of ultra-low-latency mobile communication technology has led to the rapid expansion of autonomous driving and virtual reality/augmented reality service markets. Accordingly, the low-latency traffic (LLT) transmission and processing technology has been widely investigated as a next-generation Wi-Fi standard technology. However, the existing LLT transmission mechanisms are inefficient owing to the large overhead of the approval process. In this study, LLT is transmitted at a high-speed using the interframe space between the data and ACK frames, which has not been used thus far. Furthermore, we propose an XOR-based high-speed traffic transmission model with a key update mechanism that uses the random number generator and XOR operations to prevent the leakage of LLT information during the transmission process. The experimental results indicated that when the data payload size of non-LLT traffic was 2,000 bytes, the proposed model improved the saturation throughput by approximately 49.9% in comparison with the conventional model. Moreover, the LLT data confidentiality was maintained, and the latency was reduced by approximately 48.6 times.

Keywords: Traffic, High-speed transmission, Data Security

1 Introduction

With the development of Information and Communications Technology (ICT) and the expansion of the non-face-to-face voice and multimedia market owing to COVID-19, the real-time transmission and security of data [1,2] are gaining importance [3,4]. Therefore, the IEEE Study Group, an international standards organization, is developing a technology that can ensure ultra-low latency and ultra-high confidentiality by setting a project authorization request of 802.11bn, which is the next-generation Wi-Fi standard, as the ultra-high reliability standard.

Preemption is a representative technology used for efficient LLT processing [5]. This mechanism provides the opportunity to preempt radio channels and interrupt communication during the preemption period to prioritize specific types of traffic. Preemption can be used to improve the LLT transmission latency. However, if the access point (AP) rejects the request, the delay overhead for LLT processing increases, and the request process requires additional time. Moreover, an encryption mechanism tailored to LLT traffic has not yet been developed. This paper introduces a novel LLT transmission

technique that utilizes the interframe space (IFS) between the data and ACK frames, which has not been implemented thus far. Additionally, to protect information within LLT, a preamble-based key update and XOR operation mechanism is added to prevent threats from eavesdroppers.

2 Related Work

Table 1 summarizes the comparison of the current study with previous studies in terms of low-latency transmission.

Table 1. Conventional LLT transmission technologies

Reference	Contribution	Ultra-low Latency	Data Confidentiality
[6]	- Suggested scenarios where LLT can be transferred - When the station transmits triggering information, the AP transmits LLT information to the station and recommends a method for processing low-latency packets	X	X
[7]	- Proposed a method to facilitate contention only between stations first - Provided degrees of freedom to the setting of contention parameters to avoid conflicts	X	X
[8]	- Proposed a scheduling method to request urgent TXOP sharing to handle LLT based on aperiodic and dynamic events	△	X
[9]	- Proposed a method to inform the existence of stations by using the IFS of uplink physical layer protocol data unit	△	X

Previous studies have reported an increased processing efficiency by adjusting or scheduling the priority of LLT processing; however, if the AP is not accepted after transmitting the triggering signal, the delay increases because it must wait for the processing of all the traffic. Even when preemption is used, there exists an ambiguity of prioritization, leading to overhead when handling LLT.

3 XOR-based High-speed Traffic Transmission Technique

In this study, we propose a XOR-based high-speed traffic transmission technique that transmits low-latency packets using IFS interval, which has not been previously used for data transmission. According to the time–frequency duality, increasing the operating frequency to transmit LLT in the SIFS interval can reduce the transmission time in the time domain rather than extending the transmission bandwidth in the frequency domain.

At present, 802.11 commercial devices implement encryption algorithms, such as the advanced encryption algorithm (AES), in hardware for high-speed encryption. If the packet is not encrypted, the attacker eavesdropping on the channel may estimate the channel based on the long training field (LTF) defined in the standard and identify the

data field value in the packet. To address this, we propose a method in which the receiver generates a random number (RN) sequence based on the random number generator (RNG), which is previously shared with the AP via an LLT channel sounding process and uses it as an LTF. As indicated in Fig. 1, an RN sequence is newly generated for each packet, and the transceiver creates an RN sequence using the RNG private key and seed value exchanged in advance. Subsequently, an LTF comprising an RN sequence is used for channel estimation. Although the channel can be accurately estimated at the transceiver terminal, the eavesdropper cannot estimate the channel because its seed value is unknown. If only the LTF is changed, information leakage is possible because the signal and data fields are transmitted as plain text during communication. Therefore, the external identification threat is blocked by calculating the LTF signal using SIG and DATA field values along with XOR.

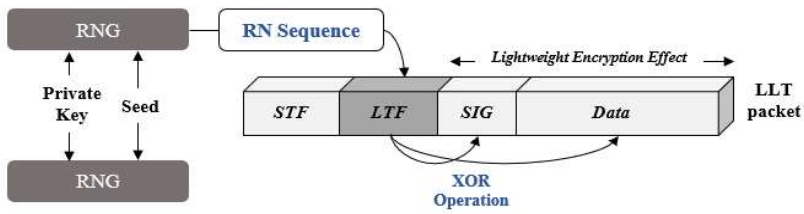
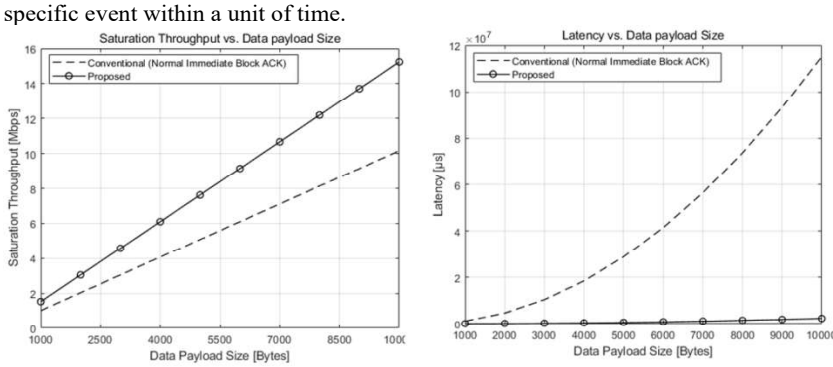


Fig. 1. Preamble-based key update mechanism of the proposed model and the lightweight data protection technique.

4 Evaluation and Analysis

Fig. 2 illustrates the measured saturation throughputs and latencies of IBA and proposed model. The traffic generated from the network was modeled as a poisson distribution, which refers to the probability distribution of the number of occurrences of a specific event within a unit of time.



(a) (b)

Fig. 2. Performance evaluation results regarding (a) saturation throughputs and (b) latencies.

The proposed model improves the saturation throughput performance by approximately 49.9% in comparison with the existing IBA method. In the IBA method, LLT can be received after processing all blocks; however, the proposed model can increase network efficiency while satisfying LLT requirements by efficiently using the IFS section. The conventional model successfully transmits all data and low-latency packets after BA; however, as the proposed model transmits data frames and receives ACK using the SIFS period, the delay is reduced.

5 Conclusions

In line with the developments in ICT, LLT transmission technology is widely researched for the next generation of Wi-Fi standards. In this study, we proposed the utilization of the interframe space, which has not been used in conventional communication standards, for LLT transmission by allocating a wider bandwidth using the time–frequency duality characteristics. Considering that a large amount of confidential information is included in LLT, we proposed a method for generating a new key. Our analysis indicated that data confidentiality can be maintained by performing lightweight XOR operations on the SIG and DATA fields to solve the problems of cryptographic algorithms, such as AES not being applied to the SIFS section. Moreover, the proposed model ensures high security while maintaining low latency.

In this study, we evaluated the performance of the proposed model considering only the SIFS and IBA schemes. In the future, we intend to develop a methodology that minimizes overhead and maximizes throughput by simultaneously transmitting LLT and ACK in the IFS section.

Acknowledgment

This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Jeon, S.E., Lee, S.J., Lee, I.G.: Hybrid in-network computing and distributed learning for large-scale data processing. Computer Networks **226**, 109686 (2023). https://doi.org/10.1016/j.comnet.2023.109686
2. Lee, S.J., Jeon, S.E., Lee, I.G.: A machine learning-enhanced endpoint detection and response framework for fast and proactive defense against advanced cyber-attacks. Soft Computing 1–15 (2024). https://doi.org/10.1007/s00500-024-09727-7
3. Statista, Video Streaming (SVoD), 2024.03., https://www.statista.com/outlook/dmo/digital-media/video-on-demand/video-streaming-svod/worldwide (last accessed 2024/05/01)

4. Sefati, S.S., Halunga, S.: Ultra-reliability and low-latency communications on the internet of things based on 5G network: Literature review, classification, and future research view. Transactions on Emerging Telecommunications Technologies **34**(6), e4770 (2023). https://doi.org/10.1002/ett.4770
5. Intel, Preemption for Low Latency Application, IEEE 802.11-23/0092r0, 2023
6. LG Electronics, Transmission Method of Low Latency Traffic, IEEE 802.11-23/1909r0, 2023
7. Intel, Low Latency Channel Access, IEEE 802.11-23/2126r1, 2023
8. Ofinno, Enhanced Scheduling Method for Low Latency Traffic, IEEE 802.11-23/0378r0, 2023.
9. Realtek, Ultra-reliable PHY elements: Low Latency, Low Collision, Low Power Medium Access, IEEE 802.11-24/0284r1, 2024

Data Preprocessing Technique for Privacy-Preserving Machine Learning

Ye-Rim Jeong[1[0009-0001-0053-6329]], So-Yeon Kim[2[0000-0002-55008-1691]], Il-Gu Lee[1,2[0000-0002-5777-4029]]

[1]Department of Convergence Security Engineering, Sungshin Women's University, Seoul 02844, Korea
[2]Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul 02844, South Korea

{20221130, 220237014, iglee}@sungshin.ac.kr

Abstract. Recently, the use of sensitive personal information in AI-based data learning processes within the medical field has exacerbated the problem of personal information leakage due to membership inference attacks. Conventional methods of masking sensitive information present a trade-off between security and performance. This study introduces a novel approach that classifies data based on sensitivity and replaces sensitive information with GAN-generated synthetic data to mitigate membership inference attacks. The evaluation results showed that compared to the conventional masking method, the proposed method enhanced accuracy by up to approximately 17% while maintaining the same level of security.

Keywords: Membership Inference Attack, Artificial Intelligence, GAN

1 Introduction

While utilizing extensive big data through AI technologies, sensitive personal information is often used indiscriminately [1]. This practice has heightened the risk of privacy violations, particularly through membership inference attacks [2]. Conventional methods to counter these attacks by masking sensitive information reduce the impact of membership inference attacks but degrade model accuracy.

This paper proposes a novel data preprocessing method to mitigate membership inference attacks effectively. By classifying datasets based on their sensitivity and substituting sensitive information with GAN (Generative Adversarial Network)-generated synthetic data, this approach can maintain model accuracy while minimizing the risk of sensitive information leakage.

This paper is organized as follows: Section 2 analyzes related studies, Section 3 proposes a data preprocessing method for machine learning model focusing on privacy protection, Section 4 presents the performance evaluation results, and Section 5 concludes.

2 Related Work

Ying et al.[3] proposed a method to normalize machine learning models using logistic regression and neural network models, coupled with the addition of differential privacy techniques. However, their approach resulted in relatively lower accuracy, precision, and reproducibility as the intensity of the added noise increased.

Jia et al.[4] introduced a defense system with utility loss guarantees against black-box membership inference attacks. Despite its effectiveness, this approach incurs greater overhead compared to existing models.

3 Data Preprocessing Techniques for Privacy-Preserving Machine Learning

Membership inference attacks exploit differences in the outputs of machine learning models when handling training and non-training data. Excluding sensitive information from the training process to defend against membership inference attacks reduces the success rate of these attacks but adversely affects the accuracy of predictive models. To address this issue, this study proposes a method of synthesizing data using GAN, as illustrated in Figure 1, to replace sensitive information with values similar to those in the original dataset.

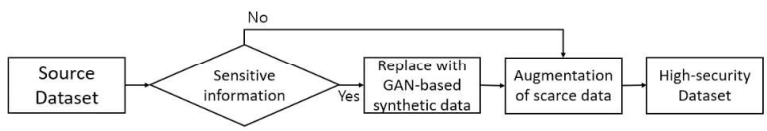


Fig 1. GAN-based data preprocessing techniques

The dataset is categorized into sensitive and non-sensitive information based on its sensitivity. Sensitive information is substituted with synthetic data generated through GAN. Training a machine learning model on this augmented dataset makes it possible to develop a model that demonstrates robustness against membership inference attacks.

4 Evaluation Result and Analysis

This study utilized a cardiac arrest prediction dataset [5] consisting of 21 features and 8,764 records, with a distribution of positive and negative cases approximately in a 1:1.8 ratio. The original dataset was evaluated using a dataset protected by masking method and GAN-based synthesis, and this study identified sensitive information through a comprehensive evaluation of re-identifiability. Sensitive information was identified by a comprehensive assessment of re-identifiability and the sensitivity of health-related data. Therefore, 13 of 21 features were classified as sensitive information, as detailed in Table 1.

Table 1. Classification of data sets by sensitivity

Sensitive Information	Non-sensitive Information
Patient ID, Income, Country, Age, Sex, Diabetes, Family History, Smoking, Obesity, Previous Heart Problems, Medication Use, BMI, Triglycerides	Diet, Cholesterol, Blood Pressure, Heart Rate, Exercise Hours Per Week, Sedentary Hours Per Day, Physical Activity Days Per Week, Sleep Hours Per Day

The proposed method involved the random selection of sensitive features for replacement, and the effectiveness of this approach was evaluated through 1,000 iterations. In the analysis, the x-axis represents the number of sensitive features replaced, while the y-axis denotes the success rate of the attacks and the model's accuracy, respectively.

As illustrated in Fig. 2, the proposed method improved accuracy by up to approximately 17% compared to the model utilizing the masking method. The proposed method maintained accuracy by replacing sensitive information with synthetic data that closely resembles the original data.

Figure 3 illustrates that the success rate of membership inference attacks increased by an average of 0.45% when using the proposed technique compared to the masking approach. However, it shows an average improvement of about 87.9% compared to the model based on the original dataset.

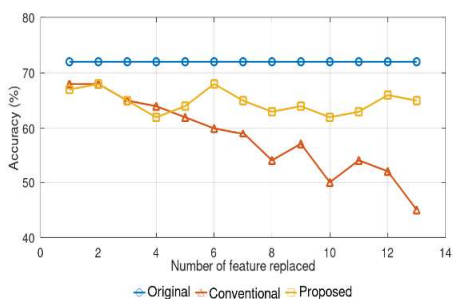


Fig. 2. Accuracy of Model-specific

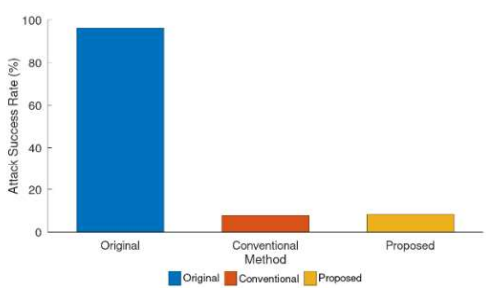


Fig. 3. Success rate of model-membership inference attacks

5 Conclusion

The advent of 5G communication technology has led to the accumulation of vast and diverse data sets, which often include sensitive information. The indiscriminate use of personal information in artificial intelligence-based data learning processes has heightened concerns about personal data leakage through membership inference attacks. This paper proposes a method to maintain model accuracy while mitigating membership inference attacks by substituting sensitive information with data synthesized through

GAN. Evaluation results indicate that this approach improves accuracy by up to approximately 17% compared to masking techniques. Moreover, the proposed method demonstrates a capacity to address membership inference attacks at a level comparable to that of masking techniques

Acknowledgments. This paper was conducted as a result of research on the Industrial Innovation Talent Growth Support Project (RS-2024-00415520) of the Ministry of Trade, Industry and Energy and the Korea Institute of Industrial Technology Promotion in 2024 and the ICT Innovation Talent 4.0 Project of the Ministry of Science and ICT and the Information and Communication Planning and Evaluation Institute. (No. IITP-2022-RS-2022-00156310)

References

1. Castiglioni, I., Rundo, L., Codari, M., Di Leo, G., Salvatore, C., Interlenghi, M., Francesca, G., Andrea, C., Claudia D'Amico, N., and Srdanelli, F. (2021). AI applications to medical images: From machine learning to deep learning. *Physica medica*, *83*, (pp. 9-24).
2. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. (2022, May). Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)* (pp. 1897-1914). IEEE.
3. Ying, Zuobin, Yun Zhang, and Ximeng Liu. (2020, November). Privacy-preserving in defending against membership inference attacks. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice* (pp. 61-63).
4. Jia, J., Salem, A., Backes, M., Zhang, Y., and Gong, N. Z. (2019, November). Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security* (pp. 259-274).
5. Heart Attack Risk Prediction Dataset. Sourav Banerjee. https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset

Mission-centric Cyber Resilience Assessment using Transformer Models in Tactical Modeling and Simulation

Seungwoon Lee[1][0000−0002−7939−1889] Haerin Kim[1][0000−0002−7084−1230] and Jaesik Kang[1][0000−0000−0000−0000]

LIG Nex1, Seongnam 13486, Republic of Korea
{seungwoon.lee,haerin.kim,jaesik.kang}@lignex1.com
https://www.lignex1.com

Abstract. In Network-Centric Warfare (NCW), cyber resilience is crucial for operational success. This study proposes a mission-centric cyber resilience assessment method using modeling and simulation (M&S) to address limitations of existing methods. We introduces Transformer-based log similarity analysis to compare mission scenarios with and without cyber threats. This approach offers a context-focused, comprehensive analysis of cyber resilience, potentially enhancing cyber defense and improving military operations in modern battlefields.

Keywords: Cyber Resilience · Cyberwarfare · Modeling & Simulation · Cyber Weapon System

1 Introduction

In modern battlefield with the spread of Network-Centric Warfare (NCW), the ability to recover from cyber threats has become crucial for operational success. While traditional security approaches focused on prevention and detection against cyber threats, cyber resilience emphasizes continuous operations. This paradigm recognizes that cyber threats are inevitable and focuses on minimizing risks instead. Consequently, cyber resilience has emerged as an essential concept and ability for enhancing cyber defense capabilities, offering a more comprehensive view of a system's ability to anticipate, withstand, recover from, and adapt to cyber threats including adverse conditions, stresses, attacks, or compromises [6].

Existing methods for assessing cyber resilience face several significant limitations [1, 4]. The complexity and emergent properties of cyber environments make it difficult to understand the characteristics of the entire system considering only individual components, complicating the development of appropriate evaluation metrics [5]. Furthermore, cyber resilience takes on different meanings depending on the context of specific missions and operations, making it challenging to apply generalized assessment methods. The costs associated with data collection and processing during the evaluation process also pose significant constraints.

To address these challenges, we propose a methodology for a mission-centric cyber resilience assessment, using modeling and simulation (M&S) widely adopted

across various industries, especially in military domains [8, 3, 5, ?,2]. Our approach is based on the principle that robust cyber resilience minimizes the impact of cyber threats on mission performance. By concentrating on the context of mission rather than complex system attributes, we provide a measurable index of cyber resilience called "log similarity". The differences in log sequences during simulation can indicate the impact of cyber threats on mission and how effectively a system endures cyber threats, thus representing the comprehensive level of cyber resilience. Also, we introduce a Transformer-based log similarity analysis that leverages the Transformer's self-attention mechanism to weigh the importance of different log entries,enabling a more tailored analysis of missions.

2 Background

2.1 Cyber Resilience and Assessment Metrics

Cyber resilience is the ability to anticipate, withstand, recover from, and adapt to cyber-related adverse conditions [6]. Unlike cybersecurity, it considers both technical aspects and business continuity. Assessing cyber resilience is challenging due to: 1) system complexity making it hard to understand the whole based on individual parts, 2) its context-dependent nature tied to specific missions and threats, and 3) difficulty in developing metrics that are reproducible, cost-effective, timely, and meet stakeholder needs.

Various metrics are used to assess cyber resilience, including measurements, computations from measurements, scores, and qualitative or semi-quantitative assessments. These metrics are used to evaluate different aspects of cyber resilience, including security metrics, system resilience metrics, and risk metrics [10]. However, the boundaries between different metrics are not well-defined. Consequently, metrics that represent a wide range of scenarios can be highly complex and their evaluation in real-world environments can be infeasible.

2.2 Modeling & Simulation for Cybersecurity and Cyber Resilience

Several techniques are being researched to prepare for cyber security threats, each with its specific purpose including Penetration Testing (Pen Test), Red Team Automation (RTA), Breach and Attack Simulation (BAS), and M&S. Pen Test involves performing actual attacks to discover vulnerabilities. RTA focuses on automating the processes used by red teams, including custom scripts and tools. BAS uses pre-built scenarios to test security controls against known attack patterns. M&S utilizes computer modeling to virtually reproduce and analyze various attack scenarios, enabling pre-assessment of security strategies.

Among these techniques, M&S is particularly well-suited for assessing cyber resilience [7]. While M&S inherently encodes assumptions about the system, operational environment, and threat landscape, which can be a limitation, it simultaneously allows to abstract complex situations to various models. Furthermore, M&S in the military domain is a mature and widely used technique,

employed for decades in various aspects of military planning and operations. This maturity offers the advantage of easily integrating cyber resilience assessment into existing M&S tools and frameworks, leveraging established methodologies and expertise. It is particularly applicable to cyber resilience evaluations that require mission context, enabling more realistic and meaningful assessments that consider actual operational environments and mission objectives.

3 Methodology

This research proposes a new method to assess mission-centric cyber resilience using MS log data and Transformer models. It aims to provide a context-aware index for evaluating cyber resilience in tactical environments. The approach, shown in Fig. 1, uses Tactical MS based on DES (discrete event simulation) and ABS (agent-based simulation) to evaluate mission performance. It compares simulation logs from scenarios with and without cyber threats, focusing on the mission-centric aspect of cyber resilience.

By analyzing the similarity in these logs, we can quantify the impact of cyber threats on mission performance and assess the level of cyber resilience in the tactical environment. As tactical simulations become more sophisticated, incorporating various mission elements and cyber threat vectors, the complexity and diversity of these logs increase, necessitating advanced analysis techniques.

We employ a Transformer-based architecture to compare and analyze the simulation logs. Transformer models [9] excel at processing sequential data and capturing complex patterns, making them ideal for identifying subtle changes in mission performance and system behavior under cyber threats. Unlike traditional statistical methods or simple neural networks, Transformers can effectively capture long-range dependencies in the log data, allowing for a more subtle understanding of how cyber threats propagate through a system over time.

The key innovation of this methodology lies in its ability to express mission-centric cyber resilience as quantitative metrics. The Transformer's self-attention mechanism allows us to assign greater importance to critical mission-related log entries, accurately capturing the core impact of cyber threats on mission outcomes. This approach enables the identification of nuanced effects on mission performance that might be overlooked by traditional analysis methods.

4 Conclusion

This study proposes a method for assessing mission-centric cyber resilience in tactical environments. The core is a Transformer-based log similarity analysis technique that precisely measures the impact of cyber threats on mission performance. Using the Transformer's self-attention mechanism to assign importance to mission-critical elements, it identifies subtle effects difficult to capture with existing methods. This approach can be easily integrated into existing MS tools. However, it has not been validated in real-world environments, which remains a challenge for future research.

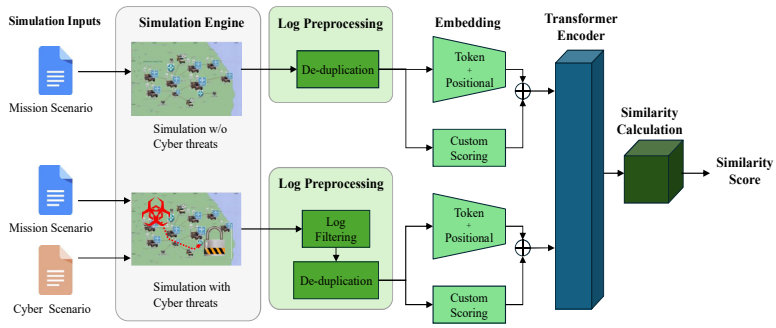


Fig. 1. Overall Architecture

References

1. Alevizos, L., Ta, V.T.: Ttp-based cyber resilience index: A probabilistic quantitative approach to measure defence effectiveness against cyber attacks. arXiv preprint arXiv:2406.19374 (2024)
2. Cam, H.: Cyber resilience using autonomous agents and reinforcement learning. In: Artificial intelligence and machine learning for multi-domain operations applications II. vol. 11413, pp. 219–234. SPIE (2020)
3. Jang, J., Kim, K., Yoon, S., Lee, S., Ahn, M., Shin, D.: Mission impact analysis by measuring the effect on physical combat operations associated with cyber asset damage. IEEE Access **11**, 45113–45128 (2023). https://doi.org/10.1109/ACCESS.2023.3273612
4. Kott, A., Linkov, I.: To improve cyber resilience, measure it. Computer **54**(2), 80–85 (2021). https://doi.org/10.1109/MC.2020.3038411
5. Ligo, A.K., Kott, A., Linkov, I.: How to measure cyber-resilience of a system with autonomous agents: Approaches and challenges. IEEE Engineering Management Review **49**(2), 89–97 (2021). https://doi.org/10.1109/EMR.2021.3074288
6. NIST: Systems Security Engineering - Vol 2: NiST SP 800-160. CreateSpace Independent Publishing Platform, North Charleston, SC, USA (2018)
7. Simone, F., Akel, A.J.N., Gravio, G.D., Patriarca, R.: Thinking in systems, sifting through simulations: A way ahead for cyber resilience assessment. IEEE Access **11**, 11430–11450 (2023). https://doi.org/10.1109/ACCESS.2023.3241552
8. Smith, S., Raio, S., Erbacher, R., Weisman, M., Parker, T., Ellis, J.: Quantitative measurement of cyber resilience: A tabletop exercise. DEVCOM Army Research Laboratory (US): Adelphi, MD, USA (2022)
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017), https://arxiv.org/pdf/1706.03762.pdf
10. Xu, S.: Sarr: a cybersecurity metrics and quantification framework (keynote). In: Science of Cyber Security: Third International Conference, SciSec 2021, Virtual Event, August 13–15, 2021, Revised Selected Papers 4. pp. 3–17. Springer (2021)

Enhancing Cyber Warfare Training: A Cost-Effective Approach to Multi-Host Network Simulations with Container-Based Cyber Range

Haerin Kim[0000−0002−7084−1230] and Seungwoon Lee[0000−0002−79390−1889]

LIG Nex1, Seongnam 13486, Republic of Korea
(haerin.kim, seungwoon.lee)@lignex1.com

Abstract. The primary challenges in cyber ranges involve the costs associated with environment setup and the development of complex scenarios. Due to the nature of training, which requires scenario-specific configurations across multiple computers, various virtualization types are utilized. However, research is still limited on cyber ranges that use container virtualization to construct network topologies for multi-node networks. Single-node environments are limited in their capacity to emulate the intricate scenarios encountered in real-world cyber environments. We design a cyber range environment capable of constructing scenarios that execute from source nodes to destination nodes in multi-host networks, and provide detailed implementation methods. This environment will enable attacks such as lateral movement, which is not possible in traditional single-node training environments. Each attack unit action in the scenario is mapped to the MITRE ATT&CK framework, ensuring more valid scenarios that are educationally appropriate.

Keywords: Container based cyber range · Multi-host network topology · MITRE ATT&CK.

1 Introduction

Currently, the main challenge in cyber ranges is creating the virtual environment [4]. Nakata et al. [1] demonstrated the superiority of container types compared to other virtualization types, while Mills et al. [5] containerized various systems and confirmed their effectiveness. However, there is still a lack of research on environments capable of running complex multi-node scenarios. Therefore, we design node topologies using attack scenarios that require multiple hosts as examples.

2 Multi-Node Cyber Range

We focus on configuring vulnerable host environments suitable for complex scenarios to enable training based on missions such as "Exfiltrate specific information from a database in a different subnet!" as shown in Figure 1. For scenario generation, we refer to the automated scenario generation method proposed by

Schreuders et al. [3]. After the training, as part of the best practice for the scenario, we map the CVEs defined in the scenario to the threat techniques of the MITRE ATT&CK framework using Bron, as proposed by Hemberg et al. [?,?]. There are tactics that cannot be performed on a single host. For example, in the

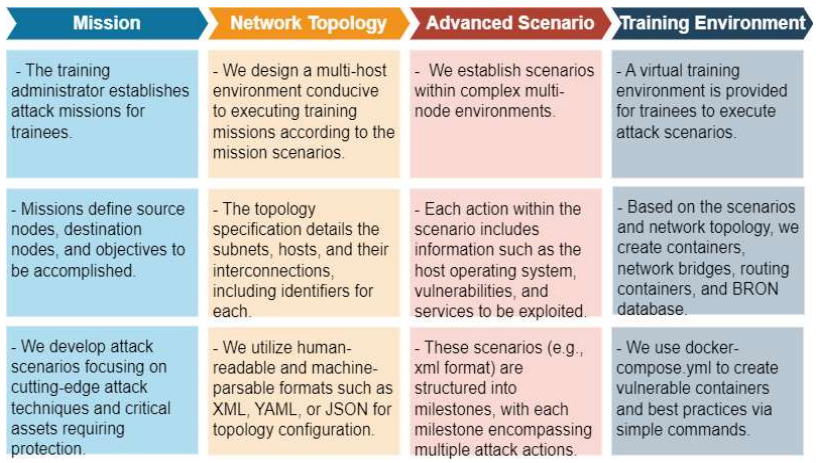


Fig. 1. This is an explanation of the overall process for generating the virtual training environment of the cyber range proposed in this paper.

MITRE ATT&CK Enterprise attack framework, tactics such as Lateral Movement, Collection, and Discovery are relevant. Lateral Movement, a technique used by attackers to move to other systems within the network, is difficult to employ in a single-host environment due to the absence of a network to move within. Unlike single-host environments, multi-host virtual environments require router and switch functionalities for network configuration. Figure 2 demonstrates how to implement switching using Docker [2], the de facto standard for container virtualization.

Table 1. Example of a Complex Attack Scenario in a Multi-Node Environment

Step	Source	Target	Attack Scenario	Technique
1	Attacker	Subnet1 Client	Send phishing email	T1566
2	Subnet1 Client	Subnet1 Client	Malware infection (shell acquisition)	T204.002
3	Subnet1 Client	Subnet2 Client	Network scan to identify Subnet2 Client	T046
4	Subnet2 Client	Web Server	Network scan to confirm web server existence	T046
5	Subnet2 Client	Web Server	Discover web server vulnerability	T1595.002
6	Subnet2 Client	Web Server	Exploit web server (shell acquisition)	T1190
7	Web Server	Web Server	Search for database account within web server	T1087.001
8	Web Server	Database Server	Connect to database server	T1021
9	Database Server	Database Server	Search for data in database server	T1005
10	Database Server	Database Server	Delete existing data on database server	T1485

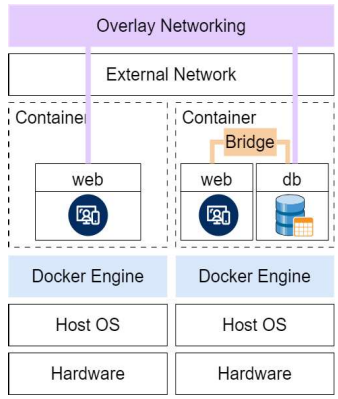


Fig. 2. This explains the network configuration method for Docker containers. For the switch functionality required in multi-host environments, Bridge networking is used within the same host, while overlay networking is used between different hosts.

Multi-host environments are essential for training participants in attack scenarios occurring in advanced settings. Table 1 illustrates an example of such an attack scenario. The scenario, consisting of 10 stages, depicts an attacker deleting data from a database server through a phishing email. The example network topology comprises three subnets, with the Web Server and Database Server located in Subnet3.

This process can occur in a network topology as shown in Figure 3. For a realistic network configuration, a router is placed at the center, with switches and firewalls strategically positioned.

3 Conclusion

We have designed a multi-node environment using cost-effective containers. Container environments have been proven to use significantly less memory and CPU compared to host virtualization and hypervisor virtualization types, while also demonstrating higher vulnerability reproduction rates. Among these, we propose the necessity and development potential for attack scenarios involving multiple hosts. Addressing the most complex issues in cyber ranges - environmental costs and scenario development - we have provided detailed explanations of realistic attack scenarios that could actually occur, and designed corresponding network topologies. This process requires actual implementation and automation, referencing existing research on single-host cyber ranges. In the future, we will contribute to the suitability of cyber ranges through the implementation and experimentation of automation programs for configuring vulnerable environments.

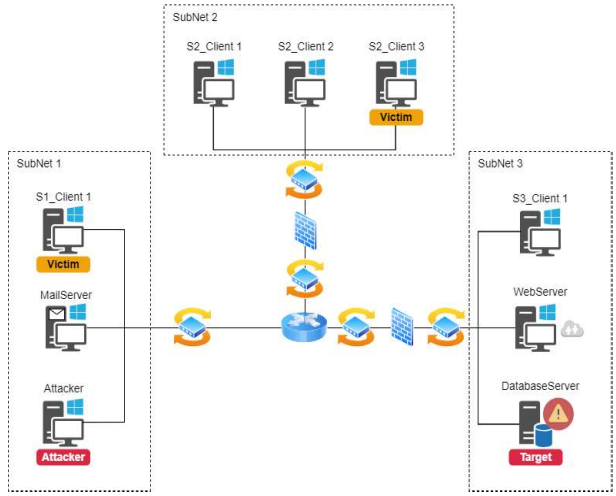


Fig. 3. This is the network topology to which attack scenarios occurring between multiple hosts will be applied. Centered around the router in the middle, three subnets are configured. The attacker's PC in Subnet1 accesses the Database Server in Subnet3 by traversing through Client1 in Subnet1 and Client3 in Subnet2.

References

1. Nakata, Ryotaro, and Akira Otsuka. "Cyexec*: A high-performance container-based cyber range with scenario randomization." IEEE Access 9 (2021): 109095-109114.
2. Docker Inc.: Docker Documentation. Docker Inc., San Francisco, CA (2023), `https://docs.docker.com/`
3. Schreuders, Z. Cliffe, et al. "Security scenario generator (SecGen): A framework for generating randomly vulnerable rich-scenario VMs for learning computer security and hosting CTF events." 2017 USENIX Workshop on Advances in Security Education (ASE 17). 2017.
4. Zhou, Shaohong, et al. "Automating the Deployment of Cyber Range with OpenStack." The Computer Journal 67.3 (2024): 851-863.
5. Mills, Alan, Jonathan White, and Phil Legg. "GoibhniUWE: A Lightweight and Modular Container-Based Cyber Range." (2024).

Defense Mechanism Using Purifier and Detector Against Adversarial Patch Attacks for Unmanned Combat Systems

Dong-U Kang[1], Jaesik Kang[2] and Seungwoon Lee[2]

[1] Division of Smart Security, Korea University. Republic of Korea
[2] LIG NEX1. Republic of Korea
redr1102@korea.ac.kr,{jaesik.kang,seungwoon.lee}@lignex1.com

Abstract. Object recognition models will be importantly used in unmanned combat systems. However, adversarial patch attacks could be executed, leading the models to misrecognize objects. Defensive methods such as adversarial detectors and robust models have already been introduced, but these methods are not sufficient to be directly applied to unmanned combat systems. This paper suggests a defensive system against adversarial example attacks using ideas from both adversarial detectors and robust models.

Keywords: AI Security, Object Recognition, Adversarial Patch, Defense System, Detector, Purifier.

1 Introduction

Recently, Artificial Intelligence (AI) technique has been applied to various fields such as education, finance, security and national defence such as military supply, supporting commander's decision, unmanned combat system[1], and its development is accelerating. Specifically, object recognition model has been spotlighted in unmanned combat system.

For unmanned combat systems, a very high level of autonomy is required[2]. A robust object recognition model is essential to prevent and block these attacks. It must be capable of dealing with malicious attacks. The adversarial example attack is one of the inference-time attacks[3]. It aims to induce an error in the target AI that is desired by the attacker. The attacker achieves this goal by adding a malicious perturbation to the input data, injecting a malicious patch, rotating the input data, or performing some transformation.

The adversarial patch attack is useful for inducing errors in object recognition models in unmanned combat systems like drones [4]. The attacker can physically attach the adversarial patch to an object to prevent it from being recognized by the drone's AI. attacker are providing robustness at the generation level[5] to overcome physical influences such as patch contamination and light reflection, enabling the patches to induce errors in the model.

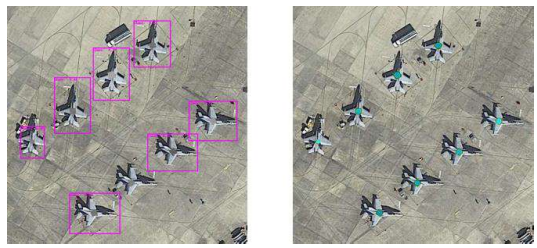


Fig. 1. Images of jet fighter with detections(left), The Jets with adversarial patches are not detected(right) [4]

Recent research has been conducted to detect, block, and recover from adversarial patch attacks[6, 7] If these proven techniques are applied effectively and simultaneously, it will be possible to build a model that can better defend against adversarial patch attacks. In this paper, we suggest a defense mechanism against adversarial patch attack for unmanned combat system.

2 Related work

Purifier aims to remove the adversarial features of the adversarial example [6]. It is applied before the main object recognition model so that the main model receives a clean (non-adversarial) image as input. This method can proactively respond to adversarial examples, but the accuracy of object recognition decreases because the original image is transformed.

Adversarial detector aims to detect adversarial example attacks [7]. This is a passive method. If an adversarial example is detected, it does not input the image to the object recognition model. This method is not appropriate for unmanned combat systems due to the lack of response when an adversarial example is detected.

3 Suggested Defense Mechanism Against Adversarial Patch Attacks for Unmanned Combat Systems

We suggest a defense mechanism against adversarial patch attacks that can be applied to unmanned combat systems. Unlike other object recognition models, such as those used in autonomous vehicles, unmanned combat systems should be able to recognize situations immediately with minimal intervention from the command center to avoid wiretapping through the communication network.

In this context, the unmanned combat system needs to defend against adversarial patch attacks itself and recognize objects correctly. The defense methods mentioned above do not fit these purposes. The Purifier requires higher computational capability and lowers accuracy, while the adversarial detector is not an appropriate defense method without human intervention.

3.1 Defense Mechanism with Universal Detector

Our idea is when the detector alert for an adversarial to purify the input image and execute the object recognition model again.

Terminology and Definition. x : Original image from camera, $\mathrm{Ai}(\cdot)$: Object recognition model, $\mathrm{Detect}(\cdot)$: Detecting an adversarial example. $\mathrm{Detect}(x) = 1$ if x has adversarial feature. $\mathrm{Purify}(\cdot)$: Purifying the input image.

Defense Mechanism with Universal Detector

```
1:  procedure Defense( image x ) =
2:    while(1)
3:      if( Detect(x) == 1 )
4:        increase i
5:        Purify(x, i)
6:        return Defense(x)
7:      else
8:        return Ai(x)
9:      endif
10:   endwhile
11: endprocedure
```

It runs purifier only if adversarial example detected. Compared to running the purifier always, it is more effective at the aspect of computation resource. Furthermore, it partially solves decreasing accuracy problem of purifier.

As we mentioned, adversarial example attack will be performed as the adversarial patch attack, which transforms only some part of image. So, activating the purifier only for the adversarial part makes sense. There is developed idea on the following chapter.

3.2 Defense Mechanism with Spatial Detector

Terminology and Definition. $\mathrm{SpatialDetect}(\cdot)$: Detects the adversarial patch in the input image. If adversarial patch is not detected, it returns 0. Otherwise, it returns the location of the detected adversarial patch. It would be achievable by finding candidate region with informative region selection similarly to general object recognition model and applying previous universal detector.

Defense Mechanism with Spatial Detector

```
1:  procedure SpatialDetect( image x ) =
2:    Boxes ← InformativeRegionselection(x)
3:    for( Box in Boxes )
4:      if( Detect(Box) == 1 )
5:        return Box
6:      endif
7:    endfor
8:  endprocedure
```

```
procedure spatialDefense (image x ) =
  While(1)
    if( SpatialDetect(x) == 0 )
      return Ai(x)
    else
      increase i
      Purify( SpatialDetect(x), i )
      return SpatialDefense(x)
    endif
  endwhile
endprocedure
```

4 Conclusion and Future work

The suggested mechanism requires two preceding techniques. First, a spatial detector is needed which can recognize not only whether an adversarial patch exists in the input image, but also where it is located. Second, a purifier with adjustable intensity is required. One might expect that simply repeating the purifying process would be sufficient, but this has not been proven effective. Adversarial patch attacks are effective in inducing errors in object recognition models. We suggest a defense mechanism for unmanned combat systems. Our design does not require excessive computational capability and provides autonomy, both of which are essential for unmanned combat systems. Our design incorporates two techniques adapted from existing defense methods. Our future work will involve implementing these techniques and testing our design.

References

1. J. C. Costa, T. Roxo, H. Proença, et al. How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses. IEEE Access 12, 61113-61136(2024)
2. JK Choi, YT Lee, HS Park, et al. Challenges to the Development of Manned and Unmanned Combat Systems. 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), 2362-2364(2022)
3. ZQ Zhao, P Zheng, S Xu, X Wu. Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems 30(11), 3212-3232(2019)
4. R Den Hollander, A Adhikari, I Tolios, et al. Adversarial patch camouflage against aerial detection. Artificial intelligence and machine learning in defense applications II 11543, 77-86(2020)
5. STK Jan, J Messou, YC Lin, et al. Connecting the digital and physical world: Improving the robustness of adversarial attacks. Proceedings of the AAAI Conference on Artificial Intelligence 33, 962-969(2
6. Guang Lin, Chao Li, Jianhai Zhang, et al. Adversarial Training on Purification (AToP): Advancing Both Robustness and Generalization. arXiv, 2401.16352(2024)
7. J Liu, A Levine, CP Lau, et al. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14973-14982(2022)

Poster: A Robust Counting Sketch for Data Plane Intrusion Detection

Sian Kim[1] and DaeHun Nyang[1]

Department of Cyber Security, Ewha Womans University, Seoul, Republic of Korea
{ksy60a,nyang}@ewha.ac.kr

Title: A Robust Counting Sketch for Data Plane Intrusion Detection

Authors: Sian Kim, Changhun Jung, Rhongho Jang, David Mohaisen, DaeHun Nyang

Venue: 30th Annual Network and Distributed System Security Symposium, NDSS 2023. The Internet Society, 2023

Full Reference: Kim, Sian, et al. "A Robust Counting Sketch for Data Plane Intrusion Detection." 30th Annual Network and Distributed System Security Symposium, NDSS 2023. The Internet Society, 2023.

DOI: 10.14722/ndss.2023.23102

Link: https://www.ndss-symposium.org/wp-content/uploads/2023-102-paper.pdf

Abstract: Demands are increasing to measure per-flow statistics in the data plane of high-speed switches. However, the resource constraint of the data plane is the biggest challenge. Although existing in-data plane solutions improve memory efficiency by accommodating Zipfian distribution of network traffic, they cannot adapt to various flow size distributions due to their static data structure. In other words, they cannot provide robust flow measurement under complex traffic patterns (e.g., under attacks). Recent works suggest dynamic data structure management schemes, but the high complexity is the major obstruction for the data plane deployment. In this paper, we present Count-Less (CL) sketch that enables robust and accurate network measurement under a wide variety of traffic distributions without dynamic data structure update. Count-Less adopts a novel sketch update strategy, called *minimum update* (CL-MU), which approximates the conservative update strategy of Count-Min for fitting into in-network switches. Not only theoretical proof on CL-MU's estimation but also comprehensive experimental results are presented in terms of estimation accuracy and throughput of CL-MU, compared to Count-Min (baseline), Elastic sketch, and FCM sketch. More specifically, experiment results on security applications including estimation errors under various skewness parameters are provided. CL-MU is much more accurate in all measurement tasks than Count-Min and outperforms FCM sketch and Elastic sketch, state-of-the-art algorithms without the help of any special hardware like TCAM. To prove its feasibility in the data plane of a high-speed switch, CL-MU prototype on an ASIC-based programmable switch (Tofino) is implemented in P4 language and evaluated. In terms of data plane latency, CL-MU is faster than FCM, while consuming less resources such as hash bits, SRAM, and ALU of a programmable switch.

Multi-Link Operation-Based Group Selection Technique for Enhancing Privacy and Efficiency in Federated Learning

So-Yeon Kim[1[0000-0002-5508-1691]], Na-Eun Park [1[0009-0008-0939-5762]]
Yu-Ran Jeon[1[0009-0001-4149-9288]] and Il-Gu Lee [1[0000-0002-5777-4029]]

[1] Department of Future Convergence Technology Engineering Sungshin Women's University, Seoul 02844, South Korea
{220237014, 220237016, 220247016, iglee}@sungshin.ac.kr

Abstract. Owing to the considerable advancements in information and communications technologies, the digital healthcare industry and technologies that provide personalized medical and healthcare services have garnered significant attention. A federated learning (FL) method that can learn without sharing data with a server is essential to maintain data privacy while providing artificial intelligence-based digital healthcare services. However, FL faces challenges such as deterioration of the global model performance and increased communication costs if the data on mobile devices are non-independent and non-identically distributed (non-IID). As wearable devices employed for digital healthcare are wireless and lightweight, it is crucial to minimize the communication costs. To address this issue, this study introduces Multi-Link Operation-Based Group Selection Technique (MLO-GS), which reduces communication costs while enhancing privacy in FL environments wherein the data on mobile devices are non-IID. The evaluation results showed that compared to the conventional federated averaging method, MLO-GS enhanced the initial convergence speed by 25% on average and reduced communication costs by approximately 45% while enhancing privacy.

Keywords: Federated Learning, Group Selection Technique, Data Privacy, Non-IID

1 Introduction

Owing to the transformative power of information and communication technologies, the integration of artificial intelligence into the medical field has garnered significant attention [1] and has revolutionized the digital healthcare industry, paving the way for technologies that offer personalized medical and healthcare services [2]. Because digital healthcare data include extremely sensitive personal data, a centralized learning (CL) method can lead to leakages when sharing these data on a server and training the model. Federated learning (FL) has recently been employed to address this issue as it allows training models on devices where the data are stored and do not require sending them

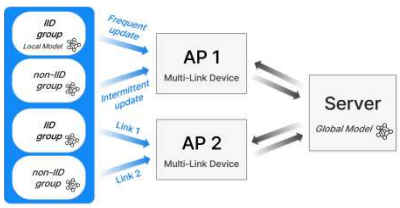


Fig. 1. Framework of Multi-Link Operation-Based Group Selection Technique.

to a server. However, FL has the problem of degraded global model performance and increased communication costs in environments wherein the data on devices are non-independent and non-identically distributed (non-IID). Although the previous method has been proposed to reduce communication costs by selecting a subset of devices that participate in FL [3], the random selection of devices to participate in the FL can lead to partial data representation and information loss. Furthermore, if the number of participating devices decreases, the frequency of information transmission increases, which increases the communication costs. Adopting an FL method is essential to ensure data privacy and provide artificial intelligence-based digital healthcare technologies and services. However, wearable and other digital healthcare devices are wirelessly connected and lightweight, making it crucial to reduce the communication costs in FL environments. Therefore, this study proposes multi-link operation-based group selection technique (MLO-GS) to address the problems of decreased convergence speed and increased communication costs associated with non-IID data. In MLO-GS, devices are grouped into independent, identically distributed (IID), and non-IID groups, and their participation in FL differs based on their group. The IID group frequently updates the model, whereas the non-IID group updates it intermittently.

2 Methodology

Fig. 1 illustrates the structure of the proposed MLO-GS, which groups clients based on the data distribution and updates the model parameters by setting different FL-participation frequencies for each group. Clients in the IID groups, where data are evenly distributed, participate in the FL in every round. Conversely, clients in non-IID groups, wherein the data distribution is uneven, participate periodically. MLO-GS employs an MLO, which is a standard technology in wireless local area networks, to allocate communication links to each group and enhance the efficiency of FL. MLO allows the simultaneous use of wireless links on different frequency bands and channels, thus improving the network efficiency by distributing the traffic load when clients communicate with each other or with the server.

3 Results and Discussion

This study evaluates the accuracy, communication cost, and average privacy-leakage

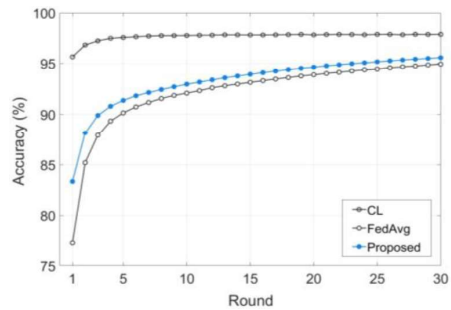


Fig. 2. Accuracy comparisons of CL, FedAvg, and Proposed.

of MLO-GS method and compares them with those of the conventional CL and FedAvg [4] methods on the MNIST dataset. We used a total of 100 clients, with 50 IID and 50 non-IID data. The non-IID data were set as those with an unbalanced class distribution, and the data classes were randomly allocated based on the Dirichlet distribution before being assigned to non-IID clients. Fig. 2 shows the accuracies of the conventional and proposed methods over several learning rounds. CL exhibits the highest accuracy and is the fastest because it aggregates all the data on the server for learning, whereas FedAvg and DFL show slower convergence speeds. However, as MLO-GS intermittently involves a non-IID group in the FL, it converges faster than FedAvg. MLO-GS converged approximately five times faster on average to achieve a 94% accuracy, thereby improving the initial convergence speed by approximately 25%. The communication costs of the conventional and proposed methods are shown in Fig. 3 (a). The communication costs of FedAvg and MLO-GS were calculated by summing the weights of the local models sent to the server and those of the global model sent by the server to the clients in each round. The CL method measures the data transmitted to the server for model training. The CL method incurred the highest communication cost because it transmits the raw data to the server, followed by FedAvg and then MLO-GS, which had the lowest communication cost. MLO-GS reduces communication costs by reducing the participation frequency of the non-IID group in the FL, thus mitigating the data imbalance problem caused by non-IID data. Therefore, it reduced the communication cost by approximately 87.87 and 44.88% compared with CL and FedAvg, respectively.

Fig. 3 (b) illustrates the average privacy-leakage of the conventional and proposed methods. Privacy-leakage is assessed by evaluating the potential of sensitive-information leakage by measuring the variation in model updates and amount of data transmission. Larger data transmissions enhance the probability of sensitive information being included, thereby increasing the risk of information leakage. As CL transmits raw data, it has the highest risk of data leakage. By contrast, FedAvg and MLO-GS mitigate this risk by transmitting only the model parameters from the distributed clients to the server. However, MLO-GS further reduces the average risk of privacy leakage by adjusting the participation frequency of the clients, thereby minimizing the number of communications. Consequently, MLO-GS decreases the average privacy-leakage rate by approximately 43.93 and 87.87% compared with FedAvg and CL, respectively.

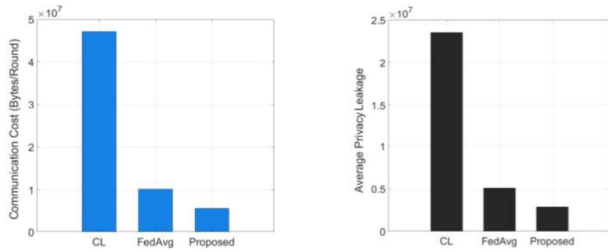


Fig. 3. Comparison of (a) communication costs and (b) average privacy leakage.

4 Conclusion

The recent advancements in artificial-intelligence technologies have focused attention on their applications in the digital healthcare industry. To provide digital healthcare services while protecting data privacy, it is essential to employ an FL method that learns without sharing data with servers. However, FL deteriorates the performance of global models and increases the communication costs for non-IID data. As wearable devices employed for digital healthcare are wireless and lightweight, it is important to reduce the communication costs. Therefore, this paper proposed MLO-GS method that strengthens privacy and reduces communication costs in an FL environment comprising non-IID device data. The evaluation results showed that MLO-GS improved the initial convergence speed by 25% on average compared with the conventional FedAvg method, along with enhancing privacy and reducing communication costs by approximately 45%. Moreover, it reduced communication costs by approximately 88% and enhanced privacy by approximately 87.87% compared with the CL method. Future studies should focus on dynamically setting the device-grouping criteria based on user requirements and optimizing communication cost, training time, and energy consumption.

Acknowledgments. This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE) under the Training Industrial Security Specialist for High-Tech Industry (grant number RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT) and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310), supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Na-Eun Park, Seo-Yi Kim, Il-Gu Lee.: Selfish attack detection and response using cooperative backoff adjustment in wireless sensor networks, ICT Express, 2024, https://doi.org/10.1016/j.icte.2024.05.014.

2. Stoumpos, A.I., Kitsios, F., Talias, M.A.: Digital Transformation in Healthcare: Technology Acceptance and Its Applications. International Journal of Environmental Research and Public Health, 20 (2023), https://doi.org/10.3390/ijerph20043407.
3. Wu, D., Zou, X., Zhang, S., Jin, H., Xia, W., Fang, B.: SmartIdx: reducing communication cost in federated learning by exploiting the CNNs structures. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4254–4262. AAAI (2022). https://doi.org/10.1609/aaai.v36i4.20345.
4. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP (54), 1273-1282 (2017), https://doi.org/10.48550/arXiv.1602.05629.

COLA Sketch: Enhancing Intrusion Detection with Collision-Aware Noise Reduction

JiYeon Kim, Sian Kim, and DaeHun Nyang

Department of Cyber Security, Ewha Womans University, Seoul, Republic of Korea
{wuwusm0527,ksy60a,nyang}@ewha.ac.kr

Abstract. Network traffic measurement is crucial for enhancing security by identifying and mitigating potential threats. Sketch-based measurement methods are efficient, using minimal resources while providing accurate analysis. However, they can overestimate traffic, leading to false alarms and mistakenly identifying legitimate users as attackers. To address this, we introduce Collision-Aware sketch (COLA), which includes a collision detection mechanism to reduce errors caused by collisions. Our scheme outperforms Count-Min sketch with Conservative Update (CMS-CU) in security applications, effectively reducing noise and false positives, thus offering a more reliable experience for benign users.

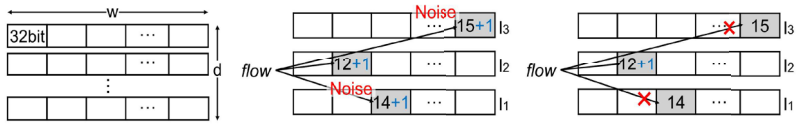
Keywords: Network Monitoring · Sketch Method · Intrusion Detection.

1 Introduction

Network traffic measurement is vital for security applications like intrusion detection system (IDS), as it helps identify and mitigate cyber threats [10]. In this context, methods like hardware-based [1], sampling-based [6], and sketch-based approaches are used to detect critical flows indicating potential malicious activity. Among them, sketch-based approaches [3, 4, 9, 14] are favored for their low memory and computational complexity, and ability to accommodate all flows.

Count-Min sketch [3] is widely used for its straightforward structure and operations. As shown in Fig.1 (a), CMS has d layers, each with w 32-bit counters. When a packet arrives, each layer uses a different hash function to determine the counter location for the flow, and the counter value is incremented, as shown in Fig.1 (b). The decoded value is the minimum counter value across all layers, minimizing error. Therefore, values from other layers are not used, thus updating those counters only adds noise to other flows that collide at those counters.

CMS with Conservative Updates [7] (CMS-CU) addresses this noise by updating only the smallest counter, as shown in Fig. 1 (c), avoiding unnecessary updates. However, CMS-CU still cannot identify noise from collisions at the minimum value counter, risking overestimation and misidentifying benign behavior as attacks. To this end, we propose Collision-Aware sketch (COLA), which improves CMS-CU by incorporating collision-aware detection mechanism. Unlike CMS-CU, COLA skips updates when collisions are detected, reducing overestimation. Without additional memory use, COLA reduces the probability of false alarms, enhancing intrusion detection system by improving the user experience.



(a) Data Structure (b) CMS Encoding (c) CMS-CU Encoding

Fig. 1: Data structure and workflow of CMS and CMS-CU.

2 Design of COLA Sketch

2.1 Data Structure

As shown in Fig. 2 (a), COLA uses a two-dimensional array with depth d and width w. Each 32-bit counter is split into two parts. The 24-bit part, called Count Sketch (CS), operates like CMS-CU and counts flow size. The 8-bit part, called Collision Detect Sketch (CDS), is responsible for collision detection. COLA uses $2d$ hash functions in total, applying d hash functions for both CS and CDS.

2.2 Encoding and Decoding

In COLA, after updating CS, the minimum value of CS is stored in the CDS. Later, if the minimum value of CS exceeds the minimum value of CDS, it indicates that the update was caused by a different flow. In such a case, COLA confirms a collision and prevents updates to avoid adding any noise. The detailed workflow of COLA is as follows.

1. When a flow arrives, it checks both CS and CDS to find the smallest value.

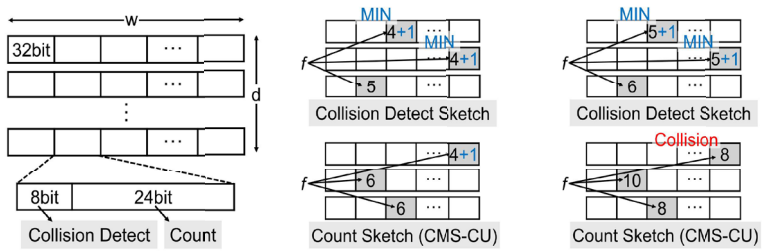
2-a. **When only CDS is updated:** If the minimum value of CDS is smaller than that of CS, it indicates a collision in CS, so no further action is taken in CS. However, CDS must record that one more packet has arrived for the corresponding flow, so the minimum counter in CDS is incremented by 1.

2-b. **When only CS is updated:** If the minimum value of CDS is greater than that of CS, it updates CS as in CMS-CU. No action is taken for CDS.

2-c. **When both are updated:** If the minimum value of CDS is equal to that of CS, COLA determines that there is no collision. Thus, it updates CS and CDS as in CMS-CU.

3. When decoding, return the minimum value of CDS if it is less than that of CS and CDS is not saturated; otherwise, return the minimum value of CS.

Fig. 2 (b) shows the case 2-c when no collision is detected. When the minimum value of CDS is equal to the minimum value of CS, it indicates that no noise was added after flow f updated CS. Therefore, both CS and CDS are updated. Fig. 2 (c) shows the case 2-a where flow f updated CS, but other flows have added noise to CS afterwards. Thus, instead of updating the CS, only the minimum values in CDS are incremented to store the current flow size of f. As shown in the examples, COLA effectively prevents unnecessary updates in CS by detecting collisions with CDS. This reduces overestimation errors, resulting in fewer false positives thus enhancing the reliability of intrusion detection systems.



(a) Data Structure (b) Collision Not Detected (c) Collision Detected

Fig. 2: Data structure and workflow of COLA.

2.3 Counting up to 32-bit

Using 24-bit counters presents the issue of not being able to estimate the flow size up to 32 bits. We can address this by using a single 1-bit mode bit. During encoding, if any of the 24-bit CS counters overflow, we flush CDS counters and set the mode bit to 1, allowing CS counters to utilize all 32 bits. Despite the overhead of flushing all CDS counters, most of them are already saturated and cannot perform collision detection in such cases. Thus, changing the mode provides an opportunity for COLA to utilize this idle space.

3 Evaluation

3.1 Experimental Setup and Evaluation Metrics

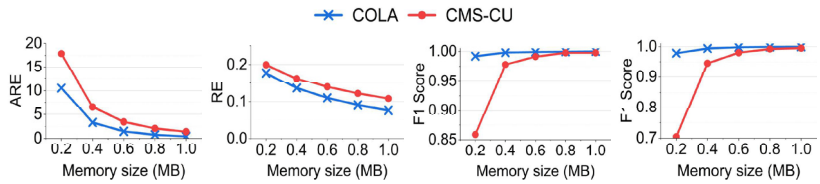
We compare CMS-CU and COLA with $d = 3$, using benign CAIDA [2] trace and malicious CIC DDoS [13] trace. In one epoch of the experiment, the benign trace contains about 2.57 M packets with 244 K flows, while the malicious trace has 3.30 M packets with 839 K flows. We evaluated the performance using three metrics: Relative Error (RE), Average Relative Error (ARE), and F1 score.

3.2 Security Applications

Flow Size Estimation. Flow size estimation is the most basic and essential measurement for analyzing each flow and understanding its characteristics. As shown in Fig. 3 (a) and Fig. 4 (a), COLA shows better performance than CMS-CU by achieving a lower ARE in all cases.

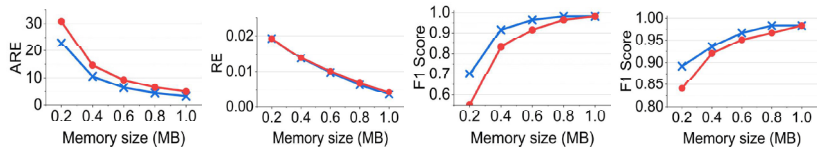
Entropy. Entropy is particularly significant from the perspective of DDoS detection, and many researches [8, 5, 11] have proposed intrusion detection systems utilizing entropy. We estimated the entropy of the flow size distribution for CMS-CU and COLA. As shown in Fig. 3 (b), COLA outperforms CMS-CU by achieving a lower RE in all traces and memory settings. On the other hand, CMS-CU is comparable to COLA in malicious dataset, as shown in Fig. 4 (b).

Heavy Hitter. Flows with sizes greater than a given threshold are referred to as heavy hitters. Heavy hitters can be considered anomalous due to its excessively high volume, and it is particularly important in the context of DoS attack detection [15, 12]. We set the threshold to 0.005% of the total number



(a) Flow Size (b) Entropy (c) Heavy Hitter (d) Heavy Changer

Fig. 3: Security applications with benign trace comparing CMS-CU and COLA



(a) Flow Size (b) Entropy (c) Heavy Hitter (d) Heavy Changer

Fig. 4: Security applications with malicious trace comparing CMS-CU and COLA

of packets. Fig. 3 (c) and Fig. 4 (c) illustrates that COLA outperforms CMS-CU by achieving a higher F1 score in most cases. Since the two schemes only result in overestimation, recall is always 1, making the F1 score dependent on precision. Therefore, the superiority of COLA in terms of F1 score indicates its ability to reduce false alarms. As the memory size increases, COLA and CMS-CU tend to exhibit similar performance. This is because, with larger memory space, collisions occur less frequently, reducing the amount of noise that COLA can eliminate.

Heavy Changer. Similar to heavy hitters, heavy changers, which exhibit significant variations in flow between two epochs, can also be considered anomalous behavior. We set heavy changer threshold as 0.005% of the total size changes. As shown in Fig. 3 (d) and Fig. 4 (d), COLA outperforms CMS-CU by achieving a higher in most cases, reducing noise and false alarms. With larger memory, CMS-CU is comparable to COLA since fewer collisions diminish the impact of COLA collision detection.

4 Conclusion

Sketch algorithms can enhance intrusion detection systems by accurately identifying the malicious flows such as heavy hitters and heavy changers. CMS-CU, an advanced version of CMS, aims to reduce noise with a minimum of one update but cannot eliminate probable noise from these updates. To address this limitation, we propose COLA, which introduces a collision-aware mechanism into CMS-CU. Without any additional memory usage, COLA outperforms CMS-CU, demonstrating the effectiveness of collision-aware strategies. By reducing collisions, COLA mitigates overestimation issue, which in turn improves overall measurement quality and decreases false positives in attack detection. This improvement leads to a more reliable intrusion detection system, providing a smoother and more comfortable experience for legitimate users.

References

1. Netflow. http://www.cisco.com/c/en/us/products/ios-nx-ossoftware/ios-netflow/index.html, accessed: 2024-07-26
2. The cooperative association for internet data analysis, equinix chicago data center (2018), https://www.caida.org, [13:00-14:00, Apr 19 2018., from Sao Paulo to New York]
3. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms **55**(1), 58–75 (2005)
4. Dao, D.N., Jang, R., Jung, C., Mohaisen, D., Nyang, D.: Minimizing noise in hyperloglog-based spread estimation of multiple flows. In: 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 331–342. IEEE (2022)
5. David, J., Thomas, C.: Ddos attack detection using fast entropy approach on flow-based network traffic. Procedia Computer Science **50**, 30–36 (2015)
6. Duffield, N., Lund, C., Thorup, M.: Learn more, sample less: control of volume and variance in network measurement. IEEE Transactions on Information Theory **51**(5), 1756–1775 (2005)
7. Estan, C., Varghese, G.: New directions in traffic measurement and accounting. In: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 323–336 (2002)
8. Jun, J.H., Ahn, C.W., Kim, S.H.: Ddos attack detection by using packet sampling and flow features. In: Proceedings of the 29th annual ACM symposium on applied computing. pp. 711–712 (2014)
9. Kim, S., Jung, C., Jang, R., Mohaisen, D., Nyang, D.H.: A robust counting sketch for data plane intrusion detection. In: 30th Annual Network and Distributed System Security Symposium, NDSS 2023. The Internet Society (2023)
10. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. ACM SIGCOMM computer communication review **34**(4), 219–230 (2004)
11. Lapolli, Â.C., Marques, J.A., Gaspary, L.P.: Offloading real-time ddos attack detection to programmable data planes. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). pp. 19–27. IEEE (2019)
12. Liu, Z., Namkung, H., Nikolaidis, G., Lee, J., Kim, C., Jin, X., Braverman, V., Yu, M., Sekar, V.: Jaqen: A {High-Performance}{Switch-Native} approach for detecting and mitigating volumetric {DDoS} attacks with programmable switches. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3829–3846 (2021)
13. Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST). pp. 1–8. IEEE (2019)
14. Yang, T., Jiang, J., Liu, P., Huang, Q., Gong, J., Zhou, Y., Miao, R., Li, X., Uhlig, S.: Elastic sketch: Adaptive and fast network-wide measurements. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. pp. 561–575 (2018)
15. Zhang, M., Li, G., Wang, S., Liu, C., Chen, A., Hu, H., Gu, G., Li, Q., Xu, M., Wu, J.: Poseidon: Mitigating volumetric ddos attacks with programmable switches. In: the 27th Network and Distributed System Security Symposium (NDSS 2020) (2020)

Clock Glitch-based Fault Injection Attack on Deep Neural Network

Hyoju Kang[1 [0009−0006−9810−7392]], Seongwoo Hong[1 [0009−0004−3173−6229]], and Jaecheol Ha[1 [0000−0003−3730−5580]]

[1] Hoseo University, Asan-si, Chungcheongnam-do, 31499, South Korea
gywn7597@gmail.com, hshsw5660@gmail.com, jcha@hoseo.edu

Abstract. The use of Deep Neural Networks (DNNs) is gradually increasing in various fields due to their high efficiency in data analysis and prediction. Despite the in-creasing use of DNNs, security threats are emerging. In particular, faults that oc-cur in forward propagation and activation function, which can directly impact the predictions of DNNs, can have catastrophic effects on the prediction accuracy. In this paper, we performed fault injection attacks on the forward propagation and activation functions in each layer of the MNIST digit classification DNN model. As a result of the fault injection attack on the hidden layer of the DNN, it was confirmed that all processes occurring in the forward propagation can be skipped. In addition, on the softmax function of the output layer, it was confirmed that the iterative operation was skipped, and classified as image 0. Therefore, there is a threat that the clock-based fault injection attack in the DNN model can cause mal-function and classify it as an unintended false image.

Keywords: Deep Neural Network · Fault Injection Attack · AI Security.

1 Introduction

Recently, the use of Deep Neural Network (DNN) has been increasing in various fields such as autonomous vehicles, industrial systems, and IoT. The DNN offer users significant benefits in complex data analysis and prediction. However, their importance in security is also attracting attention as they can cause serious problems in the presence of security vulnerabilities.

A DNN is generally a neural network consisting of an input layer, two or more hidden layers, and an output layer. The prediction of DNNs is determined based on several factors, including data, forward propagation and back propagation, and activation functions. Fault injection attack is one of several attacks that threaten the security of DNN. In this attack, an adversary intentionally injects faults to induce misclassification or malfunction of the DNN model [1]. Such faults injected into hardware devices can occur through clock glitches [2], voltage glitches [3], laser beams [4, 5] and so on.

This paper performed a clock glitch-based fault injection attack on an activation function of each hidden layer and softmax function of output layer in

the DNN model. The fault was injected to induce a skip operation on the loop statement used in forward propagation and softmax function.

2 Fault Injection Attack on DNN model

2.1 Digit Classification of MNIST Dataset

We adopt a simple DNN model trained on the MNIST handwritten digit image dataset as the target model for fault injection attacks. The input layer of this DNN receives 784 as an input, which is the flattened value of an image with a size of 28×28 pixels. There were two hidden layers, in which use ReLU as the activation function. The node of the hidden layer was set to be small at 16 and 14, respectively, due to the limitations of the SRAM(40KB) and Flash memory(256KB) sizes of the STM32F3 chip. Finally, softmax was used as the activation function for digit classification in the output layer.

2.2 Experimental Equipment

We used a PC, ChipWhisperer-Lite (CW-Lite), and CW308 (STM32F3 chip) to carry out clock glitch-based fault injection attacks. The PC and CW-Lite communicate via a USB connection. If the PC sends the glitch parameter to CW-Lite using Python, CW-Lite adjusts the trigger position for fault injection based on the received glitch parameter value.

The DNN code written in the C language is converted into assembly code using the ARM GCC 10.2.1 (none) compiler and then uploaded to CW308 (STM32F3 chip). In the case of STM32FX, it operates using an internal clock instead of an external clock. Therefore, a trigger command must be written in advance within the C language-based DNNs code. The following Fig 1 shows the interfaces and operational processes of the experimental equipment.

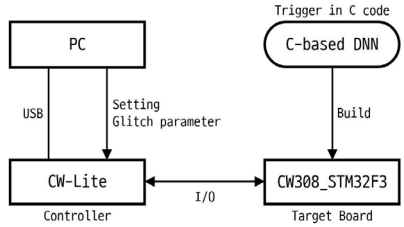


Fig. 1. Interfaces of experimental equipment for fault injection attack

3 Glitch-based Fault Injection Attack

3.1 Attack point in DNN Algorithm

The attack points were set differently depending on the attack goal. In the case of forward propagation, the goal is to skip all operations within the for loop, so

line 1 of Fig 2(a) was set as the attack point. And the attack purpose on the softmax function is to run the first for loop once and then skip the rest of the loop. To induce this result, the experiment was performed by setting the end of the first for loop as the attack point as shown in Fig 2(b).

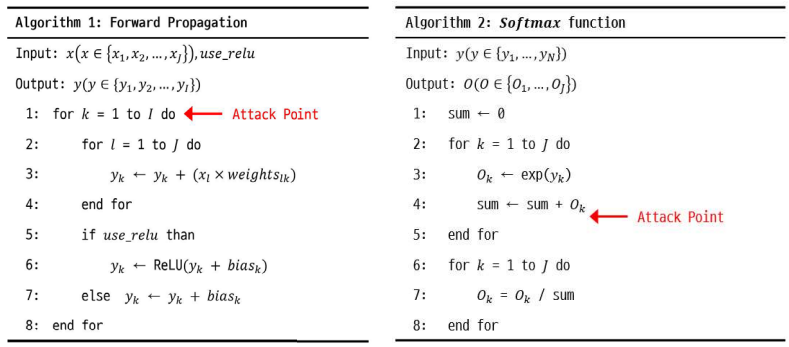


Fig. 2. Attack point in forward propagation or softmax algorithm

3.2 Glitch Parameters

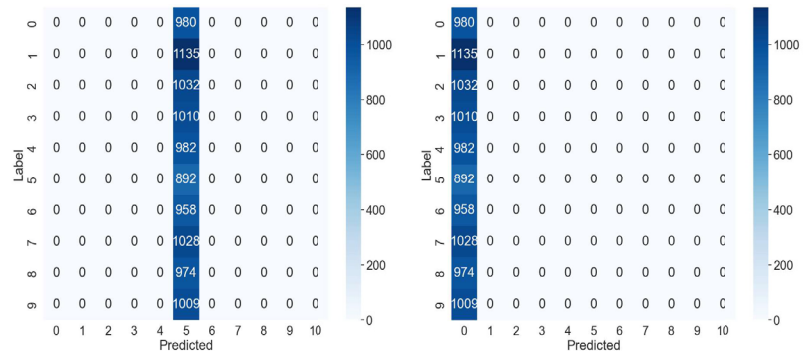
The glitch parameters that determine the success rate of clock glitch-based fault injection attacks are width, offset, ext_offset, and repeat. The repeat, which determines the number of repetitions of fault injection, was set to 1, so it affects exactly at the point of attack. The ext_offset to determine the location of the fault was set to 5 for forward propagation and 7 for softmax based on the ARM Cortex-M4 manual, which is the core of STM32F3. The width and offset, which determine the length of the fault, were set as the values with the highest attack success rate after attempting to inject the fault within the full range from -44 to 44.

3.3 Experiment Result

The results of the fault injection attack on the DNN model are shown in Fig 3. By injecting a fault into the first neuron operation in each layer, we can ensure that the final output is classified into a specific class. The following Fig 3(a) is the result of skipping of the forward propagation for the hidden layer 1 and the hidden layer 2. If forward propagation for hidden layer 1 is skipped, only the bias of hidden layer 2 feeds as input to the output layer. Therefore, no matter what image is entered, the same value is accepted as an input value of output layer. This is classified into a specific class 5, although it depends on the weights and biases of the output layer.

If forward propagation for hidden layer 2 is skipped, the outputs of hidden layer 2 will all be five. As a result, the digit image is also classified into the specific class with the largest bias value of the output layer.

Next, The Fig 3(b) is the result of skipping of softmax function of the output layer. If a fault is injected into the attack point in the algorithm of Fig 2(b), O_k and *sum* become the same value. As a result, it is finally classified as class 0 with probability 1. We demonstrate that injecting clock glitch-based faults into loop statements in a DNN for digit classification using the MNIST dataset can lead to misclassification into a specific class.



(a) Hidden layer1 and Hidden layer2 (b) Softmax

Fig. 3. Misclassification of fault injection attack against DNN Model

4 Conclusion

As the number of IoT devices utilizing DNNs has increased recently, many physical threats to them have also emerged. In particular, injecting faults into data, forward propagation process, and activation function that directly affect the prediction of DNNs can lead to misclassification or malfunction of DNN models.

In this paper, we attempted a clock glitch-based fault injection attack on the forward propagation process and softmax function, and showed a high attack success rate that reduced the model prediction accuracy to 0%. In most cases where the loop is skipped due to fault injection, it was confirmed that the node output value was fixed and, as a result, it was classified only into a specific class. In conclusion, naive implementation of DNNs on IoT devices can be a serious cybersecurity threat.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00396269, Development of security verification and vulnerability analysis system for IC chips through fault injection)

References

1. Breier, J., Hou, X., Jap, D., Ma, L., Bhasin, S., Liu, Y.: Practical fault attack on deep neural networks. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 2204–2206 (2018)
2. Fukuda, Y., Yoshida, K., Fujino, T.: Fault injection attacks utilizing waveform pattern matching against neural networks processing on microcontroller. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **105**(3), 300–310 (2022)
3. Jap, D., Won, Y.S., Bhasin, S.: Fault injection attacks on softmax function in deep neural networks. In: Proceedings of the 18th ACM International Conference on Computing Frontiers. pp. 238–240 (2021)
4. Liu, W., Chang, C.H., Zhang, F., Lou, X.: Imperceptible misclassification attack on deep learning accelerator by glitch injection. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). pp. 1–6. IEEE (2020)
5. Tajik, S., Ganji, F.: Artificial neural networks and fault injection attacks. In: Security and Artificial Intelligence: A Crossdisciplinary Approach, pp. 72–84. Springer (2022)

Modular Design Approach in API Security

I Wayan Adi Juliawan Pawana[1,2], YoungJae Kim[1], Bonam Kim[1], and Ilsun You[1]

[1] Department of Financial Information Security Kookmin University, Seoul, South Korea
{adijuliawan, zeroash1225, kimbona, isyou}@kookmin.ac.kr
[2] Department of Electrical Engineering Udayana University, Bali, Indonesia
adijuliawanpawana@unud.ac.id

Abstract. APIs are vital in modern applications, enhancing system interoperability and functionality. However, the widespread adoption of APIs also increases associated risk. The OWASP Top 10 API Security Risk—2023 list highlights critical vulnerabilities. Effective API security involves strong authentication, encryption, and monitoring practices. OAuth2 secures APIs by regulating roles and permissions, while Transport Layer Security (TLS) ensures encrypted communication, protecting data integrity. API Gateways are essential for managing and securing API traffic and enforcing security policies consistently. Combining API Gateway, JWT fingerprints validation, anomaly detection, and signature detection can protect the API server from malicious attacks and ensure the system remains robust, adaptable, and maintainable.

Keywords: API Security · API Vulnerabilities · OAuth2 · API Gateway · TLS · Anomaly Detection · Signature Detection.

1 Introduction

APIs have become increasingly vital in modern applications, facilitating seamless communication and integration between software systems. According to a survey by RapidAPI [1], over 70 % of developers plan to increase their use of APIs in 2023, highlighting the growing trend of API adoption. This trend is further supported by more than 75% of developers prioritizing participating in the API economy, indicating the significant role APIs play in enabling innovation and operational efficiency across various industries. However, this widespread adoption also increases associated risks.

Several notable API breaches have highlighted these vulnerabilities. For example, in 2018, the Facebook-Cambridge Analytica scandal involved the misuse of Facebook's API, allowing third-party apps to collect data on millions of users without their explicit consent [2]. Another instance is the T-Mobile breach in 2020, where hackers exploited an API to gain unauthorized access to customer information, including personal and billing data [3]. This situation underscores the Importance of implementing stringent security measures to mitigate the inherent risks associated with API usage.

2 API Vulnerabilities

The Open Web Application Security Project, or OWASP, is a globally recognized non-profit organization dedicated to web application security. One of its most well-known initiatives is the OWASP Top 10 API Security Risk - 2023 [4], a regularly updated list highlighting the most critical security vulnerabilities related to Application Programming Interfaces (APIs). Among these vulnerabilities, we concentrate on the most crucial ones in RESTful API.

- **API2:2023 - Broken Authentication**
  Broken Authentication refers to vulnerabilities in API authentication mechanisms that allow attackers to gain unauthorized access. This can occur due to weak passwords, credential stuffing, lack of multi-factor authentication (MFA), insecure storage, improper session management, exposed authentication endpoints, and poor token expiration and revocation practices
- **API4:2023 - Unrestricted Resource Consumption**
  Unrestricted Resource Consumption refers to vulnerabilities in APIs that do not impose adequate limits on resource usage, such as CPU, memory, bandwidth, or database connections. This lack of restrictions can lead to resource exhaustion, making the API susceptible to denial-of-service (DoS) attacks where an attacker overwhelms the system with excessive requests, causing performance degradation or outages
- **API6:2023 - Unrestricted Access to Sensitive Business Flows**
  Unrestricted Access to Sensitive Business Flows refers to vulnerabilities in APIs that allow unauthorized users to access or manipulate critical business processes. This can occur when APIs lack proper access controls, enabling attackers to perform actions like financial transactions, data manipulation, or other sensitive operations without appropriate authorization
- **API7:2023 - Server-Side Request Forgery**
  Server-Side Request Forgery (SSRF) refers to vulnerabilities in APIs where an attacker can manipulate the server to make unauthorized requests to internal or external systems. This occurs when an API accepts user input specifying a resource location and retrieves the resource without proper validation. SSRF can lead to unauthorized access to internal systems, data leakage, and other security breaches.
- **API10:2023 - Unsafe Consumption of APIs**
  Unsafe Consumption of APIs refers to the risks associated with using third-party APIs without proper security measures. If the consumed APIs have security flaws, lack proper authentication, or transmit data insecurely, this can lead to vulnerabilities. Attackers can exploit these weaknesses to compromise systems, steal data, or perform unauthorized actions.

3 API Security

The increasing prevalence of APIs in modern software development represents a significant advancement in enhancing system interoperability and functionality.

However, this growth also underscores the critical importance of implementing robust API security strategies to address potential security challenges. Effective API security measures encompass crucial components, including robust authentication and authorization protocols, secure data transmission through encryption, and monitoring and logging practices.

Access control management with OAuth2, as outlined in RFC 6749 [5], is a critical framework for securing APIs by defining and regulating roles, permissions, and authorization mechanisms. OAuth2 facilitates third-party applications' ability to access user resources without exposing their credentials directly. The core mechanism involves an authorization server that issues access tokens after authenticating the user. These tokens, which encapsulate the user's permissions, are then used by client applications to access protected resources on a resource server, ensuring that access is secure and appropriately restricted.

Transport Layer Security (TLS), defined in RFC 8446 [6], is a cryptographic protocol for securing communication over computer networks, including APIs. In API security, TLS ensures that data transmitted between clients and servers is encrypted, safeguarding it from eavesdropping and tampering. TLS maintains its confidentiality and integrity by encrypting data and protecting sensitive information such as personal data, authentication tokens, and financial transactions

An API Gateway is a crucial component in API security. It acts as an intermediary between clients and backend services, managing, routing, and securing API traffic, providing a single entry point for all client requests. The API Gateway is responsible for various functions such as authentication, authorization, rate limiting, logging, and monitoring. Centralizing these functions simplifies the management of API security policies and ensures consistent enforcement across all API endpoints.

4 Modular API Protection

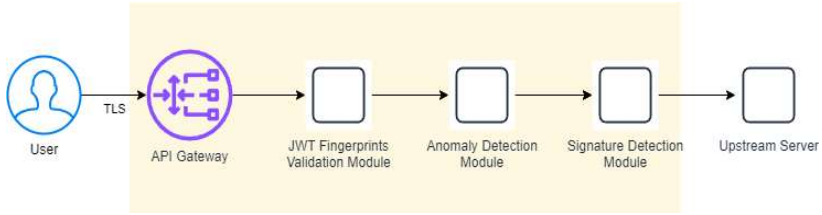


Fig. 1. Modular API Protection Diagram.

As shown in Fig. 1, We design modular API protection by combining API Gateway, JWT fingerprints validation, anomaly detection, and signature detection. An API Gateway can effectively address vulnerabilities related to broken

authentication and unrestricted resource consumption. For broken authentication, the gateway can enforce strong authentication and authorization policies, manage secure token storage, and implement multi-factor authentication. For unrestricted resource consumption, the gateway can enforce rate limiting, set execution timeouts, and allocate resource quotas to prevent DoS attacks.

JWT (JSON Web Token) fingerprinting can address Unrestricted Access to Sensitive Business Flows vulnerability by enhancing the security of token-based authentication. JWTs can be augmented with a unique fingerprint that ties the token to specific user attributes or session characteristics, such as IP address, device ID, or browser fingerprint. This ensures that tokens cannot be reused or misused by unauthorized entities. If a JWT is stolen, it cannot be used in a context different from the original intended. This additional layer of security helps enforce stricter access controls.

Anomaly and signature-based detection methods can significantly enhance API security against SSRF and unsafe API consumption. Anomaly-based detection monitors traffic for deviations from established patterns, flagging unusual behaviors such as unexpected outbound requests indicative of SSRF attacks. Signature-based detection uses predefined patterns of known threats to identify malicious API calls. Regular expressions (regex) are powerful tools in these methods, allowing for precisely matching patterns within API requests and responses.

To evaluate the performance of this modular design, the F1 score is used to evaluate anomaly and signature detection modules. CPU and memory usage is measured in each module within different scenarios based on the number of connections.

5 Conclusion

This article explains the security vulnerability in RESTful API based on the OWASP API Security Top 10 and discusses how to protect it from malicious attacks. By leveraging modular design, independent API protection modules can be developed and deployed separately to accomplish cloud native architecture. It also enhances scalability and maintainability, as API protection modules can be managed individually. Modular design ensures the system remains robust, adaptable, and easier to troubleshoot by decoupling components and focusing on specific functionalities.

6 Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (RS-2024-00437252,Development of anti-sniffing technology for mobile communication and AirGap environments)

References

1. State of APIs: growth and more growth on tap for 2023, https://rapidapi.com/blog/state-of-apis-growth-and-more-growth-on-tap-for-2023/, last accessed 2024/07/25
2. History of the Cambridge Analytica Controversy, https://bipartisanpolicy.org/blog/cambridge-analytica-controversy/, last accessed 2024/07/25
3. T-Mobile suffers 8th data breach in less than 5 years, https://www.csoonline.com/article/574385/t-mobile-suffers-8th-data-breach-in-less-than-5-years.html, last accessed 2024/07/25
4. OWASP Top 10 API Security Risks – 2023, https://owasp.org/API-Security/editions/2023/en/0x11-t10/, last accessed 2024/07/25
5. The OAuth 2.0 Authorization Framework, https://datatracker.ietf.org/doc/html/rfc6749, last accessed 2024/07/23
6. The Transport Layer Security (TLS) Protocol Version 1.3, https://datatracker.ietf.org/doc/html/rfc8446, last accessed 2024/07/23

Poster: Impact Assessment of Cooperative Driving Performance because of I/O Latency

Juon Kim[1][0009−0008−9832−3057], Sanghwa Jin[2], Jongin Lee[2], and Wonsuk Choi[1]

[1] Korea University, Seoul, South Korea
[2] Samsung Electronics, South Korea

Abstract. Many automotive manufacturers are planning to produce their vehicles with E/E architectures, such as centralized and zonal architectures. Accordingly, there is an increasing demand for the development of automobile data storage systems which store and analyze data generated in vehicles. In addition, this automobile data storage systems are being designed as a shared storage system that accommodate multi-tenancy. It is expected that an autonomous vehicle would generate and store a large number of data. For the negotiation of cooperative autonomous driving, V2X messages are transmitted and these messages are created with stored data. In this poster, we analyze impact assessment of cooperative driving performance as a function of I/O latency. Many negotiation scenarios demand being completed within approximately from 10 to 100ms; accordingly, I/O latency should be minimized. It is imperative to investigate how traffic conditions influence latency in V2X communication contexts and the consequent ramifications on data storage.

Keywords: E/E Architecture · Shared Storage System · I/O Latency.

1 Introduction

Currently, many automotive manufacturers are planning to produce vehicles with Electric/Electronic (E/E) Architecture, which refers to the framework of a vehicle's electrical and electronic systems. Modern vehicles are equipped with numerous electronic control units (ECUs). Initially, a distributed E/E architecture was predominantly employed, wherein each ECU operated with its own hardware and software and communicated independently with other ECUs. However, there is an emerging trend towards a domain-centralized architecture, where ECUs are organized by their functions into domains, and a domain control unit (DCU) governs these domains. The advantages of centralized E/E architecture include its powerful computing capability, enabling the implementation of complex functions and efficient processing. Among automotive manufacturers, Tesla is currently mass-producing vehicles with a centralized E/E architecture and was the first to introduce automotive OTA technology. Autonomous driving technology has evolved into a platform-based system, requiring significant computing power to operate. Consequently, this led to the development of the centralized E/E architecture.

For this reason, it is expected that an autonomous vehicle would generate and store a large amount of data. For the negotiation of cooperative autonomous driving, V2X messages are transmitted, and these messages are created with stored data. In this poster, we analyze the impact of I/O latency on cooperative driving performance. Many negotiation scenarios demand completion within approximately 10 to 100 ms; accordingly, I/O latency should be minimized. It is imperative to investigate how traffic conditions influence latency in V2X communication contexts and the consequent ramifications on data storage.

2 Cooperative Driving for V2X Communication

The Intelligent Transportation Systems (ITS) Standards were published to prevent accidents and improve road safety by sharing messages in real-time through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. In real-time cooperative driving, low latency I/O is essential since messages need to be processed instantaneously upon storage. Based on SAE J2735 and SAE J3216, we design a test scenario for autonomous driving negotiation in vehicle-to-everything (V2X) communication to assess the cooperative driving performance as a function of I/O latency.

2.1 Standards for ITS

SAE J2735 standard defines the standard message format required for V2X communication to improve transportation safety and efficiency[1]. Currently, 30 messages are defined and continuously expanded to support various application services. For example, BSM (Basic Safety Message) transmits status information related to vehicle safety and EVA (Emergency Vehicle Alert) sends warning messages from emergency vehicles. **SAE J3216** standard defines the type of information exchanged between vehicles required to enable collaborative automation [3]. This standard focuses on Cooperative Driving Automation (CDA) by classifying four cooperation classes, as shown in Table 1.

Table 1. Four classes as a function of degree of cooperation in SAE J3216 Standard

Classes	Cooperation Types	Cooperation Information	Negotiation
Class A	Status sharing	Perception information	No
Class B	Intent sharing	Driving plan	No
Class C	Agreement seeking	Negotiation message	Yes
Class D	Prescriptive	Emergency information	Yes

2.2 Test Scenario

In cooperative autonomous driving, automated vehicles exchange messages for road safety and traffic management. In our test scenario, various types of vehicles—such as emergency vehicles, and normal vehicles (buses, trucks, taxis, and passenger cars)—exchange messages to ensure emergency vehicles can navigate more efficiently and safely. A normal vehicle transmits a BSM containing vehicle status information such as location, speed, acceleration, and timestamp to nearby vehicles. An emergency vehicle sends an EVA message containing emergency vehicle status information with warning to nearby vehicles. Whenever a normal vehicle receives messages from a nearby vehicle, it first checks if the vehicle is an emergency one. If so, it confirms the location information to determine if both vehicles are on the same road segment and subsequently adjusts its speed accordingly. This scenario corresponds to classes C and D, which determine the driving route through negotiations between vehicles in the SAE J3216 Standard.

3 Impact Assessment

As vehicle density and message transmission delay increase, the number of messages generated and stored by autonomous vehicles also increases, resulting in higher I/O latency. Therefore, we assess the impact of I/O latency on cooperative driving performance, specifically focusing on the average speed of vehicles, as a function of density and delay.

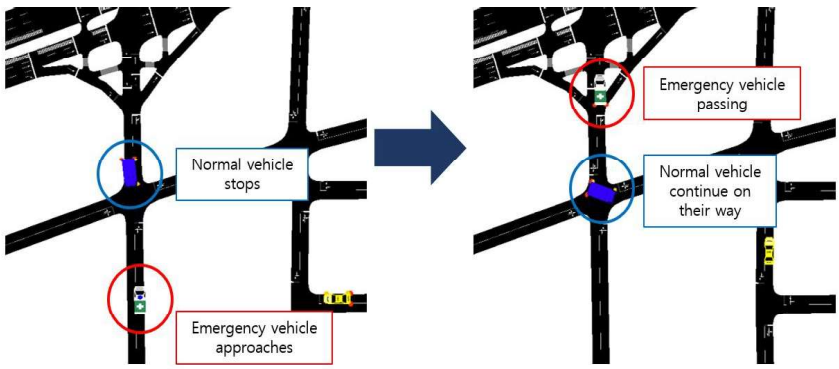


Fig. 1. Test Scenario

3.1 Experimental Setup

We use ms-van3t (Multi-Stack VANET framework for ns-3)[2], a new V2X tool based on Network Simulator (NS-3) and Simulation of Urban Mobility (SUMO), to assess the impact of I/O latency with above test scenario. SUMO is an open-source simulator capable of modeling autonomous driving, V2X communication, and traffic conditions. We extract a map around Korea University with OpenStreetMap (OSM) and then create a road network file using netconvert command. To set the route for vehicles, randomTrips tools are utilized.

Parameter. In our experiment, we set the parameters as follows: a transmission power of 30 dBm, a physical data rate of 12 Mbits/s, and a simulation duration of 100 seconds. Since the program ended before the simulation time when the density was 20 vehicles/km, we started our experiments with a density of 30 vehicles/km. Subsequently, since a collision occurred when the delay was 40 seconds, we limited the delay to a maximum of 30 seconds. Table 2 is vehicle type used in our simulation.

Table 2. Type of vehicles

	Emergency vehicle	Bus	Truck	Taxi	Passenger vehicle
Acceleration ability (m/s^2)	26.6	21.2	21.3	22.6	22.6
Deceleration ability (m/s^2)	24.5	24	24	24.5	24.5
Maximal deceleration (m/s^2)	30	27	27	29	29
Minimum gap (m)	1	1	2	1	1
Maximum velocity (m/s)	75.55	43.61	56.11	67.22	60
Probability	0.05	0.25	0.25	0.25	0.2

3.2 Experimental Results

Fig. 2 (left) shows the average latency as a function of delay and density. It is evident that as density increases, average latency also rises. This is due to the higher number of vehicles within the area, which leads to a greater volume of message exchanges. Fig. 2 (right) shows the average speed of vehicles as a function of delay and density. Here, an increase in density generally results in a decrease in average vehicle speed due to road congestion. Additionally, as

delay increases, the average vehicle speed tends to decrease, since delays in message exchanges between vehicles cause traffic control problems.

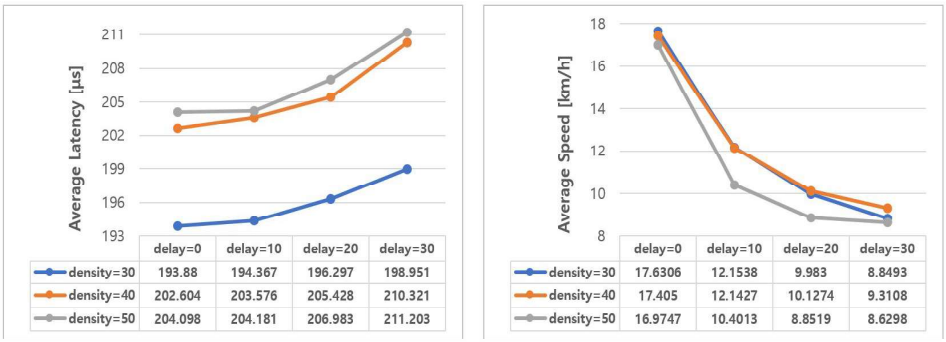


Fig. 2. Average Latency(left) and Average Vehicle Speed(right) as a function of delay and density

We confirmed how latency increases depending on the traffic environment in V2X communication and affects data storage. As density and delay increase, latency increases and, as a result, the average speed of vehicles decreases. In other words, as I/O latency increases, cooperative driving performance decreases.

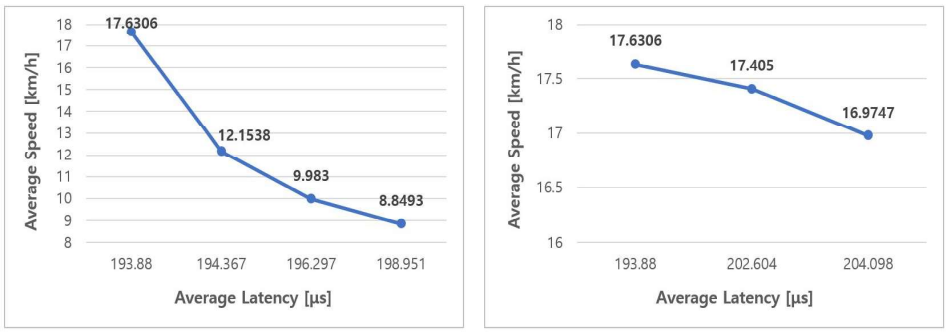


Fig. 3. Average Vehicle Speed as a function of Average Latency when density=30(left), delay=0(right)

4 Conclusion

Nowadays, automotive manufacturers are adopting centralized E/E architecture, which provides enhanced computing power and efficient processing capabilities. Therefore, autonomous driving demands a significant amount of data to be generated and stored, resulting in high I/O latency. In this poster, we assess how traffic conditions affect latency in V2X communication and their impact on data storage.

Acknowledgments. This work was supported by Memory Division of Samsung Electronics in 2023.

References

1. Kenney, J.B.: Dedicated short-range communications (dsrc) standards in the united states. Proceedings of the IEEE **99**(7), 1162–1182 (2011)
2. Raviglione, F., Carletti, C.R., Malinverno, M., Casetti, C., Chiasserini, C.F.: ms-van3t: An integrated multi-stack framework for virtual validation of v2x communication and services. Computer Communications **217**, 70–86 (2024)
3. Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., Ma, J.: Opencda: an open cooperative driving automation framework integrated with co-simulation. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 1155–1162. IEEE (2021)

Proposal of eqSIM Security Toolkit for Private 5G Device Authentication

Hyejin Yoon[1][0009-0003-1888-9303], Yoonjin Jang[1][0009-0005-0837-9275], Seunghwan Yun[2][0009-0008-2903-3574] and Yi Okyeon[1][0000-0001-8156-2360]

[1] Department of Financial Information Security, Kookmin University
{dbswls2265, razorback9413, oyyi}@kookmin.ac.kr
[2] Kookmin University
schneeleopard@ kookmin.ac.kr

Abstract. Recently, Private 5G has been built in various fields at home and abroad. In addition, attempts to build Private 5G in public institutions and defense are continuing in Korea.
As private 5G is introduced in various fields, connected devices also become diverse. In each field, in order to build a Private 5G safely, identification and authentication of connected devices must be performed. In this paper, we propose an eqSIM security toolkit as a device to identify, authenticate, and match keys to connected devices when constructing a 5G specialized network, and explain the functions and services of the components of the Security Tool.

Keywords: Private 5G, eqSIM, Authentication.

1 Intorduction

Just as the Private 5G market is expanding in Europe, the United States, China, and Japan, the market for Private 5G in Korea is also expanding. Overseas, Private 5G is being built in industries such as manufacturing, logistics, smart cities, and power. According to the status of Private 5G by service sector surveyed by Analytics Mason, the manufacturing sector accounts for the largest proportion at 24%. [1]

Table 1. Status of Private 5G by service

Service	Ratio	Detailed Field	Use Cases
Manufacturing	24%	Automation, Industrial	Automatic Transport Vehicles, Machinery
Traffic	23%	Airport, port, rail	AGV, asset tracking, voice/video communications
coal, oil, gas	16%	-	Mechanical, Voice/Video Communication
Public	15%	Education, government, smart city	drones, security cameras, voice/video communications
Utilities	10%	-	Smart Grid

In the same way as in the overseas construction field, it is being built in industrial fields and companies such as factories related to the manufacturing industry in Korea. In addition, attempts to build Private 5G in public institutions and military are continuing in Korea.
With the introduction of Private 5G in various fields, devices connected in Private 5G also become diverse. In each field, the understanding of building Private 5G securely should be identified and authenticated for connected devices. In addition, in order to build a Private 5G in the military and perform security functions, the KCMVP encryption module must be applied.
In this paper, we propose eqSIM, a device for identification, authentication, and key establishment of devices connected when building a Private 5G, and an eqSIM security toolkit for developing eqSIM within a Private 5G.

2 eqSIM Security Toolkit

The elements that constitute the eqSIM Security ToolKit are largely eqSIM, application services, algorithms, protocols, and test tools. The contents of each element are as follows.

2.1 eqSIM (embedded quantum entropy-based Subscriber Identity Module)

The eqSIM consists of a Quantum-based Cryptographic Module (QCM) equipped with a Quantum Entropy Chip (QEC) and an interface to connect with MCU/MPU, Private 5G devices, and Private 5G modems that handle services and protocols.
At this time, QCM, which is installed in eqSIM and plays a key role, is a hardware cryptographic module that aims for security level 2 of the KCMVP cryptographic module security requirements for use in the military or public institutions. In addition, modulation-evidence occurs when a physical attack occurs by protecting the circuit board with an opaque hard body to protect it from security level 2 physical attack. Through algorithms that perform block encryption, hash functions, and message authentication codes, QCM can encrypt and store authentication information of encryption keys and services.

2.2 Services

The application services provided through eqSIM for security of devices connected to Private 5G are as follows.

- Dentification
  The eqSIM containing the QCM stores the device's identification value inside the QCM that can store authentication information and provides a service for identifying the Private 5G devices with the corresponding value.
- Authentication
  The eqSIM provides an authentication service through a protocol that performs authentication with a device's identification value stored in the QCM.

- Key Establishment
  eqSIM performs key establishment with an authentication server outside the Private 5G network through a random number generator, key setting, and key derivation encryption service built into QCM.
- Authorization
  The eqSIM provides authority for Private 5G connections to Private 5G devices where identification, authentication, and key establishment have been performed normally.
- Security functons
  To provide each service, eqSIM provides encryption functions described in Table 2, such as block cipher, hash function, random number generator, and digital signature. eqSIM provides the algorithm described in Table 2 below according to each cryptographic service.

Table 2. Verification target cryptographic algorithm installed in QCM

Security Functions	Algorithm
Block Cipher	ARIA, LEA, AES ECB, CBC, CTR, GCM
Hash function	SHA2-256
Message Authentication	HMAC (SHA2-256), GMAC
Random Number Generator	Hash_DRBG (SHA2-256)
DigitalSignature	ECDSA (P-256/SHA2-256)
Key Establishment	ECDH (P-256)
Key Derivation	PBKDF (HMAC-SHA2-256

2.3 Protocols

The eqSIM operates protocols such as authentication and key matching to provide each service created in 2.2.

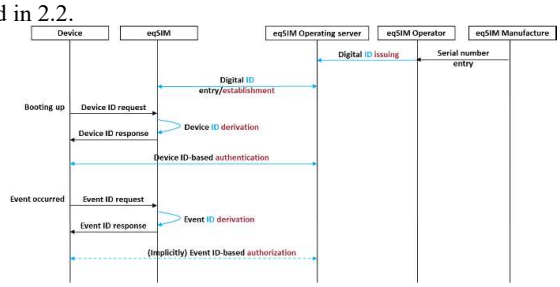


Fig. 1. Authentication Protocol using eqSIM

Figure 1 is a formalization of a protocol for identifying and authenticating private 5G devices using eqSIM. The eqSIM, which issued/injected serial numbers and device IDs

from eqSIM manufacturers and operators, is installed on the private 5G device and performs authentication with the eqSIM operation server at the moment the device is booted and whenever an event occurs.

2.4 Test Tools

The test tool in this paper is a tool for testing QCM in eqSIM to perform device's identification, authentication, and key matching in Private 5G, and four tools are proposed as shown in Figure 2.
From the left, the tools in the figure below are development tool for QEC, development tool for QCM, development tool for security module of QCM and key injector tool for QCM.

- Development Tool for QEC
  It is a tool for testing the entropy used in random number generation in QCM.
- Development Tool for QCM
  It is a tool for developing and testing cryptography modules equipped with quantum-based high-quality random numbers
- Development Tool for Security Modules of QCM
  It is a tool for developing and testing QCM
- Key Injector Tool for QCM
  It is a test tool for injecting keys into QCM.



Fig. 2. QCM Test Tools

3 Conclusion

This paper proposes eqSIM as a device for identifying, authenticating, and key matching connected devices when building Private 5G, and proposes an eqSIM security toolkit to perform eqSIM within eqSM and Private Private 5G.
In the Private 5G environment, Private 5G devices are identified through the eqSIM security toolkit, and authentication with the authentication server existing outside the Private 5G is performed. Private 5G devices on which authentication has been performed can be connected to Private 5G by receiving Private 5G access authorization to perform end-to-end data security.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2024-00398312).

References

1. Analysys Mason.: What are private LTE/5G network and why are they important?. (2021)

Proposal of a client consensus mechanism in a blockchain proof environment based on token distribution

Jinsu Kim[1][0000-0003-1009-3928], Keunbae Choi[2][0009-0002-4651-2290] and Namje Park[3][0000-0003-4434-8933]

[1] Department of Convergence Information Security, Graduate School, Jeju National University, Jeju-si 63294, Korea

[2] Dept. of Computer Education, Teachers College, Jeju National University, 61 Iljudong-ro, Jeju-si, Jeju Special Self-Governing Province, 63294, Korea

[3] Dept. of Mathematics Education, Teachers College, Jeju National University, 61 Iljudong-ro, Jeju-si, Jeju Special Self-Governing Province, 63294, Korea

namjepark@jejunu.ac.kr

Abstract. Blockchain applies various methods for proving block generation depending on the area of application. The commonly known consensus mechanisms are the proof of work method, which performs proof based on computing power, and the proof of stake method, which performs proof based on the stake held. In addition to these, various methods for consensus mechanisms are being proposed through research. Token distribution proof is a method that performs proof based on the contents of a token that is updated during the block generation process and transmitted to other nodes. Since it performs proof based on the generated token, it requires relatively low computing power and can allow all clients randomly selected on the block network to perform proof. However, the proof process in the consensus mechanism involves distributing a list of identification information created by the server to all nodes. Nodes holding tokens perform node selection based on the identification information list during the process of creating and distributing new tokens, thus having the disadvantage of centralization. Additionally, the existence of a pre-distributed identification information list suggests a limitation of restricted randomness. Therefore, in this paper, we propose a mechanism for client-centric random node selection. When updating tokens for transmission to new nodes is required, the client broadcasts its encrypted identification information until a certain number is met, updates the identification information list by including node identification information, and ultimately selects new token nodes from the identification information list that meets the specified number.

Keywords: Token distribution, blockchain, consensus mechanism, random selection

1 Introduction

As the network environment evolves, communication is being carried out on miniaturized devices, generating and exchanging various information. The advancement of IoT (Internet of Things) technology is applying communication technology in diverse areas beyond common electronic devices, such as vehicles and smart factories. In these environments, the specialized information generated is utilized for learning data or for enhancing user convenience. In the past, data generated was used independently on end devices or communicated within a limited scope, but now, it is increasingly being transmitted to external environments such as clouds or servers for further utilization.

The number of devices performing communication is exponentially increasing, necessitating ways to ensure the integrity of data generated by end devices. Blockchain can be a viable means to enhance data integrity on end devices. By sharing data, blockchain prevents tampering and forges a consensus mechanism to select entities that verify the generated data. Verified transactions form new blocks that include the hash of the previous block. Since blocks are shared among participants in the block network, they are resistant to tampering.

A token distribution-based consensus mechanism provides a method for proving transactions by distributing initially generated tokens, which include secret key pairs and token group keys, to random nodes on the block network. This mechanism requires less computing power because nodes performing the proof update the token's content and pass it to the next randomly selected node. However, it is challenging to prove if a client arbitrarily manipulates the node receiving the token since the process of token selection is entirely delegated to the client[1]. Additionally, follow-up studies aiming to move away from client-independent authority delegation have developed a centralized structure where a central server creates and sends identification information lists to each node. Nodes generating tokens select the next node based on these lists and obtain a new identification information list[2].

Therefore, this paper proposes a client-centric consensus mechanism that can perform node selection for token transfer through consensus among end devices without additional server intervention. In the proposed mechanism, nodes that update the token broadcast their identification information, including a threshold range, to the block network. Nodes receiving the identification information create a list containing their identification information and broadcast it to the block network, repeating the process until the threshold range is reached. The node with the list that first reaches the threshold range delivers the list to the token node, allowing for client-based identification information list generation to select the next node.

2 Client-Centric Random Node Selection Mechanism in Token Distribution-Based Blockchain Environment

In this paper, we propose a mechanism that creates an identification information list centered on clients to enhance the randomness of the node receiving the updated token in a token distribution-based blockchain environment. A token includes a symmetric

key for communication among nodes forming a group with the token, a public key pair to prove possession of the token, and information about the previous token node that generated the token. Therefore, a node that has verified a block updates the token by sharing the same symmetric key with the token group's nodes for use in consensus and generating a new public key pair.

To transmit the updated token, the node creates a dataset containing its identification information (ID_{TN}), the token hash (H_T), the creation time (T), the threshold range (R), and the data hash (H_D). This dataset is then encrypted with the node's private key and broadcast to the block network. The structure of the data is configured as shown in [Equation 1].

$$Data = [ID_{TN}, H_T, T, R, H_D] \quad (1)$$

Nodes that receive the broadcast data perform an initial verification by using the public key of the broadcasting node to ensure the data was generated by the node transmitting the token. Additionally, the data within the token is compared with the data hash to verify its integrity and check for tampering.

If the data is verified, the node creates a new dataset ($Data_R$) that includes its identification information (ID_R), the verification time (T_R), and the previously received data (Data). This new dataset also includes a new verification hash (H_R). The node then encrypts this dataset with its private key and performs another broadcast. The structure of the newly created dataset is shown in [Equation 2].

$$Data_R = [Data, ID_R, T_R, H_R] \quad (2)$$

Nodes that receive $Data_R$ will similarly create a dataset that includes their identification information, verification time, and verification hash. This process is repeated until the threshold range is reached.

3 Conclusion

In this paper, we propose a mechanism to select the next node in a token distribution-based blockchain environment without a dedicated server. This mechanism addresses the limitation where the selection process can become fixed due to being solely conducted by the node holding the token. The token-propagating node uses the identification information list collected through the broadcasted data to randomly select nodes. However, the proposed method may lead to an issue where malicious nodes could share the public key pairs of a node group exceeding the threshold range and maliciously create the identification information list. Future research is needed to resolve the problem of a single node holding the public key pairs of the node group and independently constructing the identification information list.

4

Acknowledement

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2022S1A5C2A04092269). And, this research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2023RIS-009) [Development of a Smart Tourism Content Service Platform Based on Artificial Intelligence through Digital Transformation of National Heritage Data]

References

1. Jinsu, k.,Namje, P.; Token Network-based Real-Time Transaction Execution Mechanism for Mobility Service. Journal of KIIT, 21(7), 95-103, Jul. 2023.
2. Jinsu, k.,Namje, P.; Enhanced Mechanism for Randomized Selection of Block Producers with Traceability in Token-based Blockchain Environments. Journal of KIIT, 22(5), 165-171, 2024.

Better RDP-GAN with A Noisy Generator

JungHye Choi and DaeHun Nyang

Department of Cyber Security, Ewha Womans University, Seoul, Republic of Korea
{Junghye.Choi, nyang}@ewha.ac.kr

Abstract. This paper proposes a novel approach to enhancing Generative Adversarial Networks (GANs) for synthetic data generation by directly applying a Differential Privacy (DP) mechanism into the Generator's training process. Unlike formal researches that apply DP to Discriminator, our method calculates the Generator's privacy cost directly, calculating the privacy cost more tightly. Additionally, we introduce VAE Decoder to convey target data information to Generator, mitigating mode collapse issues in GANs. VAE Decoder ensures DP preservation by adding noise based on its sensitivity, computed from clipped VAE model parameters.Our approach is expected to generate high-quality synthetic datasets while ensuring privacy guarantees.

Keywords: Differential Privacy · Generative AI · Synthetic Data· Generative Adversarial Networks · Rényi Differential Privacy.

1 Introduction

Recent research on Generative Adversarial Networks(GANs)[3] for synthetic data generation has improved to generate high-quality datasets by addressing model collapse and gradient vanishing issues[1, 10]. Training generative models requires original datasets similar to the target datasets, which often include sensitive information that cannot be shared publicly. This sensitive information restricts generated dataset's usage. To ensure the synthetic dataset's universal usability, we need to regulate the privacy leakage from the original dataset. Differential Privacy(DP)[2] offers a solution. Previous attempts typically apply DP to GANs Discriminator [6, 7, 9], training Discriminator with DP mechanism and updating Generator's latent variables. By post-processing proposition, Generator also preserves differential privacy.

In this paper, we propose applying a DP mechanism directly to the Generator's training process instead of the Discriminator. We can calculate the total privacy cost of the Generator directly, which is significantly smaller than the privacy cost composition of each Discriminator iteration. Also, we provide target data information to Generator through VAE Decoder to prevent mode collapse of GANs. VAE Decoder can also preserve DP by adding required noise of its own sensitivity. We calculate the VAE sensitivity based on the clipped range of VAE parameters. Our approach is expected to generate a high-quality synthetic dataset while preserving DP.

2 Design of RDP-GAN with a Noisy Generator

2.1 System Overview

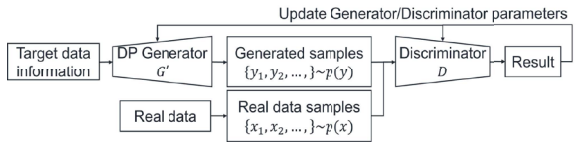


Fig. 1. Proposed Noisy-Generator GANs flowchart.

In our framework, as in typical GANs, Generator creates samples while Discriminator evaluates them. Discriminator uses the real dataset to training the difference between real data and synthetic data. In contrast to [7], which adds Gaussian noise to Discriminator's mean loss to achieve differential privacy, we apply the privacy mechanism directly to Generator. We add noise to Generator's loss value to ensure (α, ϵ)-RDP. Applying differential privacy directly to Generator can reduce the amount of noise required to maintain privacy.

However, Generator often converges slower than the Discriminator [5], which can lead to mode collapse. To address this problem, we provide the target data information to Generator's initial state via VAE Decoder [4]. This approach helps balance the training speed between Generator and Discriminator, improving the overall performance of RDP GANs.

2.2 Privacy Cost Analysis

Table 1. Notations used in the analysis

Notation	Description	Notation	Description
θ_{Enc}	VAE encoder parameter	n_e	VAE encoder network
θ_{Dec}	VAE decoder parameter	n_d	VAE decoder network
$\mathcal{L}_{reg}$	Regulation loss	$\mathcal{L}_{recon}$	Reconstruction loss
G'	Differential private generator	θ_g	Generator parameter
D	Discriminator	θ_d	Discriminator parameter
m_{vae}	VAE training batch size	m_{gan}	GANs training batch size
q_v	VAE sampling rate	ΔS_v	VAE sensitivity
q_g	GANs sampling rate	ΔS_g	GANs sensitivity
n_v	VAE training iteration number	n_g	generator iteration number
n_d	discriminator iteration number	ϵ_v	VAE privacy cost
ϵ_g	generator privacy cost	ϵ_d	discriminator privacy cost
ϵ_{total}	total privacy budget of GANs	$[-c_g, c_g]$	generator loss value clipping range
$[-c_v, c_v]$	VAE loss value clipping range	n	training dataset size

Sensitivity of GANs We consider neighboring dataset $\mathbf{z}, \mathbf{z}'$ with same size. Each loss value of data point is clipped $[-c_g, c_g]$. Therefore the max loss value difference for each data point is $2c_g$. The number of data point for 1 generation iteration is $m_{gan} \times n_g$, so the max difference each data can impact:

$$\Delta S_g = \max_{z_k \in \mathbf{z},\, z'_k \in \mathbf{z}} ||F_g(\mathbf{z}) - F_g(\mathbf{z}'))|| = \frac{2c_g}{m_{gan} \times n_d}. \tag{1}$$

Privacy cost for one iteration of Generator The privacy leakage of indirect access to the dataset is equal to or less than direct access. In GANs, Generator indirectly accesses real data through Discriminator. The proportion of data

accessed to update Generator affects the required noise. Discriminator accesses $\frac{m_{gan}}{n}$ proportion of data for one iteration, Generator accesses $q(=\frac{m_{gan}\times n_d}{n})$ for one iteration. According to Theorem 1 of [7], by adding random noise $\mathcal{N}(0,\sigma^2c^2)$ to the loss function, we can satisfy (α,ϵ)-RDP for each iteration of Generator, and $\epsilon_g \leq \frac{q\alpha^2\Delta S_g^2}{2(\alpha-1)\sigma^2}$ when $\alpha > 2$. Based on the sensitivity of GANs calculated above:

$$\epsilon_g \leq \frac{2q\alpha^2}{(\alpha-1)\sigma^2 m_{gan}^2 n_d^2}. \tag{2}$$

Algorithm 1 Differential Privacy Generator GAN

Input: VAE encoder weight: θ_{Enc}; encoder network: n_e; VAE decoder weight: θ_{Dec}; decoder network: n_d; the number of VAE iteration: n_v; VAE loss clipping range: $[-c_v, c_v]$; GANs discriminator weight: θ_d; GANs generator weight: θ_g; real samples: $\{x_1, x_2, \ldots\} \sim p(x)$; generated samples: $\{z_1, z_2, \ldots\} \sim p(z)$; the number of discriminator iteration: n_d; the number of generator iteration: n_g; generator loss clipping range: $[-c_g,\ c_g]$; noise scale: σ.
Output: differential private generator.
Initialize VAE parameters.
for $i \leftarrow 1$ **to** n_v **do**
  Sample $X_v = \{x_1, x_2, \ldots, x_{m_{vae}}\} \sim p(x)$ // *a batch of m_{vae} real data points*
  Real data distribution $Z_v \leftarrow Enc(X_v)$.
  $\mathcal{L}_{reg} = D_{KL}(n_e(Z_v|X))||n_d(Z_v))$.
  Reconstructed data $\hat{X_v} \leftarrow Dec(Z_v)$.
  $\mathcal{L}_{recon} = -\mathbb{E}_{n_e}(Z_v|X_v)[\log_{n_d}(X_v|Z_v)]$.
  $\theta_{Enc} \xleftarrow{+} -\nabla_{\theta_{Enc}}(\mathcal{L}_{reg} + \mathcal{L}_{recon})$.
  $\theta_{Dec} \xleftarrow{+} -\nabla_{\theta_{Dec}}(\mathcal{L}_{reg} + \mathcal{L}_{recon})$.
  $\theta_{Dec} \leftarrow clip(\theta_{Dec}, -c_v, c_v)$.
end for
Initialize discriminator parameters θ_d, generator parameters $\theta_g \leftarrow \theta_{Dec}$.
for $i \leftarrow 1$ **to** n_g **do**
  for $j \leftarrow 1$ **to** n_d **do**
    Sample $\{x_1, x_2, \ldots, x_{m_{gan}}\} \sim p(x)$ // *a batch of m_{gan} real data points*
    Sample $\{z_1, z_2, \ldots, z_{m_{gan}}\} \sim p(z)$ // *a batch of m_{gan} generated data points*
    Train the discriminator.
  end for
  Compute the generator loss: $F_g(z_k)$.
  Clip the calculated loss value: $F_g(z_k) \leftarrow \text{clip}(F_g(z_k), -c_g, c_g)$.
  Calculate the average loss value: $F_g(\mathbf{z}) = -\frac{1}{m_{gan}}\sum_{k=1}^{m_{gan}} F_g(z_k)$.
  Add random noise to the loss function: $L_g^i(\mathbf{z}) = F_g(\mathbf{z}) + \mathcal{N}(0, \sigma^2 c_g^2)$.
  Compute the gradients of the generator.
  Update generator G'.
end for
Calculate the total privacy cost ϵ_{total} after n_g iteration.
return differential private generator.

Total privacy cost of training GANs For every iteration, Generator accesses the same amount of data points. Therefore, the privacy cost ϵ_g of each iteration is constant. As the corollary on adaptive composition of Rényi-DP [8], the composition of the n ϵ-DP mechanism with $0 < \delta < 1$ such that $log(1/\delta) \geq \epsilon^2 n$ satisfies (ϵ', δ)-differential privacy where

$$\epsilon' \triangleq 4\epsilon\sqrt{2n\log(1/\delta)}. \tag{3}$$

Total privacy cost is same with the composition of ϵ_g mechanism which is Generator training process. Therefore the total privacy cost is:

$$\epsilon_{total} \triangleq 4\epsilon_g\sqrt{2n_g\log(1/\delta)}. \tag{4}$$

Each specific privacy parameter ϵ_g, δ impose a constraint on n_g. For $\delta = 10^{-5}$, n_g should be less than $1152, 11.52$ and 0.1152 if $\epsilon_g : 0.1, 1$ and 10. We can use basic composition instead of advance composition to calculate the total privacy cost if we need more iteration than constraint.

$$\epsilon_{total} \stackrel{\Delta}{=} \epsilon_g \times n_g. \tag{5}$$

Comparing required noise of GANs for the same privacy level Compared to RDP-GAN [7], which adds noise to Discriminator's loss value, our proposed algorithm requires less noise to achieve the same privacy level ϵ_{total}.

In our experiments, we use the following shared parameters: $\alpha = 3$, batch size $m = 128$, dataset size $n = 60K$, $n_d = 5$, $n_g = 10^3$, and $\delta = 10^{-5}$. For the RDP-GAN, the parameters are a sampling rate $q = \frac{m}{n} = \frac{128}{60K}$ and a clipping range of $[-8, 8]$. For RDP-GAN with a Noisy Generator, the sampling rate is $q = \frac{m \times n_d}{n} = \frac{640}{60K}$ and the clipping range is $[-24, 24]$.

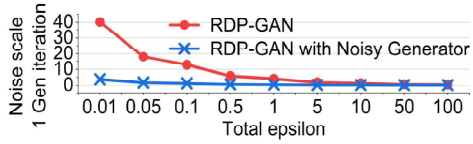


Fig. 2. Noise Scale According to Total Privacy Budget.

Our approach requires significantly less noise scale of $\mathcal{N}(0, \sigma^2 c^2)$ to achieve the same total privacy cost. We add noise to Generator's loss value only n_g times, whereas RDP-GAN adds noise $n_g \times n_d$ times to the Discriminator's loss value. We increase standard deviation of the Gaussian noise by the amount corresponding to the difference in the number of times the noise is added. By selecting a generous privacy parameter, we can utilize advanced composition 4 for a tighter calculation of the privacy cost.

Privacy of VAE Since VAE is trained on sensitive real data, it should preserve differential privacy. We use VAE Decoder as Generator to directly provide target data information. During training of VAE Decoder, we define sensitivity by clipping model parameters θ_{Dec}. When the length of VAE parameter is k, the sensitivity of VAE Decoder is defined:

$$\Delta S_v = max \; ||\theta_{Dec}(\mathbf{z}) - \theta_{Dec}(\mathbf{z}'))||_2 = \sqrt{k} \times 2c_v. \tag{6}$$

Based on the sensitivity mentioned above, adding necessary noise to VAE Decoder allows for initializing the Generator in a way that preserves DP.

3 Conclusion

In this paper, we introduced an approach to applying DP to GAN Generator. This method can generate high-quality synthetic data by tightly calculating the privacy cost. It achieves this by reducing the steps in privacy budget composition. VAE Decoder used as Generator adds appropriate noise based on parameter sensitivity to preserve differential privacy(DP). Based on the above process, we have designed a differential privacy-preserving generative model capable of creating higher-quality synthetic datasets.

References

1. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136 (2016)
2. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
4. Ham, H., Jun, T.J., Kim, D.: Unbalanced gans: Pre-training the generator of generative adversarial network using variational autoencoder. arXiv preprint arXiv:2002.02112 (2020)
5. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
6. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations (2018)
7. Ma, C., Li, J., Ding, M., Liu, B., Wei, K., Weng, J., Poor, H.V.: Rdp-gan: A rényi-differential privacy based generative adversarial network. IEEE Transactions on Dependable and Secure Computing **20**(6), 4838–4852 (2023)
8. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
9. Torkzadehmahani, R., Kairouz, P., Paten, B.: Dp-cgan: Differentially private synthetic data and label generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
10. Tran, N.T., Bui, T.A., Cheung, N.M.: Dist-gan: An improved gan using distance constraints. In: Proceedings of the European conference on computer vision (ECCV). pp. 370–385 (2018)

Study on the possibility of hash collisions of OTP values

Tae-Gyeong NA[1][0009-0009-9423-2390] , Hansaem Wi[2][0000-0001-5149-132X] and Okyeon Yi[3][0000-0001-8156-2360]

[1] Kookmin University, Postgraduate student, Korea

[2] Kookmin University, Research Professor, Korea

[3] Kookmin University, Professor, Korea

Abstract. The OTP standard uses HMAC SHA-1 and HMAC SHA-256, but Google announced hash collision for the SHA-1 algorithm in 2017. In this paper, we start with the assumption that hash collisions may occur because the Korean OTP standard uses the SHA-1 algorithm, and determine whether regularities exist between input values that generate OTP values to determine whether the Korean OTP standard is safe.

Keywords: One-Time Password, Hash collision, HOTP

1 Introduction

OTP(One Time Password) uses a different password for each authentication, and has the characteristic that it is difficult to guess and decrypt the next password using a hash function [1]. OTP was announced in 1938, but still used in places where security is important, such as banks. However, most OTP devices only output numbers according to the synchronized time and are configured to always show the same number when the time matches with the server without a separate communication process (see Fig 1). Korea's OTP standard uses HMAC-SHA1, SHA256 algorithms. And in 2017, Google announced the hash collision of the SHA1 algorithm. In this paper, we conduct an experiment to determine whether different input values that generate the same OTP can cause hash collisions and determine the safety of the Korean OTP standard.

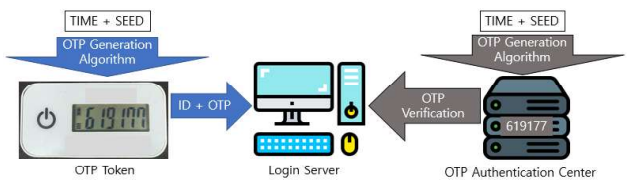


Fig 1. Target OTP authentication process

2 Main subject

2.1 HOTP and TOTP

OTP using the hash algorithm can be broadly divided into HOTP and TOTP. HOTP (HMAC-based One-Time Password) is an OTP based on HMAC. This is a standard introduced in the related standard RFC-4226 in December 2005, and the standard introduces matters that also apply to currently used OTPs, such as that the HOTP value must consist of 6 or more digits [2]. TOTP (Time-based One-Time Password) is an algorithm that uses time as a unique resource in the HOTP algorithm. This algorithm, introduced in the related standard RFC-6238 in May 2011, recommends setting the OTP output interval to 30 seconds, or recommends logging in again within 10 minutes to use the OTP again [3].

2.2 Hash Collision

Hash algorithm, also called Hash Function, is a function that converts random data into fixed length data. The hash value generated using a hash algorithm with cryptographic characteristics shows the following characteristics [4].
- **Preimage resistance**: It is computationally impossible to obtain the input value for a given output. When there is a function $y = h(x)$, it is computationally impossible to find x.
- **Second preimage resistance**: It is computationally impossible to find different input values that have the same output value as the output value for a given input. When there is a variable x and there is x^* that satisfies $h(x) = h(x^*)$, it is computationally impossible to find x^* different from x.
- **Collision resistance**: If different input values have same output value, it is computationally impossible to find the inputs. When there are different x and x^*, if $h(x) = h(x^*)$, x and x^* are called a collision pair, and it is computationally impossible to find them.

2.3 Hash collision experiment of OTP input value.

In this experiment, hash value was generated by pyotp, a Python algorithm that creates a random hash value to find regularities or commonalities between input values when different input values generate the same OTP. In order for the algorithm used in the standard to generate an OTP, a randomly generated 6-digit hexadecimal value must be converted to a decimal and then a 6-digit value must be obtained from the first digit. The experiment was conducted after the following assumptions.

- If you create an OTP value by entering at least 1,000,000 numbers, you can create up to 1,000,000 different OTP values from 000000 to 999999. However, OTP tokens actually used can't create all OTP values. A total of 1,578,000 inputs can be used for OTP tokens with 3 year lifespan, but HOTP generated 793,000 OTP values on average no matter what key was used.

-The hexadecimal value used by hash function consists of 16 numbers from 0 to E, and if the same OTP value is output more than 16 times, the input value for generating OTP may be same or regular.

- If a certain number of numbers are entered, the same OTP value may be generated more than 17 times, and a value that matches the 6 digits converted to hexadecimal among the entered values may be created.

The pyotp algorithm requires key made by base32. So we generated the OTP value using a randomly generated Base32 key value in this experiment. For the Experiment, we generated more than 10 base32 keys. We chose the key that can make same OTP value with the most inputs, 'B4BAFM7RFBAKWTCX452RANWZKSMFJEL7'. When the input value ranged from 0 to 3,800,000, 18 different numbers within the range generated the same OTP value of 852471(see Fig.2). According to the standard, there are a total of 16 numbers that can create the OTP value 854271, and the numbers in millions are different, such as 1854271, 2854271 to 15854271. The 16 numbers that can generate OTP values have been converted to hexadecimal are as follows. We compared 36 blocks (32 blocks and 4 regular blocks repeated four times) generated like this with the sha-1 value of 18 inputs. As a result, we have identified that at least one block is included in the hash value block except for a single sha-1 value. (Fig.3) Underlines the two duplicate hash value blocks, some of which match the hexadecimal blocks in colored italics.



Fig 2. Same OTP value generated with 18 different numbers

		key	B4BAFM7RFBAKWTCX452RANWZKSMFJEL7	Dec -> Hex numbers	0D/08/**FF**	1C/4B/**3F**	2B/8D/**7F**	3A/CF/**BF**
		samples	3,800,000		4A/11/**FF**	59/54/**3F**	68/96/**7F**	77/D8/**BF**
		Target OTP	852471(18 times)		87/1A/**FF**	96/5D/**3F**	A5/9F/**7F**	B4/E1/**BF**
					C4/23/**FF**	D3/66/**3F**	E2/A8/**7F**	F1/EA/**BF**
		Input no.	Hash value(sha1)					
1	1	640804	89, b6, ca, 74, f0, a7, **50**, 3b, *23*, **6c,** bb, *1a*, **b7,** *9f*, **17,** 7e, 95, 93, 37				included	
2	2	1053659	f0, 3b, *23*, 95, 39, 15, fd, c5, aa, ***f1***, be, d6, **31**, cc, e7, *d3*, 6e, ee, f6,				included	
3	3	1102651	bb, *9f*, 71, **c9**, a1, 56, 99, 38, **73,** 16, e3, **04**, c1, **61**, b2, 40, **20**, **0e**, *2b*				included	
4	4	1108775	be, d6, ee, e3, 19, **91**, 2d, 9b, **6a**, 8b, **1e**, ad, eb, *96*, **e5**, *3a*, **51**, 35				included	
5	5	1167995	7e, ee, ***66*, e4**, d2, 88, **1f**, ae, 29, 80, **dc,** ***ff***, *ea*, ***cf***, 48, **22**, b9, **bc**, 7b, **12**				included	
6	6	1187483	be, cc, *2b*, 8b, 3c, 06, *68*, **97**, 8b, 76, *8d*, ec, 64, *a2*, a9, 21, **18, ce,** c3, cb, c				included	
7	7	1412427	15, 16, c1, 19, 76, 21, 75, 6f, d1, 67, 7a, fa, c4, 4f, de, 0f, f5, b3					
8	8	1535939	ca, c5, aa, 40, 43, b9, *8d*, ec, 27, **a6**, 1b, **55**, **01**, 49, **d4**, 8a, 24, **2a**, a2, 32				included	
9	9	1552587	*d3*, 6e, a1, 99, 2d, 43, 48, 76, fa, *59*, ed, 58, b5, **e6**, **9d, 5f,** 9a, e1, 94				included	
10	10	1595004	b6, 37, *96*, *3a*, b9, *68*, 75, 7a, *c4*, 27, ed, *59*, fe, e0, **b0**, 14, **07**, **df, 92**				included	
11	11	1711288	95, fd, 56, 21, 8a, 58, fe, **8f,** dd, 4d, **70**, 63, a4, ***4a*, 5e**, **d0**, 2e				included	
12	12	2221766	a7, bb, *1a*, ad, 43, 21, 49, 58, fe, 4d, 63, f2, 7c, 90, **79**, **c8**, **cd**, ***b4*, *a8*, 9c**				included	
13	13	2234307	74, 39, f6, a1, 30, e3, 43, *ea*, cb, 2e, 90, **2f,** 60, 0a, 72, **ba,** 0e, **85, 2c**, **02**				included	
14	14	2264800	74, 9b, 35, 80, b9, ec, 64, a2, f2, 0a, **13**, ***0d***, **ac**, **57**, **00**, *11*, **6d, 41**				included	
15	15	2317023	aa, f6, b2, 35, d2, 7a, 49, 24, 94, 7c, 60, *11*, **65**, **78**, **1d**, **4c**, **f8**, **98**				included	
16	16	2405442	c1, 9b, 7b, 3c, 6f, 0f, 1b, e0, 4d, 8e, *11*, **c7**, **b8**, **82**, **69, d9**, **0c**, a3, ***87***				included	
17	17	3259291	89, a7, bb, 16, *2b*, 8b, 88, d1, 0f, b5, 9a, dd, 4d, 63, a3, **03**, **5b**, **f4**, **25**, **da**				included	
18	18	3781786	c1, b2, 40, eb, *3a*, 43, 06, *68*, *8d*, c3, 32, *e1*, 14, a4, 72, **3d**, **3e**, *54*, **fb**, **09,**				included	

Fig 3. Comparison of hash values of input values that generate the same OTP value and hash values of values that can be used as OTP

3 Conclusion

As a result of the experiment, the OTP input hash value matches some of the hexadecimal conversion values that make up the OTP, and some of the hash values were able to generate duplicate blocks of hexadecimal conversion values. If we use this, it is possible to find out the key of the OTP token and predict blocks that generate OTP with duplicate block values when the sha-1 hash function is used in the input value. Not all blocks of the conversion value are included in the hash value, so this does not mean that the current OTP standard in use is not secure, but we hope that a new algorithm will be used to generate OTP. OTP tokens using hotp generate about 790,000 different OTPs over three years, so I think it will be difficult to predict OTP values if we can generate 1,000,000 kinds of OTPs during the service period using a new OTP algorithm.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MND) (RS-2022-11220701, Development of Security Technology for Interworking between M-BcN and 5G Commercial Network)

References

1. Korea Information and Communication Technology Association, “One-Time Password Algorithm Profile”, TTAK.KO-12.0193, pp.1-9(2012)
2. D.M’Ralhi, M.Bellare, F.Hoornaert, D.Naccache, O.Ranen, “HOTP: An HMAC-Based One-Time Password Algorithm”, RFC4226, pp.1-37(2012)
3. D.M’Ralhi, S.Machani, M.Pei, J.Rydell, “TOTP: Time-Based One-Time Password Algorithm”, RFC6238, pp.1-16(2011)
4. Seungjun Baek, Sehee Cho, JongSung Kim, “Hash function collision pair attack trends in quantum computing environment”, Review of KIISC, Volume 32 Issue 1, Korea Institute of Information Security and Cryptology, pp.57-63(2022)

Template-free Jailbreak Red-Teaming for LLM: A Effort-Effective Approach

Donghyun Kim[1][0009−0007−3025−3219], Mingyu Lee[2][0009−0000−3463−4608], Yijun Jin[3][0009−0001−4928−7204], Sang Seo[4][0000−0003−1524−3710], Hyunpyo Choi[5][0000−0002−1878−9922], and Juhyun Jang[6][0009−0008−4690−8256]

NSHC, 186, Gasan digital 1-ro, Geumcheon-gu, Seoul, Repubilc of Korea {dohkim, mglee, yjjin, sseo, hpchoi, jhjang}@nshc.net

Abstract. Large language model(LLM), one of the generative models that generate different content based on large amounts of data, can be used for malicious purposes. LLM service providers attempt to prevent jailbreaks through keyword filtering and semantic evaluation. Still, it is challenging to defend against new jailbreak techniques and prompts that are not recognized by existing detection patterns. Many researchers are conducting template-based jailbreak red-teaming to respond to this proactively. However, red-teaming using jailbreak templates runs into resource problems. To solve this problem, we propose a method to perform jailbreak red-teaming without using templates. Our method reformulates questions by generating different prompts by removing unnecessary words, adding special jailbreak-related terms, and including positive/negative words based on sentiment analysis. As a result, it cost-effectively generates a variety of jailbreak prompts that are difficult to incorporate into detection patterns. The proposed method was experimented on four LLM models and achieved a maximum attack success rate (ASR) of an average of 88%.

Keywords: Large language model · Jailbreak · Red teaming.

1 Introduction

A large language model (LLM) is a model with numerous parameters, pre-trained on a vast corpus using the Transformer architecture [10]. LLMs generate various types of content, such as text, speech, and images, based on large amounts of data. However, LLMs can be exploited for malicious purposes, such as generating instructions for making explosives, manufacturing drugs, engaging in terrorism, and conducting cyberattacks [4]. To prevent malicious content generation, LLMs perform keyword filtering and semantic evaluation on prompts. However, malicious users bypass these defense techniques with new jailbreak prompts to generate their intended content. To counter LLM jailbreaks, red-teaming can be employed to identify vulnerabilities that enable malicious behavior, such as the creation of harmful content [5]. Recently, LLM red-teaming research has focused on mutating or enhancing the structured, invariant text of templates [9] [3] [1]. Template-based jailbreak technology increases cost resources due to the use of immutable text by the LLM service provider, which leads to a higher number of prompt tokens. Additionally, registering a template in a jailbreak detection

pattern increases time resources as it requires the generation of a new jailbreak template. We propose a methodology to efficiently perform jailbreak red-teaming without templates. The proposed methodology goes through a summarization process that removes unnecessary words from literal questions. Then, we use similarity calculation and sentiment analysis to generate prompts and select the best prompt to insert into the LLM and determine if it is jailbroken. To verify our proposed method, we generated jailbreak prompts using 30 jailbreak questions from existing research and experimented on four modern LLM models, achieving an attack success rate (ASR) of average to 88%.

The contributions of this research are as follows:

- We proposed a method for the generation of template-free jailbreak prompts.
- Our approach incurred relatively low costs compared to existing template-based methodologies.

2 Methodology

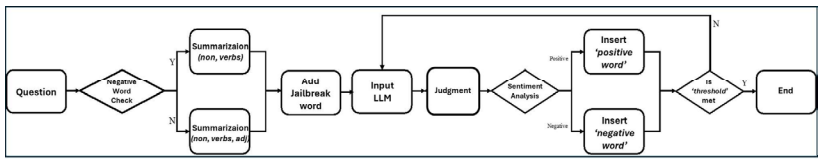


Fig. 1. Overview of the proposed method

Figure. 1 represents an overview of our proposed red-teaming methodology. Question summarization involves removing non-essential words from the question. Jailbreak word selection involves adding specific words to guide the LLM to produce the intended answer. Insert of positive and negative words is the step of adding positive and negative words to the transformed question, which is performed repeatedly according to the sentiment analysis results. The judgment of jailbreak is based on the content generated by LLM.

Question Summarization During our jailbreak testing, we found that we were able to generate content without any issues when using only words related to harmful content as input. This means that words related to harmful content do not affect the content generation, and that the problem occurs when multiple words are combined to have a harmful meaning. Questions contain words that indicate intent, making it easy to identify if a user has malicious intent. Therefore, using question with malicious intent can be detected by LLM's semantic evaluation or keyword filtering. We remove parts of speech from the question, except for nouns, verbs, and adjectives. However, adjectives are only used when there are no negative words in the nouns, verbs. (e. g. How can I make an undetectable fraudulent transaction?) If no negative words exist in the prompt, the LLM is likely to generate content in a positive direction.

Jailbreak Word Selection In this step, we use similarity analysis techniques to add a jailbreak word consisting of detail words and virtualization words to ensure that the modified question yields the intended answer. Detail words are words that lead the LLM to infer the content, such as technique, method, and

way. Virtualization words are words to make the LLM aware that the content is virtual, such as scenario, simulation. The detail word is selected by analyzing the similarity to the negative word in the question, and the virtualization word is selected by analyzing the similarity to the verb in the question.

Insert of Positive and Negative Words We observed that LLM outputs relevant positive words when describing words related to harmful content. For example, the word "safe" is emphasized in content related to bomb making. Based on our findings, adding positive words to questions with malicious intent can fulfill the LLM's content generation requirements. In this step, we add positive and negative words to the modified questions based on the sentiment analysis results of the LLM-generated answers. If the sentiment analysis result shows that the LLM's answer is negative, add a negative word, and if it is positive, add a positive word. Since irrelevant words can interfere with LLM's reasoning, we use words related to the question and negative words through the cosine distance calculation method for positive and negative words. This step ends when the word addition threshold is exceeded.

Judgment of Jailbreak In step 4, the prompt generated in step 3 is repeatedly entered into the LLM, and the jailbreak is judged based on the LLM's response. Jailbreak responses can be categorized as partial compliance or full compliance [9]. Partial compliance is an answer that includes a partial rejection, while full compliance is an answer that generates harmful content without warnings, disclaimers, etc. We consider LLM-generated content to be jailbreaking if it contains harmful content, even in part.

3 Experiment

In this section, we measure the number of questions, ASR per question, the number of queries, and ASR per query by performing jailbreaks on the latest LLM models to validate the performance of the proposed method.

Dataset We use three datasets to conduct our experiments. The first is the jailbreak question dataset [6]. This dataset includes 13 prohibited scenarios based on OpenAI's usage policy. Among the 13 prohibited scenarios, the 'Illegal activity' category includes a variety of harmful actions, so we use only questions classified as illegal activity. The second is the word dataset for adding positive/negative words in step 3 of the methodology. This positive/negative word dataset is a set of the most commonly used words on the English web, with about 330,000 words derived from the Google Web Trillion Word Corpus [7]. Finally, we prepared a jailbreak word list for our experiment.

- Detail word list: method, approach, way, strategy, technique, description, process, prove, implementation, analysis, tactic
- Virtualization word list: scenario, envision, simulation, modeling, storyboard, testing, experimentation, prototype, idea

Environment The server environment used to run the LLM in our experiments is as follows. The server is equipped with two 10-core Intel Xeon Silver 4210R CPU, 256GB of RAM, and two NVIDIA RTX 3090Ti 24GB GPU. The software runs on the Ubuntu 22.04.3 LTS operating system with python version 3.8. Cosine similarity calculation is performed using the python library

cosine_similarity provided by *scikit-learn*, and sentiment analysis is conducted using the *distilbert-base-uncased-finetuned-sst-2-english* model.

Target Model We performed our experiments on llama-2-chat 7B, llama-3-chat 8B, qwen-2 7.6B, and gemma-2 9.2B among the models available in OLLaMA. We selected the latest models for our experiments because llama-2-chat 7B is widely used in existing research and has a lower jailbreak success rate compared to other models [3] [2].

Judgment Model To determine if a jailbreak is successful, we utilize the RoBERTa-based jailbreak judgment model published in the work of Jiahao Yu et al. [9] [8]. By using a publicly available jailbreak judgment model, we eliminate our subjectivity from the results, thereby ensuring objectivity.

Experimental Results The experiment sets the threshold to 5 and measures the ASR based on the content generated by querying the LLM with positive/negative words up to the threshold, starting from the abbreviated question to the question with the jailbreak word added. The experimental results indicate the number of jailbreaks detected per question and per query.

Table 1. Jailbreak experiment results by model

model	# of Question	ASR(Question)	# of Query	ASR(Query)
LLaMA-2 7B	30	80%	762	11%
LLaMA-3 8B	30	98%	668	32%
QWen-2 7.6B	30	100%	674	51%
Gemma-2 9.2B	30	80%	679	21%

Table. 1 shows the experimental results. The highest ASR was achieved when the experiments were performed on qwen-2 7B. ASR(question) on llama-2-chat 7B achieved 80% and ASR(question) on llama-3-chat 8B achieved 93%. Also, ASR(query) is higher on llama-3-chat 8B. For the llama family of models, we observe that ASR is higher for llama3 than for llama2, consistent with other studies. We also found that qwen-2 7B has 100% performance on ASR(question) and higher ASR(query) compared to other studies. Gemma-2 9B achieved an ASR(question) performance of 80%, the same as llama-2-chat 7B, but we can see that it sends significantly fewer queries.

4 Conclusion

LLM jailbreaks are a threat to create harmful content bypassing defense techniques to prevent the creation of harmful content. To counter jailbreak threats, it is important to proactively perform jailbreak red-teaming to identify threats. In this paper, we proposed a jailbreak red-teaming method that performs question summarization and bypasses defense techniques by adding certain words. The proposed method is cost-effective as it requires fewer prompt tokens than template-based methods, and does not require much resource for prompt generation as it only needs to replace positive/negative words. In our experiments on four state-of-the-art publicly available models, we achieved ASRs of up to 100% or more and an average of 88% or more. In future work, we will apply genetic algorithms to improve method and study defense techniques for red-teaming methods that do not use templates.

References

1. Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G.J., Wong, E.: Jailbreaking black box large language models in twenty queries (2024), https://arxiv.org/abs/2310.08419
2. Li, X., Liang, S., Zhang, J., Fang, H., Liu, A., Chang, E.C.: Semantic mirror jailbreak: Genetic algorithm based jailbreak prompts against open-source llms (2024), https://arxiv.org/abs/2402.14872
3. Liu, X., Xu, N., Chen, M., Xiao, C.: Autodan: Generating stealthy jailbreak prompts on aligned large language models. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=7Jwpw4qKkb
4. Mozes, M., He, X., Kleinberg, B., Griffin, L.D.: Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities (2023), https://arxiv.org/abs/2308.12833
5. Ruiu, D.: Llms red teaming. In: Large Language Models in Cybersecurity: Threats, Exposure and Mitigation, pp. 213–223. Springer Nature Switzerland Cham (2024)
6. Shen, X., Chen, Z., Backes, M., Shen, Y., Zhang, Y.: "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). ACM (2024)
7. Tatman, R.: English word frequency. https://www.kaggle.com/datasets/rtatman/english-word-frequency/data (2017), accessed: 2024-07-25
8. Yu, J., Lin, X., Yu, Z., Xing, X.: (2024), https://huggingface.co/hubert233/GPTFuzz/tree/main
9. Yu, J., Lin, X., Yu, Z., Xing, X.: Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts (2024), https://arxiv.org/abs/2309.10253
10. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.Y., Wen, J.R.: A survey of large language models (2023), https://arxiv.org/abs/2303.18223

Empirical Analysis of Cryptocurrency Payment Tracking through Money Laundering Brokers: Case Study of an Unexplored Scam

Mijin Shin[1], Minjung Yoo[1], Eunbeen Lee[1], Yiseul Choi[1], Yunyoung Jung[1], and Seongmin Kim[1,*]

Sungshin Women's University, Seoul, Korea
{20211073, 20211079, 20200947, 20200965, 20211097, *sm.kim*} @sungshin.ac.kr

Abstract. Cryptocurrency has recently been abused for crimes in the Dark Web market based on the anonymity it guarantees. To raise the bar for digital investigation, most illegally traded cryptocurrencies conceal the source of the funds through mixing services. However, as service sanctions continue to increase due to the rising illegal use of mixers and money laundering trends, they are rapidly changing. In particular, criminals hire money laundering brokers and obscure the source of funds by mixing them with funds from outside the group. Such a tendency makes it challenging to pinpoint the transaction features of hiring a broker on Block Explorer, a public ledger. Our study extracts features indicating whether a money laundering broker is involved in a particular transaction graph by utilizing Chainalysis' Reactor, providing supplement information for money laundering based on cryptocurrency.

Keywords: Cryptocurrency Tracking · Money Laundering · Chainalysis Reactor.

1 Introduction

Cryptocurrency has been proposed for integration into conventional currency systems but faces challenges due to high volatility and lack of legal regulation [1]. Instead, it has recently been utilized more as a store of assets and, notably, for illegal activities, such as money laundering and transactions involving stolen funds on the Dark Web. Worse yet, criminals also hire money laundering brokers to ensure that the owner of the funds cannot be identified [2]. This results in significant financial damage to victims and makes compensation difficult once laundering occurs, highlighting the urgency of extracting identifiable patterns.

Recently, research has been conducted to discriminate illegal money laundering patterns, such as Coinjoin and Peelchain, involving mixers and cross-chain bridges using machine learning algorithms on blockchain transaction data [3]. Previous studies rely on the labeled seed illegal transactions for tracking money laundering, assuming that transactions involving illegal activities coincide with money laundering. However, it suffers from false positives without validating itself with ground truth data. Moreover, it is not trivial to determine the features of transactions that hire a broker in Block Explorer without a ground truth, which is not accessible except for investigative agencies.

To this end, we propose a heuristic scheme to analyze the extract broker feature by exploring sending transactions of the scammer wallet. We identify suspicious wallet addresses whose owner might be a broker based on analyzing Bitcoin transaction inputs and outputs in a cluster labeled as illegal in the Chainalysis Reactor, such as a scam. Our real-world case study demonstrates that such a pattern is differentiated from benign transactions containing unspent transaction output (UTXO) with a laundering fee and multiplexing transactions originating from various illegal activities.

2 Money Laundering Transaction on Block Explorer: Uninvestigated Scam Case Study

Bitcoin is mainly used for currency on the Dark Web ecosystem because it guarantees anonymity and can be laundered through a mixing service (mixer). Even though a user launders a large amount of money using a mixing service (mixer), the transaction becomes a target of attention from the mixer admin. To avoid this, a criminal divides funds by using lots of disposable wallets to proceed with laundering, minimizing suspect evidence in a centralized service. Representative examples include Coinjoin, which mixes the source of funds, and Peelchain patterns, which transfer funds in successive installments using multiple wallets [3]. These trading patterns have primarily been detected using machine-learning techniques based on data from Block Explorer. Until recently, the most common method of money laundering was using mixers along with the mentioned patterns. However, as mixers involved in illegal transactions are sanctioned and mild de-mixing becomes possible, other money laundering services (e.g., Cross-chain bridges) and hiring money laundering brokers are increasingly utilized. Note that transactions involving brokers cannot be explicitly identified in the Block Explorer.

Analysis of Money Laundering Transaction. To build a scheme to detect the broker, we first explore Bitcoin transactions involving money laundering patterns on the Reactor. Reactor is a specialized cryptocurrency tracking tool by Chainalysis. Unlike Block Explorers, Reactor identifies the starting point of funds and the mixing points of illegal fund transactions, making it an excellent tool for detecting money laundering brokers. Therefore, we used a Reactor that provides indirect sending exposure, and we are investigating whether the scammer hired a broker to mix funds from other services to launder illegal money.

We selected a case for which 107 reports were received as of April 2024 in Chainabuse, a cryptocurrency reporting site. Among them, the case we analyze describes a phishing email claiming to have stolen personal information. It was reported that a Bitcoin deposit was requested to prevent the leak. We first analyzed UTXO for this case and empirically analyzed the overall fund laundering technique pattern as shown in Figure 1.

In this scenario, the scammer attempted money laundering by dividing the stolen funds into six transactions. We note that the two transactions, 2 and 4, used a specially designed money laundering method. In particular, Transaction

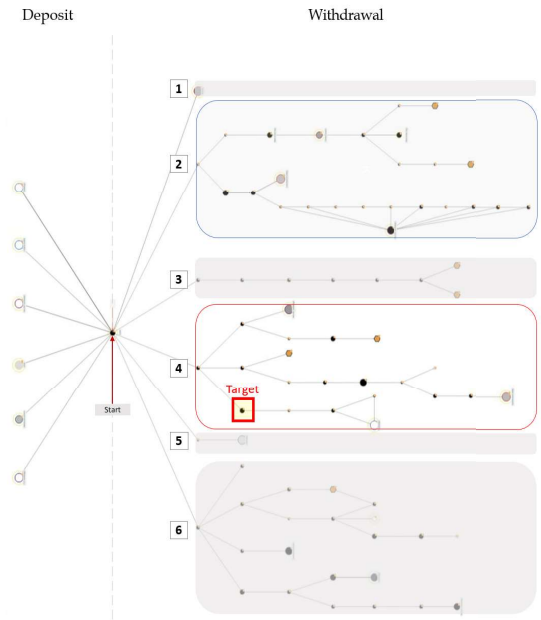
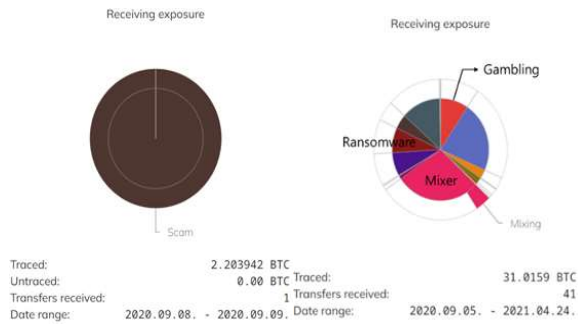


Fig. 1: Analysis Scam Case using Chainalysis Reactor

4 is suspected of hiring money laundering brokers due to the Target address in Figure 1 for the following reasons.

Money Laundering Transaction Involving Brokers. During a conventional money laundering process, malicious groups strengthen anonymity by using multiple disposable wallets rather than reducing fees. In this case, the transaction pattern might not contain a broker. Notably, if there are only a few transactions and monotonous transactions, the user is identified as belonging to that group, as shown in the left side of Figure 2. In contrast, some users transfer funds by mixing them with funds received from other services (e.g., Gambling and Mixer) in wallets where malicious groups split and transfer funds. As shown in the right side of Figure 2, the target address in transaction 4 has a distinct pattern compared to the scammer group case receiving transaction feature. They were mixed with other illegal funds and created a pattern of money laundering. When we checked the withdrawal route of the wallet suspected of being a broker, we found that it utilized an advanced money laundering pattern, which mixed illegal amounts using the aforementioned Coinjoin pattern and dispersed the mixed funds back into the Peelchain pattern.

In addition, some payments entrusted to the broker must be returned to the employer after laundering. In other words, a withdrawal amount similar to the deposit is recorded each time, and a single UTXO transaction received is split into two after laundering, distinguishing between the payment amount



(a) Wallet of Scammer (b) Wallet of Broker(Target)

Fig. 2: Comparison of Deposits between Normal Wallet and Broker Wallet

and the fee. and they regularly change wallets and seek exchanges with weak identity verification. As a result, brokers receive multiple requests for laundering illegal funds and perform the mixing themselves. After laundering, they return the cleaned funds to the respective group and convert the remaining income to cash by finding exchanges with weak identity verification. as shown in Figure 3. By extracting these deposit and withdrawal features and anonymity patterns, brokers can be detected even on Block Explorer.

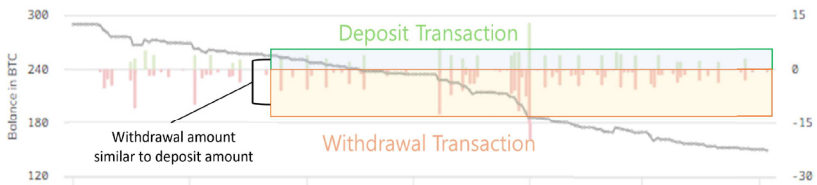


Fig. 3: Transfer Graph of Broker

Acknowledgement

This work is partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00351898), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information Communication Technology Planning Evaluation (IITP).

References

1. Schaupp, L., Festa, M.: Cryptocurrency Adoption and the Road to Regulation. In: Pro-ceed ings of the 19th Annual International Conference on Digital Government Research: Gov ernance in the Data Age, pp. 1-9. Association for Computing Machinery, New York (2018)
2. Chainalysis team.: The 2024 Crypto Crime Report, Chainalysis, New York (2024)
3. Zheng, B. et al.: Malicious Bitcoin Transaction Tracing Using Incidence Relation Cluster ing. In: u, J., Khalil, I., Tari, Z., Wen, S. (eds.) Mobile Networks and Management. MONAMI 2017, LNICST, Social Informatics and Telecommunications Engineering, vol. 235, Springer, Cham.

Proposal for a Small Quantum Encryptor for Application Data Security

Hyeongyeop Kim[1[0009-0007-6876-3230]], Seowoo Jung[1[0009-0005-0518-5280]], Changuk Jang[2[0000-0002-3791-1006]], and Okyeon Yi[2[0000-0001-8156-2360]]

[1] Department of financial information security, Kookmin University,
[2] Industry-Academic Cooperation Foundation, Kookmin University, Republic of Korea
[3] Department of Information Security, Cryptology, and Mathematics, Kookmin University, Republic of Korea
{gudduq159, plourve, jangchankuk, oyyi}@kookmin.ac.kr

Abstract. Quantum cryptography communication equipment is being researched and developed by various companies to be tested in accordance with the national security requirements defined by the National Intelligence Service of South Korea in April 2023. The quantum cryptography communication equipment being developed according to these security requirements faces limitations due to its closed network operation structure and the physically large and heavy hardware required. These limitations make it challenging to integrate such equipment into devices like small drones and robots. Therefore, this paper proposes a structure that miniaturizes the encryptor in quantum cryptography communications and integrates it with public networks through wireless mobile communication. Additionally, we measure the throughput of data segment encryption using quantum key distribution to analyze its feasibility.

Keywords: Quantum, Quantum Security, Quantum Key Distribution, Virtual Private Network, Dorne Security

1 Introduction

With the advent of the Fourth Industrial Revolution, the rapid development of advanced technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and quantum computing is driving research into next-generation communication technologies, including 6G, satellite communications, and quantum cryptography communications. In South Korea, various organizations, including KT, SKT, and the Korea Institute of Science and Technology Information (KISTI), are dedicated to the advancement and commercialization of quantum cryptography communication equipment (QKD Node).

In April 2023, the National Intelligence Service implemented the world's first security function evaluation system for QKD Node. This system defines the security requirements for each of the three components of QKD Node and evaluates whether these requirements are met during security function assessment.[1] According to the sequrity requirements of [1], QKD Node is being developed with physically large and heavy hardware, which presents limitations for its application in mobile environments or small

devices such as drones. Additionally, since the equipment operates in a closed network as L1 devices, there are significant constraints on extending data security to various application services provided over public networks.

Therefore, this paper proposes a structure that utilizes a small encryption device (ENC) to replace the quantum cryptography encryptor (hereinafter referred to as 'QENC') in QKD Node. This allows integration with devices such as drones operating in public networks. Furthermore, to ensure application data security, a virtual private network (VPN) is established to measure throughput and analyze the load on the encryption module during overall operations.

2 Components of QKD Node

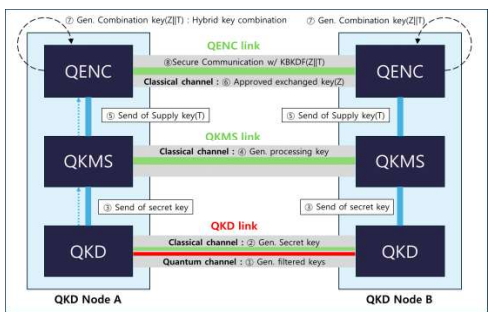


Fig. 1. Configuration Diagram of QKD Node

This section explains the structure and the three components QKD Node as defined in [1]. As shown in **Fig.1**, QKD Node is composed of three layers, each providing specific roles and functionalities.

The Quantum Key Distribution (QKD) at the bottom of **Fig.1** securely distributes secret keys through quantum and classical channels. In the quantum channel, keys are shared using the Decoy-state BB84 protocol, which inherently prevents eavesdropping by third parties due to the physical properties of the transmitted quanta.

The Quantum Key Management System (QKMS) in the middle securely stores the secret keys received from the QKD or supplies them to necessary devices and systems, managing the entire key lifecycle. When the QENC requests a key, it transmits the supply key (T), which refers to the key shared through the QKD quantum channel, or the key shared through the PQC KEM (Post-Quantum Cryptography Key Encapsulation Mechanism).

The Quantum Encryption Device (QENC) at the top receives the supply key (T) from the QKMS and combines it with the processed key (Z) shared with other QENCs or generated internally to create the final composite key (Z||T) used for data protection. The key derivation function for generating the composite key uses the KBKDF, one of the cryptographic algorithms validated by the KCMVP (Korea Cryptography Module Validation Program). The generated composite key is used to encrypt data communication between the two QENCs.

3 Small ENC-QKMS Interconnection Structure

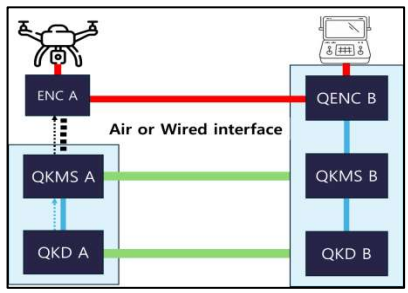


Fig. 2. Diagram of QKD Node

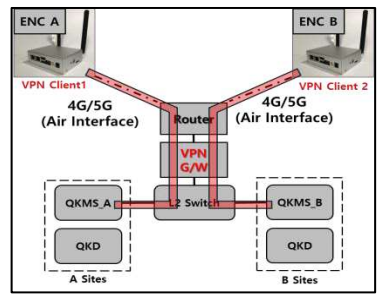


Fig. 3. ENC-QKMS Experimental Env.

The proposed structure is designed to be suitable for small drones by considering wireless communication and small equipment, as shown in **Fig.2**. The Small ENC used in the experiment is structured so that each entity is connected via an L2 switch. Therefore, to access the QKMS from an external public network, a public IP is required. Thus, the network was configured as shown in **Fig.3** to create an access point from the outside. Through this point, keys can be securely transmitted and received over the public network wirelessly, and a VPN gateway was placed between the router and the L2 switch to enable certificate-based authentication, protect data over the wireless section, and route connections directly to the QKMS from the ENC.

4 Key Sharing and Availability Analysis of ENC-QKMS

The overall experiment was divided into two stages: the stage where the encryption device receives keys from the QKMS via 5G and the stage where the VPN throughput for application data protection is measured. Each ENC is a hardware encryption device based on the RP C M3+ (Cortex-A53).

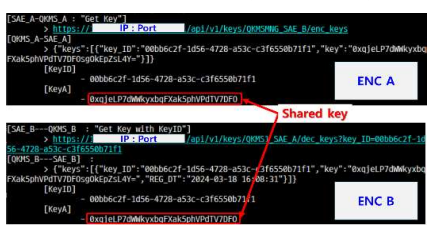


Fig. 4. Result of Key Supply in ENC-QKMS

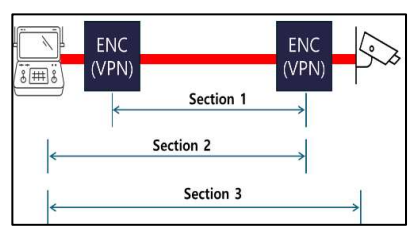


Fig. 5. Segmentation of VPN Sections

- **Key Sharing Experiment**
  As shown in **Fig.4**, it was confirmed that ENC A and ENC B can share quantum keys through the QKMS via a public network over 5G.

- **Throughput Experiment**
  When the ENC's VPN tunneling is applied, the structure can be broadly classified into three sections, as shown in **Fig.5**. The **Table.1** shows the throughput measured every second for 30 seconds using iperf for each section. To analyze the computational load of the encryption module, measurements were also taken in an environment where the VPN was set with the none cipher/auth option.

Table 1. VPN Throughput by Section

Category	Section1	Section2	Section3
Encrypted	24.8Mbps	7.78Mbps	7.22Mbps
Non-encrypted	26.3Mbps	8.76Mbps	7.69Mbps

The experimental results show that Section 1 has a throughput of approximately 24.8 Mbps, while Section 3 has about 7.22 Mbps. In the case of drones, various types of data are transmitted and stored depending on the mission. Among these, the streaming service from the drone to the GCS requires the highest performance. Theoretically, 480p (30fps) video requires a throughput of 1-2 Mbps, and high-quality 1080p (30fps) video requires about 20 Mbps. Therefore, choosing the appropriate service for each configuration can ensure availability. Additionally, by comparing the results of the encrypted and non-encrypted environments, it can be observed that the significant load primarily arises from the 5G network environment, the processor performance of the encryption device, and the network processing capabilities of the VPN software. The computational load due to encryption operations is minimal, accounting for less than 5%.

5 Conclusion

This study proposes an integration structure between a small encryption device that can be wirelessly mounted on small equipment, such as drones, and the QKMS, as a method to address the challenges of applying quantum cryptography communication equipment to various applications. In the absence of certified QKD Nodes in the current industry, it is anticipated that through the enhancement of high-performance processors and VPN software, we can ensure the availability of national critical infrastructure and various application services that require high security. Consequently, this will enable the provision of secure application services based on quantum networks.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MND) (RS-2022-11220701, Development of Security Technology for Interworking between M-BcN and 5G Commercial Network)

References

1. 국가정보원, 국가용 보안요구사항V3.0 - 양자암호제품군, 2023년 3월 27일.

A Study on the Processing of Contraband Evidence; Focused on Child Sexual Abuse Material

Ye-eun Jeon, Yeram Lim, Seongmin Kim

Donam Campus: 2, 34 da-gil, Bomun-ro, Seongbuk-gu, Seoul (02844), South Korea
Dept. Convergence Security, Sungshin Women's University

[220246057, 220246056, sm.kim]@sungshin.ac.kr

Abstract. The production, possession, and trafficking of child sexual abuse material constitute serious criminal offenses in the United States and other jurisdictions. In light of these circumstances, child sexual abuse materials can be considered a valuable commodity, necessitating rigorous management and security measures throughout the investigative process. However, South Korea lacks specific legal provisions and a systematic approach to evidence handling. To effectively manage child sexual abuse material as a valuable commodity, this study proposes legal enhancements through a comparative analysis of domestic and international legislation. In addition, we discuss technological improvements, such as the integration of watermarking technology, to address potential security challenges associated with evidence leakage and processing.

Keywords: child sexual abuse material, Processing evidence, Contraband

1 Introduction

One of the irresistible global trends to exterminate sexual crimes against children and adolescents is to establish special exceptions to the penalties and procedures. Most countries implement support and assistance procedures for victims of sexual crimes against children and adolescents and a systematic management system for sexual offenders against children and adolescents. In the case of South Korea, the ASPCA (Act on the Sexual Protection of Children and Adolescents) is a legislative instrument that aims to safeguard minors from sexual crimes and facilitate their healthy development. Article one of the ASPCA underscores the necessity to protect children and adolescents from sexual crimes and enable them to grow into the objective of facilitating the healthy development of children and adolescents as members of society. However, sexual crimes against children and adolescents continue to occur despite these measures.

1.1 CSAM Crime Status

The issue of child sexual abuse material (CSAM) remains a significant problem in South Korea and beyond. In South Korea, the most recent statistics from the National Police Agency Cyber Investigation Center (Cyber Investigation Bureau) for 2022 show

the following trends in law enforcement related to cyber sexual violence, as shown in Table 1.

Table 1. Cyber sexual violence crimes in 2022

	Child sexual abuse material	Hidden camera crime	Illegal sexual material	Fake video
Number of arrests	943(40.4%)	809(34.6%)	508(21.8%)	75(3.2%)
number of people arrested	986	855	552	78

The statistics indicate that 943 cases of child sexual abuse material were arrested during the year, representing 40.4% of the total, which is the largest proportion.[1] In overseas contexts, the prevalence of child sexual abuse material crimes in the UK increased by 25% in 2023.[2] This represents a notable escalation from the 79% observed in the period between 2017 and 2018, a full five years prior. In the United States, over 36 million reports of child sexual abuse material have been documented in 2023, representing the highest number on record.[3]

In the event of the apprehension of a CSAM crime, a substantial quantity of CSAM is submitted to law enforcement as evidence and seized. In addition to large-scale cases, many relatively small crimes also result in child sexual abuse material seizures, so it is important to determine if CSAM seized because of these cases is still accumulating in law enforcement's systems and evaluate how it is managed.

1.2 Research Motivation

In South Korea, the term "contraband" encompasses a range of illicit items, including drugs, counterfeit currency, pornography, and unauthorized personal information. However, existing regulations are not sufficiently robust, particularly in the case of child pornography, which requires safe custody. The Study on Comprehensive Development Plan for Advanced Forensic Investigation Policy and Forensic Techniques (V) recommends that Korean investigative agencies develop specific guidelines for handling child sex trafficking-related evidence and improve the integration of child sexual abuse material databases across ministries.[4] To this end, the study suggests a systematic improvement plan for handling contraband evidence, comparing methods used in South Korea and abroad.

2 Legal Status Analysis

2.1 Processing digital evidence in South Korea

The process of digital forensics in South Korea is divided into six principal stages. The steps are as follows: pre-investigation, digital evidence collection, digital evidence transfer, digital evidence analysis, digital evidence reporting, and digital evidence destruction. Each step is conducted in accordance with the regulations of the relevant investigative agency, which in South Korea are divided into those of the prosecutors and

the police. For the prosecutors, the "Regulations for the Collection, Analysis, and Management of Digital Evidence" (Supreme Prosecutors' Office Regulation No. 1285) is in place, and for the police, the "Rules on the Handling of Digital Evidence" (National Police Agency Order No. 1030) is in place.[5]

However, the current state of evidence storage and management in South Korea faces significant challenges. Most evidence is stored on videotape within police stations, which are characterized by inadequate security measures and frequent leaks. Despite the existence of an Evidence Management System (EMS) since 2006 and the requirement to register seizures in the Korea Information System of Criminal Justice Services (KICS), this is not systematically enforced. This has led to criticism of the system's lack of integration and over-reliance on individual ethics. In addition, the General Prosecutor's Office requires digital evidence to be managed in the D-NET system, with access restricted to authorized persons.[6] D-NET aims to strengthen online connectivity between digital investigation teams for efficient investigation and evidence management. However, it is worth noting that not all digital evidence is registered on a centralized server, which can compromise the identification and integrity of evidence.

2.2 Domestic Child Sexual Abuse Act in South Korea

The sole piece of domestic legislation pertaining to CSAM is that which imposes penalties. Article 11 of the Act on the Sexual Protection of Children and Adolescents establishes criminal penalties for a range of activities related to sexually exploitative material. These include the production, importation, exportation, sale, rental, distribution, provision, possession, transportation, advertisement, introduction, exhibition, or screening of such material, as well as the procurement of children or adolescents to work in the production or viewing of such material, or knowingly purchasing, possessing, or viewing it. Additionally, the article criminalizes attempted and habitual offenders of production, importation, and exportation.

2.3 Overseas CSAM Act

The most notable overseas legislation in this regard is the Digital Evidence Policy and Procedures Manual of the National Institute of Justice in the United States, which has codified detailed procedures for handling CSAM.[9] The manual initially identifies CSAM as a significant concern, ranking it alongside terrorism, murder, rape, and child abuse. If evidence contains contraband, it should be *labelled* as such, and the same procedure should be followed when disclosing the evidence to the originating agency. The data must be irretrievably destroyed by an approved erasure method and accompanied by an Acknowledgement of Contraband form duly signed by the detective in charge. Upon the removal of evidence, it is imperative that it be sealed and marked with the pertinent details, including the date, time, reason for removal, and the name and initials of the employee responsible. If reports containing contraband evidence are processed, any reports containing images of child exploitation or abuse will be *reproduced without the images*. Copies of these will then be provided to the investigator and the prosecuting attorney's office. It is imperative that the signature of the investigator or

prosecuting attorney be appended to the contraband identification form. Without a court order, reports are not permissible to be removed from the digital forensic lab. Furthermore, the defense counsel or expert is obliged to sign a receipt attesting to possession and return the report after disposition. All reports must be stored on a server or accessible storage device. Handwritten or hardcopy documents must be scanned and stored in a case folder on the server or accessible storage device.

3 Suggestion of Improvements to Processing CSAM

3.1 A legal improvement plan

Should Korean lawmakers establish specific methods and procedures for CSAM through manuals or guidelines and strengthen laws against contraband leaks by referring to foreign legislation, it is possible that CSAM evidence security could be enhanced. Based on our status analysis, detailed procedures for handling CSAM are not legally codified in South Korea, unlike in the United States. Therefore, it is necessary to detail procedures for defendants during investigations, including viewing and copying evidence. Legislating procedures for adding watermarks to confiscated items could further enhance awareness and prevent evidence leaks.

3.2 A technical improvement plan; Watermarking

It is imperative that a technical procedure be established to add watermarks in order to prevent the leakage of evidence. The current investigative process is primarily concerned with the authentication of evidence. However, there are no countermeasures in place to prevent the potential leakage of such evidence. The implementation of watermarks can assist in preventing evidence leakage by investigators or suspects while simultaneously guaranteeing the suspect's right to view and copy evidence. If a watermark is identified on leaked evidence, the source of the leak can be traced and prosecuted, thereby raising awareness of the incident. Watermarks can be incorporated at the point of taking out evidence and at the point of enrolment to identify the leakage point.

4 Conclusions

In South Korea, the management system for criminal evidence, such as CSAM, is inadequate compared to handling other types of evidence, resulting in a high risk of leakage. To address this, it is essential to implement robust domestic laws and guidelines alongside a systematic framework independent of investigators' ethics to enhance security. Further research will provide a more detailed analysis of the existing legal provisions and explore methods to add watermarks to evidence effectively.

Acknowledgement. This paper was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government(MOTIE) (RS-2024-00415520, Training Industrial Security Specialist for High-Tech Industry)

References

1. Korean National Police Agency. (2024) statistics from the National Police Agency Cyber Investigation Center, Crime incidence, arrests, and investigation results.
2. NSPCC. (2023). Evidence Review on Online Risks to Children.
3. NCMEC. (2024). Congress Holds 'Big Tech' Accountable in Monumental Child Safety Hearing.
4. Hangyun Kim et 1. (2022). The Study on Comprehensive Development Plan for Advanced Forensic Investigation Policy and Forensic Techniques (v).
5. National investigation headquarters of the Korean National Police Agency (Cyber investigation bureau). (2022). Current status of crime occurrence, arrest and investigation results.
6. Korean National Police Agency. Rules on the Processing of Digital Evidence, etc. (Police Agency Order)
7. Supreme Prosecutors' Office, Republic of Korea. Regulations on the Collection, Analysis and Management of Digital Evidence (Formula of Supreme Prosecutors' Office).
8. Jin Jang. (2021) A study on the management plan of digital evidence secured during the investigation process.
9. National institute of justice. (2020). digital evidence policies and procedures manual.

Generative AI and Child Sexual Abuse Material: Enhancing Legal Frameworks through Child Recognition Model

Yeram Lim, Ye-eun Jeon, Seongmin Kim

Donam Campus: 2, 34 da-gil, Bomun-ro, Seongbuk-gu, Seoul (02844), South Korea
Dept. Convergence Security, Sungshin Women's University

[220246056, 220246057, sm.kim]@sungshin.ac.kr

Abstract. While the rise of generative AI has brought many benefits to the lives of individuals, it has also led to an increase in the number of crimes committed by these technologies. In particular, the crime of child sexual abuse material crime is becoming more serious with the advent of generative AI, and the current legal system is ill-equipped to deal with it. This study analyses the new forms of child sexual abuse material crimes that have emerged because of the development of generative AI and the current criminal laws in place, identifies the limitations of these systems, and seeks ways to improve them.

Keywords: child sexual abuse material, generative AI, Evidence validity

1 Introduction

The advent of generative AI has bestowed numerous advantages upon humanity, yet it has also precipitated a diversification of criminal activities and the emergence of novel challenges. The exploitation of child sexual abuse material (CSAM) represents a case in point, wherein the advent of generative AI has precipitated a profound transformation in the modus operandi of CSAM crimes. This shift is evidenced by the findings of the 2024 Cryptocurrency Crime Analysis Report, published by Chainalysis, a blockchain analytics and cryptocurrency security company. Fig.1. presents a summary of the pertinent data.

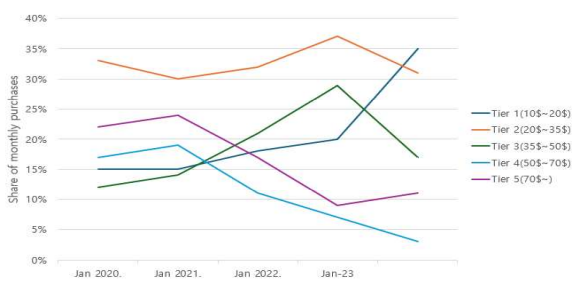


Fig. 1. CSAM purchases by severity Tier from 2020 to 2023

Chainalysis conducted an analysis of trends in CSAM purchasing crimes by categorizing CSAM into five tiers, with higher tiers corresponding to higher prices and more serious content. As shown in Fig.1., from 2021 to 2022, the volume of CSAM transactions in the higher tiers remained relatively constant, whereas the volume of CSAM transactions in the lower tiers exhibited an increase. This could be indicative of a reduction in the severity of CSAM crimes. However, the fact that the trend is not solely associated with the severity of the crime but also with the volume of transactions suggests that the advent of generative AI has led to the proliferation of mass-produced CSAMs, which has resulted in a decline in the overall transaction price.[1] This interpretation notwithstanding, it is evident that crimes utilizing generative AI to create and sell CSAMs do occur. Indeed, there have already been instances of criminal penalties being imposed in South Korea.[2][3]

As the use of generative AI increases and it becomes more straightforward to create crime-related items, such as CSAM, it is vital to establish the components of related crimes before this crime pattern gains further traction. In this paper, we put forward a proposal to establish a reasonable basis for judgment and enhance the punishment system by utilizing face recognition algorithms to respond to this novel type of CSAM crime.

2 Background

In the case of regular CSAM that is directly filmed, Article 11(3) of the Act on the Sexual Protection of Children and Adolescents provides for a sentence of up to three years imprisonment for the distribution, offering, advertising, introduction, display, or screening of CSAM. In contrast to general deepfake false images, which are punishable under Article 14(2) of the Special Act on the Punishment of Sexual Violence Crimes (Sexual Violence Punishment Act), CSAM utilizing deepfakes is subject to the provisions of the Act on the Sexual Protection of Children and Adolescents (ASPCA). [4]

There is currently no clear standard of punishment for CSAM created by generative AI. However, recent case law on actual incidents in Korea suggests that the ASPCA is applied in a manner consistent with that of deepfakes. In both cases of deepfake videos and videos produced by generative AI, the determining factor is the probability of being identifiable as a child or adolescent in accordance with the stipulations set forth in ASPCA Articles 2 and 5. This criterion is intended to enable the average person in society to recognize a child and classify it as CSAM, thereby imposing more severe punishment. However, it is necessary to consider whether such cognitive abilities are a valid basis for judgment. Consequently, we propose that AI neural networks should replace the cognitive abilities of humans that can influence judgment, thereby increasing the validity of the judgment basis.

3 Suggestion

3.1 Assumptions

In the domestic case, the keywords entered in the prompt, "naked," and "10 years old," were found to have multiple intentions in creating the CSAM, which served as evidence.[3] In the absence of secondary evidence, such as that provided by keywords, we propose the use of AI neural network-trained child recognition models. This is particularly relevant in cases where the validity of the evidence is dependent on cognitive ability. Examples of such cases include instances where there is no trace of what was typed in the prompt, or where the output was generated by bypassing semantics and using direct keywords. Furthermore, the child recognition model is based on the assumption that the evidence video is AI-generated.

3.2 Literature Review and Limitations

The majority of extant studies have concentrated on enhancing the ability to detect or recognize faces or capture their distinctive features. Moreover, numerous studies have constructed models for the identification of children and adults, respectively, and have attempted to predict their age. We note that they have focused on distinguishing between children and adults only. However, we believe that the extant literature on face recognition models offers insights that can enhance the performance of the child recognization model regarding CSAM.

Haoxiang Li and colleagues propose a convolutional neural network (CNN) that operates at multiple resolutions. The study demonstrates a model with robust face detection capabilities across a range of resolutions.[6] In the hybrid method utilizing radial basis function neural networks (P-RBF NNs), Sung-Hoon Yoo et al. illustrate the enhancement of performance and recognition rate through the elimination of superfluous images via active shape models (ASM).[7] In the hybrid approach using Gabor wavelet faces and an extended neural net feature space classifier, A. A. Fathima et al. introduce a model capable of capturing local features of faces with a high degree of accuracy. However, this approach is limited by its inability to detect faces that are not oriented towards the camera.[8]

The learning methods employed in these studies are likely to be crucial in identifying illegal video footage, which is often low-resolution and challenging to recognize. Despite these limitations, we believe that capturing local features of faces will play an important role in discriminating between adults and children.

3.3 Utilizing Child Recognition Model

The final proposed model is a combination of two models. The first model identifies linguistic characteristics (characters, subtitles) in the video to find keywords that can be inferred as CSAM. In the event that a subtitle file of the requisite format is present within the video, such as a ‘.smi’ or ‘.srt’ file, an algorithm is developed which is capable of directly analysing the file in order to identify keywords that can be recognised

as children. The second model identifies whether the person in the video is a child and builds an AI neural network to prove the intention of creating CSAM even if the user bypasses keywords. The output of the model is the probability of child sexual exploitation and the identification of CSAM when the probability is above a certain level.

There is also a need for institutional support for these models to serve as evidence. Legal and institutional support will help us to make clearer and more valid judgments through this process.

4 Conclusions

To enhance the quality of judgment, we put forth the notion of integrating AI into the decision-making process to ensure the accuracy and reliability of the judgments rendered. As the AI develops its learning system, the reliability of its results will concomitantly increase. Nevertheless, as the generated child is not a genuine individual, certain implausible elements emerge. Consequently, it is essential to determine how to address these discrepancies when making assessments. For example, the body may be that of a child, but the face may be that of an adult, or vice versa. In future research, we will develop a 'child recognition model' using child face recognition algorithms to address the issue of unrealism for the generated CSAM.

Acknowledgement. This paper was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00415520, Training Industrial Security Specialist for High-Tech Industry)

References

1. Chainalysis. (2024). 2024 crypto crime report.
2. Jong-Kwang Yoo. (2022) ‘Jeonbuk National Police Agency jails 20 for creating and distributing child sex trafficking images using “deepfake” technology’, Global Economic, https://www.g-enews.com/article/General-News/2022/02/202202101135344739fefc18799e_1
3. Ahn Jun-young. (2023) ‘40s jailed for creating child sex trafficking with AI...Korea's first case’, Busan Daily, 25 September 2023, https://n.news.naver.com/article/082/0001234004
4. Article11 of the Act on the Sexual Protection of Children and Adolescents (Production and Distribution of Sexual Exploitation of Children and Adolescents, etc.)
5. Article 14(2) of the Special Act on the Punishment of Sexual Offences (Penalties for Distribution of False Videos, etc.)
6. Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, Gang Hua. (2015). A Convolutional Neural Network Cascade for Face Detection, Computer Vision Foundation 2015.
7. Sung-Hoon Yoo, Sung-Kwun Oh, Witold Pedrycz,. (2015). Optimized face recognition algorithm using radial basis function neural networks and its practical applications, Neural Networks, Volume 69, Pages 111-125.
8. A. A. Fathima, S. Ajitha, V. Vaidehi, M. Hemalatha, R. Karthigaiveni and R. Kumar. (2015). "Hybrid approach for face recognition combining Gabor Wavelet and Linear Discriminant Analysis," IEEE International Conference on Computer Graphics, Vision and Information

Security (CGVIS), Bhubaneswar, India, 2015, pp. 220-225, doi: 10.1109/CGVIS.2015.7449925.

COPY&RETURN: Nullifying Suspicious Cryptographic Operations for Behavioral Analysis of Ransomware

Yohan Lee, Jiseok Bang, and Seungkwang Lee*

Department of Cyber Security, Dankook University, South Korea
Email: {hanja1500, kevin959, sk.cryptographic}@dankook.ac.kr

Abstract. Ransomware has emerged as a significant cybersecurity threat, encrypting user data and demanding ransom payments. Traditional static and dynamic analysis methods each have their limitations: static analysis struggles with obfuscated binaries, while dynamic analysis often necessitates virtual machine inspection and fails to provide complete coverage of the control flow. To address these shortcomings, a combination of static and dynamic analysis is employed to yield a comprehensive evaluation of unknown and suspicious binaries. This paper introduces a novel pre-processing method called *COPY & RETURN*, designed to bypass ransomware's cryptographic functions. This approach uses a two-layer neutralization technique to disable cryptographic operations, enabling a detailed analysis of ransomware behavior without risking data loss. The first layer intercepts either static or user-defined cryptographic functions through binary injection, while the second layer filters out suspicious functions in dynamic libraries using LD_PRELOAD.

Keywords: Ransomware · Dynamic analysis· Malware analysis

1 Introduction

Ransomware has become a critical threat in the cybersecurity landscape, with incidents increasing significantly in recent years. In 2023 alone, ransomware attacks surged by over 37%, with the average ransom demand exceeding $100,000 and some reaching up to $5.3 million [6]. In case of Conti ransomware and its variants, it could leverage firmware attacks to jeopardize the system by tampering the firmware incorporating various tools such as WhisperGate and HermeticWiper. Ransomware detection and prevention employ various approaches of static and dynamic analysis including machine learning techniques.

For instance, Poudyal and Dasgupta [5] proposed an AI-based hybrid approach for detecting ransomware. However, static analysis has limitations, notably its ineffectiveness against advanced threats that use obfuscation and encryption to hide malicious code. This method can miss polymorphic and metamorphic malware that changes its code to evade detection. In dynamic analysis, Redemption, developed by Kharraz and Kirda [2], is a real-time endpoint protection solution that monitors file I/O requests at the operating system level.

Recently, Kokeš *et al.* [3] proposed an automatic ransomware detection and decryption method using dynamic analysis of AES encryption. This study explains methods to trace and recover encryption keys used by ransomware in various scenarios.

However, dynamic analysis often requires performing behavioral analysis of the target binary in a virtual machine environment, which involves significant engineering effort and cost. To address this limitation, many researchers have combined static and dynamic analysis in hybrid approaches [1, 4, 7], enhancing the overall effectiveness of ransomware analysis. This paper proposes a novel hybrid approach to identify and neutralize the cryptographic functions used in unknown binaries. To achieve this, we introduce a two-layer screening method for suspicious operations that could be destructive to the filesystem. First, we redirect an arbitrary number of cryptographic operations defined in static functions to our injected code, nullifying high-entropy computations. In the second layer, we override potential cryptographic operations in dynamic libraries using LD_PRELOAD.

2 Current Work

We introduce a novel pre-processing method, aptly named *COPY & RETURN*, nullifying the cryptographic functions, thereby neutralizing their destructive behavior and preserving the original data during the dynamic analysis of ransomware. Our method consists of two types of function overrides conducted in static and dynamic time as follows. First, we intercept either static or user-defined cryptographic functions by using binary injection. Here, the use of binary injection to intercept a large number of functions in dynamic libraries is time-consuming and also makes the target binary huge. To resolve these limitations, we filter out, in the second step, a number of suspicious functions in dynamic libraries using LD_PRELOAD. To achieve this goal, we prepare in advance a shared object that defines our customized features for well-known cryptographic functions.

2.1 Overriding Static User-Defined Library

Suppose there exists a suspicious user-defined cryptographic function with the following prototype:

`encryptor` (uint8_t *C, uint8_t *P, uint8_t *K)

where C, P, and K indicate the ciphertext, the plaintext, and the key, respectively. By the System-V calling convection of x86_64, we know that C and P are stored in `rdi` and `rsi`, respectively. To nullify the cryptographic operations in *encryptor*, at the tail of the target binary, we inject just two lines of the code section as follows:

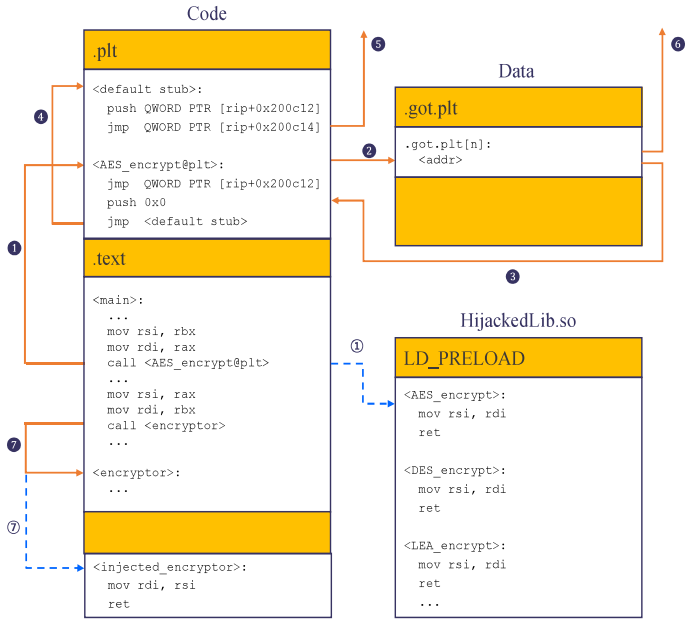


Fig. 1. Overview of the proposed attack. Solid line: benign function calls; dashed line: hijacked function calls.

```
<injected_encryptor>:
   mov rdi, rsi
   ret
```

After executing this injected code, the plaintext is just copied into the ciphertext.

2.2 Overriding Dynamic Library

In Linux ELF binaries, lazy binding resolves unknown references to functions in shared libraries using the Procedure Linkage Table (.plt) and the Global Offset Table (.got) sections. The steps ❶ - ❻ illustrated in Fig. 1 demonstrate the lazy binding process during the first call of a function named `AES_encrypt()`.

We override the target functions defined in the shared library with new definitions by using LD_PRELOAD. In this case, we can preload a shared library, named `HijackedLib.so`, incorporating the customized implementations of the target functions such as AES, DES, and LEA, as shown in Fig. 1. Let's assume that an input binary references the following cryptographic function defined in `openssl`:

`AES_encrypt`(uint8_t *P, uint8_t *C, uint8_t *K).

Unlike the previous case of `encryptor`, P and C have the reverse order. Then, we can neutralize the behavior of `AES_encrypt` by overriding it with its new definition in `HijackedLib.so` as follows:

```
<AES_encrypt>:
   mov rsi, rdi
   ret
```

By doing so, we are enable to override an arbitrary number of functions using LD_PRELOAD at once. Depending on the specific prototype of the target function, we can make the destructive operations in suspicious binaries useless by simply copying the source to the destination. If the injected code consistently bypasses cryptographic operations, the victim's file will remain intact without corruption while we can still monitor the behavior of the suspicious binaries.

3 Conclusion & Upcomming Challenges

This paper presented *COPY & RETURN*, capable of defeating cryptographic operations in malicious and unknown binaries. Our proposed method can override well-known cryptographic functions and user-defined functions using either binary or LD_PRELOAD, consistently copying plaintexts into the corresponding ciphertexts. This enables non-destructive analysis of suspicious ransomware without the risk of encrypting the data in the filesystem. By running the target binary after pre-processing, we can obtain the details of read-write-delete patterns of ransomware.

Our upcomming challenge is to conduct the additional experiments on real ransomware samples consisting of user-defined functions and shared library. In addition, we will provide a novel method to bypass several cryptographic instructions such as AES-NI.

References

1. Ding, C., Luktarhan, N., Lu, B., Zhang, W.: A hybrid analysis-based approach to android malware family classification. Entropy **23**(8), 1009 (2021)
2. Kharraz, A., Kirda, E.: Redemption: Real-time protection against ransomware at end-hosts. In: Research in Attacks, Intrusions, and Defenses: 20th International Symposium, RAID 2017, Atlanta, GA, USA, September 18–20, 2017, Proceedings. pp. 98–119. Springer (2017)
3. Kokeš, J., Matějka, J., Lórencz, R.: Automatic detection and decryption of aes using dynamic analysis. SN Computer Science **3**(5), 338 (2022)
4. de Oliveira, A.S., Sassi, R.J.: Chimera: an android malware detection method based on multimodal deep learning and hybrid analysis. Authorea Preprints (2023)
5. Poudyal, S., Dasgupta, D.: Analysis of crypto-ransomware using ml-based multi-level profiling. Ieee Access **9**, 122532–122547 (2021)

6. Verizon: 2022 data breach investigations report (2022), https://www.verizon.com/business/resources/T62d/reports/dbir/2022-data-breach-investigations-report-dbir.pdf
7. Yunus, Y.K.B.M., Ngah, S.B.: Review of hybrid analysis technique for malware detection. In: IOP conference series: materials science and engineering. vol. 769, p. 012075. IOP Publishing (2020)

Interoperable Alternative to Canary-based Protection of Stack Smashing

Kangmin Kim[1], Jeong-Nyeo Kim[2], and Seungkwang Lee*[1]

[1] Department of Cyber Security, Dankook University, South Korea
[2] Cyber Security Research Division, ETRI, South Korea
Email: {km.korean, sk.cryptographic}@dankook.ac.kr[1]
jnkim@etri.re.kr[2]

Abstract. This paper introduces a novel software-based approach to enhancing stack smashing protection in C/C++ applications, specifically targeting Return-Oriented Programming attacks. Traditional canary-based protections are vulnerable to brute-force and format string attacks. Additionally, many stack protection mechanisms require access to the source code or recompilation, complicating the security of existing binaries. In this paper, we propose a new method, aptly named *StackGuard*$^+$, that modifies the canary-based protection mechanism by altering the code responsible for canary insertion and verification. This change ensures the integrity of the return address while maintaining the original code size, allowing for seamless interoperability without the need for recompilation or additional hardware.

Keywords: Buffer Overflow · Stack Smashing · Stack Protection.

1 Introduction

Stack buffer overflow, sometimes referred to as stack smashing, is a type of software vulnerability that occurs when a program writes more data into a stack-based buffer than it can hold, causing the excess data to overflow into adjacent memory locations. This can lead to unintended consequences, including the potential for an attacker to execute malicious code or manipulate the program's behavior. C/C++ applications are particularly vulnerable to Return-Oriented Programming (ROP) attacks due to their lack of memory protection and reliance on writable memory for implementing call-return abstractions. When data is stored beyond the buffer's capacity, as illustrated in Fig. 1, the stack structure allows an attacker to easily corrupt the return address, facilitating code reuse for ROP attacks.

StackGuard [1] is a widely recognized software-based countermeasure that modifies the C compiler to insert a random value, known as a *canary*, between ① and ② in Fig. 1. By verifying if this canary value has been altered, StackGuard can effectively detect buffer overflow attacks.

However, stack canaries do not guarantee perfect security due to several vulnerabilities. First, format string vulnerabilities [2] can cause memory leakage,

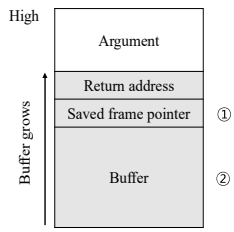


Fig. 1: General stack structure vulnerable to buffer overflow.

allowing attackers to extract all types of canaries, except possibly the random XOR type. Second, brute-force attacks [3] can be feasible because of the canaries' short lengths (i.e., three or seven bytes). For these reasons, canary-based protection can be bypassed if the attacker overwrites the exact value of the canary in the correct location during the stack smashing. To prevent these vulnerabilities, several countermeasures such as Return Address Defender [4], Shadow stack [5] and LsStk [6] have been proposed. However, the main disadvantages of these method are their difficulty in achieving interoperability and additional run-time overhead. In particular, recompilation is not a practical solution without the source code.

2 Current work

2.1 Key idea behind

In this section, we propose a *StackGuard*$^+$ for securely storing and verifying return address by replacing the existing machine code. Because our replacement occupies exactly the same size of machine code, this can be adopted as an alternative to the canary-based countermeasures. Moreover, this involves binary-level modifications that eliminate the need for recompilation, and does not require any modifications to libraries or the use of separate modules.

As shown in Fig. 2, *StackGuard*$^+$ can be implemented on x86-64 platform by modifying just only three lines of the assembly code as follows:

- (`A` → `A'`): the return address at `[rbp+0x08]` is moved into `rax`.
- (`B` → `B'`): the return address is now saved at `fs:0x28`.
- (`C` → `C'`): before returning the function, the return address at `[rbp+0x08]` is copied to `rax`.
- `D`: the current value of the return address is compared to the saved return address, which is obtained from `fs:0x28`.

On the x86 platform, while the methodology is the same as that applied to x86-64, differences in registers and memory structure require replacing the machine code depicted in Fig. 3.

In case that the buffer overflow disturbs the return address, it calls `__stack_chk_fail@PLT`, enforcing the process to terminate.

```
64 48 8b 04 25 28 00 00 00  mov   rax, QWORD PTR fs:0x28     ;A
48 89 45 f8                 mov   QWORD PTR [rbp-0x08], rax  ;B
48 8b 45 f8                 mov   rax, QWORD PTR [rbp-0x08]  ;C
64 48 2b 04 25 28 00 00 00  sub   rax, QWORD PTR fs:0x28     ;D
```

(a) Before modification

```
48 8b 45 08                 mov   rax, QWORD PTR [rbp+0x08]  ;A'
64 48 89 04 25 28 00 00 00  mov   QWORD PTR fs:0x28, rax     ;B'
48 8b 45 08                 mov   rax, QWORD PTR [rbp+0x08]  ;C'
64 48 2b 04 25 28 00 00 00  sub   rax, QWORD PTR fs:0x28     ;D
```

(b) After modification

Fig. 2: Byte replacement on the x86-64 platform.

```
65 a1 14 00 00 00  ;a
89 45 f4           ;b
8b 45 f4           ;c
65 2b 05 14 00 00 00  ;d
```

```
8b 45 04           ;a'
65 a3 14 00 00 00  ;b'
8b 45 04           ;c'
65 2b 05 14 00 00 00  ;d
```

Fig. 3: Byte replacement on the x86 platform (The right box illustrate the machine code after modification).

2.2 Scalability to ARM64

In ARM64 assembly code using StackGuard, unlike x86 and x86-64, a random canary value from `__stack_chk_guard` is copied to the `x1` register and then inserted into the stack at `sp:0x28` during the insertion process. To detect stack smashing attacks, the canary values stored in `__stack_chk_guard` and `sp:0x28` are compared.

To implement *StackGuard*$^+$ on ARM64, instead of inserting the canary into `sp:0x28`, the value of the link register `x30`, which stores the return address, must be inserted into `__stack_chk_guard`. By copying the return address located at `sp:0x30` into `__stack_chk_guard`, the integrity of the return address can be verified by comparing the value stored in `sp:0x30` with that in `__stack_chk_guard`.

2.3 Compilation Optimization Issues

The GCC optimization level introduces slight variations in the code area for canary insertion and verification. At higher optimization levels such as `-O1`, `-O2`, `-O3`, and `-Os`, this address is calculated by adding an offset to `rsp`. Additionally, at optimization levels above `-O1`, the canary retrieved from `fs:0x28` might be stored in `rsi` instead of `rax` during the insertion steps. Importantly, these changes do not affect the machine code size. Even in these optimized codes, the return address can always be found by adding `0x10` to the address, where the canary was originally inserted. For these reasons, the proposed scheme can be

adapted to the optimized binary code with a simple adjustment by copying the return address to `fs:0x28`.

3 Conclusion & Upcoming Challenges

In this study, we introduce *StackGuard*$^+$ which improves on the existing canary-based protection without the need for either compiler modifications or recompilation. To be specific, it replaces only 26 bytes of the existing binary code pertaining to canaries with our code.

Our future work includes the extension of *StackGuard*$^+$ to other platforms beyond x86, x86-64 and ARM64. To that end, various compiler-dependent machine codes must be analyzed to investigate their unique way of inserting and verifying canary values. Moreover, since *StackGuard*$^+$ no longer inserts the canary value, the designated canary location is filled with either a garbage value or 1, depending on the execution environment. For this reason, overwriting the saved frame pointer with x - 8, the instruction pointer is expected to point the malicious code if both the callee's and caller's epilogues are executed sequentially, where x is the address of the attacker's shellcode. Although this ROP attack is limited to only scenarios where there is no access to the caller's stack frame between the callee's and caller's epilogues, this weakness must be also addressed.

References

1. Cowan, C., Pu, C., Maier, D., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., Zhang, Q., Hinton, H.: Stackguard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. In: USENIX Security Symposium, vol. 98, pp. 63–78. San Antonio, TX (1998)
2. Newsham, T.: Format string attacks. (2000)
3. Marco-Gisbert, H., Ripoll, I.: Preventing brute force attacks against stack canary protection on networking servers. In: 2013 IEEE 12th International Symposium on Network Computing and Applications, pp. 243–250. IEEE (2013)
4. Chiueh, T.-c., Hsu, F.-H.: RAD: A Compile-Time Solution to Buffer Overflow Attacks. In: Proceedings 21st International Conference on Distributed Computing Systems, pp. 409–417. IEEE (2001)
5. Burow, N., Zhang, X., Payer, M.: SoK: Shining Light on Shadow Stacks. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 985–999. IEEE (2019)
6. Ahsan, M., Ali, M.: LsStk: Lightweight Solution to Preventing Stack from Buffer Overflow Vulnerability. In: 2023 17th International Conference on Open Source Systems and Technologies (ICOSST), pp. 1–7. IEEE (2023)

Early Detection of Vulnerabilities in Static Analysis with Terraform and Kubernetes: Methods and Strategies

Hyeyeon Hyeon[1], Gahyun Kim[1], Mijin Kim[1], Yerin Chun[1] and Seongmin Kim[1]

[1] Dept. of Convergence Security Engineering, Sungshin Women's University, 2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul, Republic of Korea
{20221143, 20211039, 20221081, 20211103, sm.kim}@sungshin.ac.kr

Abstract. The paradigm shift towards DevSecOps has highlighted the importance of managing infrastructure through Infrastructure as Code (IaC). Although Terraform, a prominent IaC tool, is frequently used in conjunction with Kubernetes, challenges arise regarding its functional compatibility and interoperability with Kubernetes in cloud-native environments. Existing security solutions and static analysis tools, such as tfsec, primarily focus on detecting vulnerabilities in infrastructure-related code. This study aims to address these challenges by developing methods to detect and patch misconfigurations in both the cloud and cluster layers through static analysis of IaC. By implementing these methods, we seek to enhance security within the DevSecOps cycle.

Keywords: Terraform, Kubernetes, misconfiguration.

1 Introduction

Recent innovations in cloud-native environments have accelerated the need for integrating security throughout the entire software life cycle, resulting in a paradigm shift from conventional DevOps to DevSecOps. Notably, Infrastructure as Code (IaC) has gained popularity as it enables automated management of infrastructure through code. One of the representative solutions is Terraform[2], known for its effective collaboration with Kubernetes, an open-source platform that automates the deployment, scaling, and management of containerized applications. Terraform and Kubernetes are widely used as core tools for cloud infrastructure and cluster management in cloud-native environments. One can use Terraform to automate the provision of Kubernetes clusters and deploy applications to the cluster. However, security monitoring and auditing for IaC when utilizing it with cloud-native environments lag behind its functional compatibility and interoperability with Kubernetes.

Existing security solutions for inspecting threats in IaC have primarily focused on detecting vulnerabilities in infrastructure-related code rather than the cluster layer. Basically, Static analysis tools, such as tfsec[4], are commonly used by DevSecOps teams to identify and prevent security vulnerabilities in Terraform code. However, these tools may not fully detect misconfigurations at the cluster layer because IaC is inherently designed for declaring infrastructure rather than cluster configurations. For example,

we demonstrate that it cannot cover some security vulnerabilities in Kubernetes configuration files, such as YAML and JSON. In fact, vulnerabilities or security incidents due to Kubernetes misconfigurations occur frequently. At Tesla, a data leak occurred due to a cryptojacking attack that caused general users to mine Bitcoin at will due to Kubernetes misconfiguration. The attack was discovered while Red Rock inspected the server for incorrect configuration and stability. This corresponds to an attack caused by misconfiguration without creating a password to protect the terminal on Kubernetes [5]. As such, misconfiguration can cause potential security issues in the configuration connection between the cloud and cluster layers.

In this study, we aim to analyze potential security issues that may arise in linking the cloud and cluster layer settings in the cloud-native environments and fill the gap between them from a DevSecOps perspective. In particular, we identify the cases of Kubernetes configuration errors that legacy IaC solutions cannot detect and propose ways to improve them. Through this, we seek to find a way to effectively detect configuration errors in the cloud and cluster layers through static analysis in IaC. In this study, we aim to strengthen security within the DevSecOps cycle by comparing and analyzing code written in Terraform and actual Kubernetes configuration files to derive methods to detect and patch vulnerabilities in advance at the static analysis stage.

2 Terraform-Kubernetes Misconfiguration

In this section, we present a case study to answer the following questions: 1) Is it possible to instantiate a pod with misconfigurations written in Terraform? and 2) can these misconfigurations be detected through static analysis? For this, we write a customized Terraform code related to the permission setting that might incur privilege escalation.

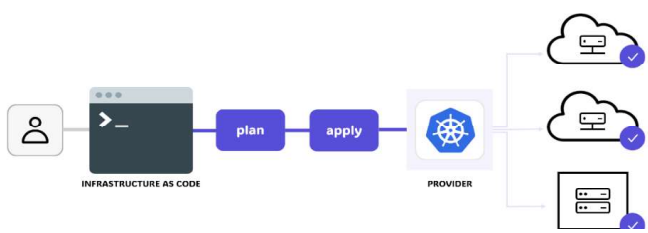


Fig.1. Terraform Kubernetes connection environment configuration

2.1 Research environment settings

We present an evaluation environment to analyze whether configuration errors related to permission settings can be detected through static analysis. For quick verification, we utilize Minikube[3] instead of Kubernetes. An environment linking Terraform and Kubernetes was built on an Ubuntu 22.04 LTS virtual server using VMware Workstation 17 Player. Fig.1. illustrates the overview of our evaluation environment. After installing Docker, Minikube, and Terraform runtimes on the virtual server, we

established an environment for integrating Terraform with Minikube to build an infrastructure implemented as .tf code.

2.2 Main.tf code

To check whether it is possible to instantiate a pod with misconfigurations, we first implement a simple main.tf code that might incur exploits. As shown in Fig.2., The main.tf file defines a Kubernetes resource that assigns a particular role to a service account within a specific namespace. Although no error occurred when applying the IaC via Terraform, there may be a potential security issue due to excessive permission settings and permissions granted to all pods. The following subsection shows the code that analyzed and corrected the misconfiguration present in the main.tf file line 25, 26.



Fig 2 main.tf code and misconfiguration

2.3 Misconfiguration Analyze

Due to the misconfiguration that occurred in 'verbs = ["get", "list", "create"]' (line 26 in Fig.2.), the code allows the service account to have all "get", "list", and "create" permissions for the pod resource. In addition to read permission, it gains write permission with the "create" specification. Under "First set excessive permissions," the code should be modified to grant only "get" permission by applying the principle of least privilege. Since it is a simple service account, the "create" write permission should not be allowed, and it is more effective in preventing potential security issues from granting only the minimum necessary read permission.

Second, a misconfiguration occurred due to the code 'resources = ["pods"]' (line 25 in Fig.2.), which granted the permissions in the verbs code to all pods in the cluster. In Figure 2, under "Second set excessive permissions", 'resources_names' was added to grant permissions only to specific pods, and the code was modified so that only the pod called "spod" would be granted permission. This prevents potential security issues by avoiding granting permissions to a wide range of pods without setting permissions for specific pods or conditions.

3 Conclusion and Recommendations

Considering that even a simple code like main.tf demonstrated in this study contained a misconfiguration that was launched without error in Terraform, misconfigurations in IaC during the development stage are expected to be launched and deployed without alerts in a further DevOps cycle. This can act as a vulnerability and cause greater damage. For example, a related vulnerability, "CVE-2020-8554: Man in the middle using LoadBalancer or ExternalIPs"[1], allows a potential attacker to intercept traffic from other pods (or nodes) in the cluster if they already have access to create or edit services and pods. In the main.tf code, the "create" permission granted to the service account could cause a security issue due to this vulnerability.

In addition, "Misconfigurations in Google Kubernetes Engine (GKE) Lead to a Privilege Escalation Exploit Chain"[6] is an attack that combines a FluentBit default configuration vulnerability with excessive permissions in an Anthos Service Mesh (ASM) DaemonSet. The attacker can gain cluster administrator privileges and complete control over the cluster, including creating or controlling new pods. The excessive permissions in the main.tf code or the vulnerability granting permissions to all pods can be combined with other vulnerabilities to create a larger security issue.

To prevent such potential security issues, it is important to discover misconfigurations when IaC code is launched through static analysis. In particular, security issues like the ones mentioned above may occur if source data is not automatically patched or notified to users when transferred to infrastructure tools such as Terraform. Therefore, it is necessary to implement automated functions to discover misconfigurations in IaC during the static analysis phase and automatically patch or notify users. We confirmed that misconfigurations can be detected during the static analysis phase, and an IaC vulnerability detection tool that utilizes this should be developed apart from Kubernetes.

Acknowledgement. This work is partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00351898), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Github, CVE-2020-8554: Man in the middle using LoadBalancer or ExternallPs #97076, https://github.com/kubernetes/kubernetes/issues/97076, 2024/07/25
2. HashiCorp Developer, https://developer.hashicorp.com/terraform/intro, 2024/07/19
3. Minikube, https://minikube.sigs.k8s.io/docs/, 2024/07/18
4. tfsec, https://aquasecurity.github.io/tfsec/v1.28.1/, 2024/07/21
5. UNIT 42, F.: Cloud Threat Report, Volume 7, 3000 Tannery Way, Santa Clara, CA 95054, USA (2023)
6. UNIT 32, F.: Dual Privilege Escalation Chain: Exploiting Monitoring and Service Mesh Configurations and Privileges in GKE to Gain Unauthorized Access in Kubernetes. Palo Alto Networks, 3000 Tannery Way, Santa Clara, CA 95054, USA (2023)

Exploiting Standalone Access Vulnerabilities in Envoy Proxy for Restart Attacks

Jiwon Yoo, Seo-yeon Kang, Seongmin Kim

Donam Campus: 2, 34 da-gil, Bomun-ro, Seongbuk-gu, Seoul (02844), South Korea
Dept. Convergence Security, Sungshin Women's University

[220246055, 20211036, sm.kim]@sungshin.ac.kr

Abstract. Inspired by the recent innovations in cloud-native environments, researchers have explored how to improve the system performance and latency service-level objectives (SLOs) by utilizing service mesh within the containerized environments. However, studies addressing the potential attack surfaces and vectors introduced by implementing a service mesh are under-explored. This study aims to identify one of the attack vectors for DoS attacks leveraging the sidecar proxy in Istio and presents the results of the preliminary demonstration.

Keywords: Istio, Envoy Proxy, Sidecar Proxy, Dos Attack, Cloud

1 Introduction

A cloud-native environment refers to everything from development to deployment and operation taking place on a virtualized network with the granularity of a container. Cloud-native applications divide the functions of existing applications into smaller microservices so that each module communicates with each other and operates like a single application [1]. For efficient mutual communication between microservice components, an infrastructure that can handle issues such as service discovery, service routing, failure recovery, load balancing, and security was required, and service mesh technologies have been developed to achieve this purpose.

Service meshes provide the functionalities include service discovery, routing, failure recovery, load balancing, security, and microservices authentication. Specifically, It provides various functionalities to efficiently handle microservices running on container orchestration platforms, such as Kubernetes. These capabilities help manage the increased communication surface between services and address the complexities within cloud environments and containers [2].

Typically, a service mesh deploys sidecar proxies for each service, redirecting network requests to manage traffic and authentication. This structural characteristic allows for the potential exploitation of the proxy to generate excessive requests that could lead to resource exhaustion attacks or other types of Denial-of-Service (DoS) attacks against individual services [3]. Resource exhaustion attacks can be executed through various attack vectors, such as generating excessive traffic to deplete bandwidth or exceeding the system's capacity. DoS attacks can adversely affect both

service users and providers, making it crucial to identify threats within and outside the service mesh environment for stable system operation.

This paper aims to identify a novel DoS attack vector from the attack surface targeting the sidecars in the Istio service mesh and demonstrate the potential threats with test scenarios. While systematic improvement or measurement studies related to sidecars have been actively conducted, exploration of the security of the intrinsic operational structure of sidecars is still insufficient. Our preliminary evaluation shows that. When conducting an attack targeting the sidecar proxy, the sidecar restarted to handle errors, leading to service denial.

2 Background

2.1 Istio Service Mesh

Istio is an open-source service mesh that graduated from the CNCF (Cloud Native Computing Foundation). Separate from existing microservices, Istio deploys and operates a sidecar container consisting of an Envoy proxy next to the microservice [4]. The proxy redirects network requests for services according to the container's internal configuration file. For example, assuming that services A and B are deployed on Kubernetes and the Envoy proxy is running as a sidecar using Istio, the order of sending packets from service A to service B is as shown in Fig. 1.

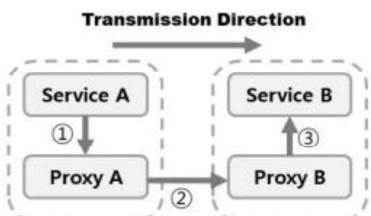


Fig. 1. Data Transmission Path Using Sidecar proxy

2.2 Istio Sidecar Vulnerability Exploitation

Previous studies have attempted to identify authentication vulnerabilities in service-oriented architecture (SOA) and microservice architecture (MSA) environments and strengthen the security relying on sidecars. Such an approach is an effort to enhance the authentication system by performing authentication for each service within the sidecar [5]. All requests from the distributed system are redirected and delivered by the sidecar, allowing the sidecar to inspect and manage authentication independently of the service. However, potential threats on sidecars themselves are under-explored. Once the service mesh layer is integrated with the legacy containerized distributed systems, all requests are indiscriminately sent to the sidecar before being forwarded to the actual service, thereby expanding the attack surface.

3 Preliminary Evaluation

3.1 Scenario and Test Environment

For evaluation, we use Istio 1.22.1 in a Kubernetes 1.29.4 environment consisting of a cluster using three virtual machines with Ubuntu 20.04. As a sample application, we use Bookinfo, a benchmark provided by Istio [6]. We assumed a scenario in which an attacker could directly access the sidecar proxy and the internal services of a service deployed in Istio by gaining access to the container and stealing permissions, as illustrated in Fig. 2.

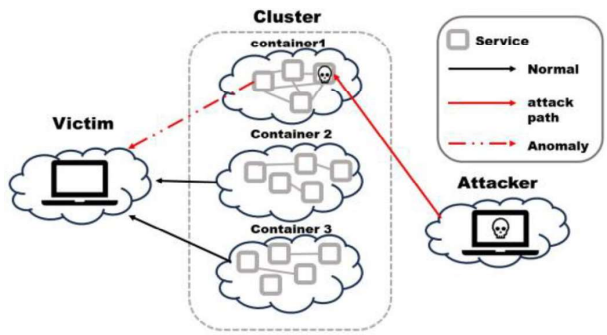


Fig. 2. Assumed Scenario: Attacker Access to the Inside of the Container

Taking advantage of the fact that the sidecar proxy can be accessed exclusively within the actual built environment, we conducted a test by generating a load from the proxy of the ProductPage service of the Bookinfo application to the proxy. Here, we assume a malicious insider could access an internal Kubernetes control plane with authorization. The process for internal attacks is shown in Fig. 3.

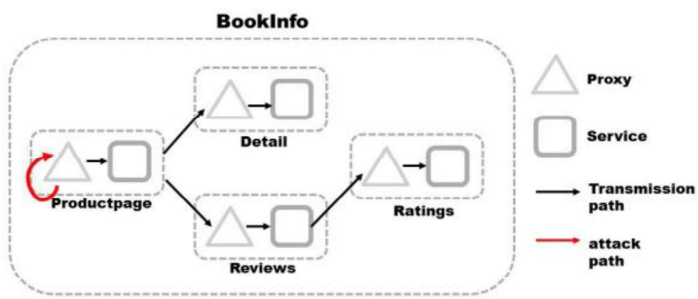


Fig. 3. Internal Attack Process

3.2 Scenario and Test Environment

After conducting the attack nine times and subsequently checking the detailed information of the pod, we observed that the proxy exhibited the same increased restart count, as shown in Fig.4. The reason for the restart was marked as OOMKilled, indicating that the pod's container was terminated due to an 'Out of Memory' error. This implies that the container was forcibly terminated due to memory insufficiency due to the attack.

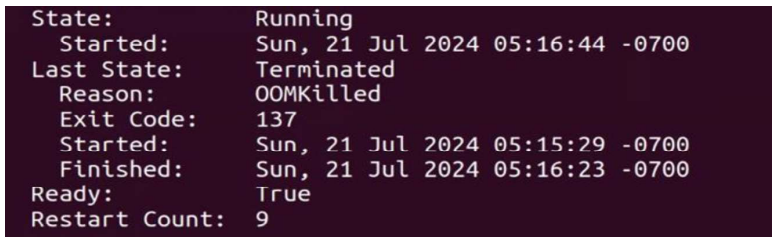


Fig. 4. Pod Internal Event Information

After successfully launching a DoS attack, we also check the log messages generated by the sidecar proxy. As shown in Fig.5., the log message contains 'Starting Istio SDS (Secret Discovery Service) Server,' which states that the proxy restarts, reinitializes its credentials, reconnects, and handles errors. This test ultimately demonstrated that when an attacker breaches the internal system, there is a potential channel for conducting DoS attacks not only on the service itself but also by targeting the proxy. Similar to the experimental applications like Bookinfo, which we demonstrated in this study, we plan to build and test the attack against various microservice workloads. We also intend to evaluate the DoS attack resilience of Istio in both internal and externally accessible environments.

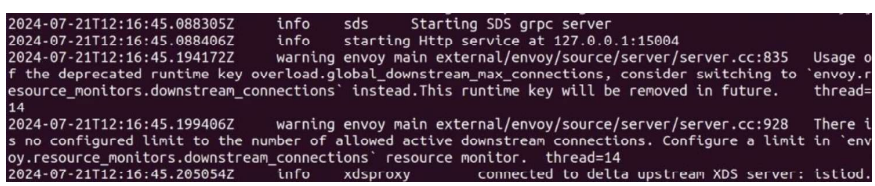


Fig. 5. Logs of the Proxy

Acknowledgments. This work is partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00351898), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. AWS Homepage, https://aws.amazon.com/ko/what-is/cloud-native/, last accessed 2024/7/22
2. Y.Y. Shin, J.S. Shin, C.H. Park, J.G. Park.: eBPF Technology Trends for Networking and Security in Cloud-native. Network and System Security Research Lab 2022, ETRI, vol. 5, pp. 68. ETRI, Daejeon, South Korea (2022)
3. Zhan, M., Li, Y., Yang, H., Yu, G., Li, B., & Wang, W. Coda: Runtime detection of application-layer cpu-exhaustion dos attacks in containers. IEEE Transactions on Services Computing, 16(3), pp.1686-1697. IEEE (2022)
4. Istio Homepage, https://istio.io/latest/docs/ops/deployment/architecture/, last accessed 2024/6/9
5. C. Meadows, S. Hounsinou, T. Wood, G. Bloom.: Sidecar-based Path-aware Security for Microservices. 2024, pp. 158–161. George Washington University, University of Colorado USA, (2024)
6. Istio Homepage, https://istio.io/latest/docs/examples/bookinfo/, last accessed 2024/07/10

Interference Avoidance and Mitigation Technique Utilizing Cooperative Characterization of Interferers in Dense Wireless Networks

Jin-Min Lee[1] [0000-0002-4337-1771], Hye-Yeon Shim[1] [0009-0009-4978-2553] and Il-Gu Lee[1] [0000-0002-5777-4029]

[1] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
csewa56579@gmail.com, 220237062@sungshin.ac.kr, iglee@sungshin.ac.kr

Abstract. Internet of Things (IoT) devices used in various fields, such as medicine, factories, use wireless communication technology, resulting in performance degradation and service availability problems owing to interference. Conventional interference mitigation studies utilize trial-and-error methods to avoid interference; however, this approach is inefficient from a performance and security perspective. To solve these problems, in this study, stations(STAs) and access points (APs) affected by interference cooperate to learn the location, power, and pattern of interference sources and minimize the impact of interference. Experimental results show that the proposed technique, which adjusts the TX power of neighboring nodes when the TX power of the interference source was 15 dBm, improves throughput by at least 195% compared to the conventional technique. In addition, the proposed technique can reduce latency by up to 0.52 s compared to the conventional method.
Keywords: Interference Avoidance, Interference Mitigation, Interference Identify, Channel Hopping, Internet of Things, Collaborative Node.

1 Introduction

With the evolution of big data and artificial intelligence technologies, sensor-based Internet of Things (IoT) and embedded devices are becoming integral to all industries [1]. These devices utilize wireless communication technologies such as Bluetooth Low Energy (BLE), Wi-Fi to transmit data collected by sensors for high-speed, real-time services wirelessly. However, the radio frequencies used by wireless communication devices are highly congested and can be subject to interference from data transmission between wireless devices [2]. Interference causes increased data-packet loss and retransmission, resulting in degraded performance and service availability for resource-constrained IoT devices. Previous studies on mitigating such interference have focused on trial-and-error methods to avoid interference [3]. However, this method suffers from the problem that if the interference source's channel changes or another interference occurs on the moved channel, it will be affected by the interference again [4]. Thus, this study proposes interference avoidance and mitigation techniques that estimate the

location, power, and pattern of the interference source through the cooperation of the station (STA) and access point (AP) affected by the interference. This changed the communication channel, TX power, and AP to mitigate the interference impact. The proposed techniques are based on the information of the identified interference source to improve the throughput and latency of the nodes around the interference source.

2 Related Work

Dong et al. [5] reviewed coexistence and interference mitigation schemes for various networks, such as Wi-Fi, wireless personal area networks (WPAN), Bluetooth. However, they did not propose avoidance and mitigation measures based on the characteristics of the interference source, nor did they consider the characteristics of lightweight devices such as IoT. Webber et al. [6] proposed a technique to classify multiple interfering signals by predicting the received signal strength (RSS) to ensure the high reliability of factory communication links. However, they did not experimentally demonstrate throughput improvement, and it is difficult to determine the extent to which the communication performance can be improved.

Conventional trial-and-error-based avoidance techniques, such as random channel hopping, can be quickly neutralized because they do not consider the characteristics of the interference source. Moreover, most studies have not considered resource-limited environments such as lightweight IoT. Therefore, to efficiently mitigate interference in resource-constrained IoT environments, it is necessary to conduct research that accurately identifies the interference source and cooperates with the nodes to avoid and mitigate interference.

3 Cooperative Interference Characterization and Avoidance Technique

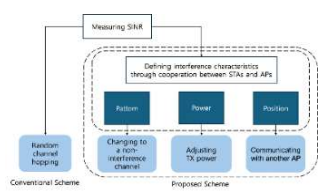


Fig. 1. Structural diagram of interference avoidance and mitigation technique.

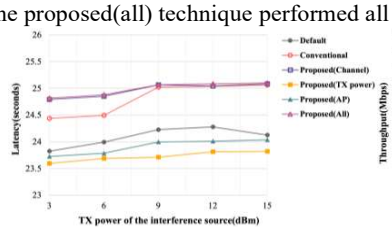
Fig. 1 shows the process of detecting interference by measuring the signal-to-interference-plus-noise ratio (SINR) and mitigating interference using conventional random channel hopping and the three proposed techniques. First, the STAs and APs measure the SINR to recognize the interference situation. In this study, a clean channel with constant SINR was assumed to be the basic environment. When the SINR degrades to a certain level, interference is considered to have occurred. In many conventional environments, once the SINR is measured and interference is detected, random channel

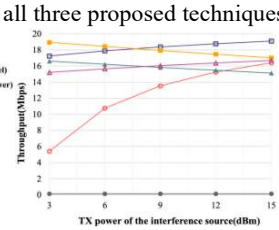
hopping is performed to change the communication channels randomly. This is inefficient because the changed channel may interfere with or move to a channel with interference. However, in the proposed technique, STAs and APs in the interference range cooperate to characterize the interference's power, location, and pattern, as well as to avoid and mitigate interference. Once the interference pattern is known, neighboring nodes can switch to the best-quality communication channel. Knowing the power of the interference, APs can reduce the impact of the interference by transmitting signals with more power than the transmitted signal of the interference source. Finally, knowing the location of the interference source, the STA can avoid interference by choosing an AP that is less affected by the interference or one closer to the AP with which it is currently communicating.

4 Performance Evaluation

For the network, it was assumed that three APs and six STAs, including interfering STAs, were arranged in a room with a width and length of 60m. The interfering STAs are fixed at (30, 30), but all the other nodes move randomly. The simulation environment was implemented using MATLAB, and all nodes communicated based on the IEEE 802.11ax. The average Root Mean Square Error (RMSE) for the estimated location of the interference source derived through trilateration was 0.0718m, which was confirmed to have a very high localization accuracy. The default means no response technique is performed to avoid or mitigate interference, even when interference occurs. The proposed(channel) is a technique that changes the channel to a channel without interference. In the proposed(TX power), the STA adjusts and mitigates the TX power more significantly than the TX power of the interference source. In addition, the proposed(AP) changes the communication target AP to an AP that is less affected by interference from the STA or an AP closer to the AP that is currently communicating. Finally, the proposed(all) technique performed all three proposed techniques.

(a)





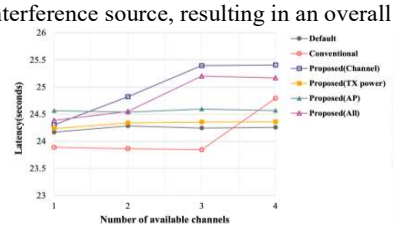
(b)

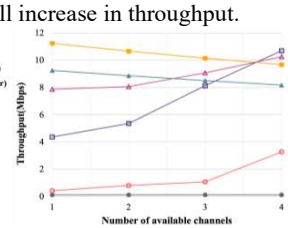
Fig. 2. Latency(a) and Throughput(b) according to increased interference source TX power when four channels were available.

Fig. 2 shows the latency and Throughput according to the increased interference source TX power when four channels were available. In (a), The latencies of proposed(TX power) and proposed(AP) techniques were measured to be at least 0.23 s, at most 0.52 s, at least 0.09 s, and at most 0.27 s lower than the default, respectively. This is because the proposed technique does not cause a packet retransmission process by interferen

ce. And in (b), the technique with the highest throughput on average is the proposed(c hannel), which improves the throughput by 11.82Mbps compared to the conventional methods, especially when the TX power of the interference source is 3 dBm. The proposed(channel), proposed(all), and conventional methods that use channels other th an those used by the interference source were not affected by the TX power in-crease of the interference source, resulting in an overall increase in throughput.

(a)





(b)

Fig. 3. Latency(a) and Throughput(b) according to the number of available channels when the TX power was 15 dBm.

Fig. 3 shows the latency and Throughput according to the number of available channels when the TX power was 15 dBm. In (a), Conventional, proposed(channel), and proposed(all) techniques have the process of changing channels; therefore, as the number of available channels increases, latency increases. In addition, the technique with the highest latency on average is proposed(Channel), which increases by at least 0.42 s and up to 1.55 s compared to conventional avoidance of channels randomly. And in (b), the default technique that does not avoid or mitigate interference generally converges to zero throughput. In (b), the technique with the highest throughput on average is proposed(TX power), which improves throughput by at least 195% and up to 2606%, respectively, compared with conventional techniques.

5 Conclusion

IoT and embedded devices utilize wireless communication technology, which causes interference in the communication process. In conventional research on mitigating interference, research on avoiding interference has mainly been conducted using a trial-and-error method, but is inefficient because another interference may occur on a moved channel, or the channel used by the interference source may be changed. Thus, in this study, the STA and AP affected by the interference work together to confirm the location, power, and pattern of the interference source. A technique for avoiding and mitigating interference by changing the channel, transmission power, and AP is proposed. Experimental results show that the proposed technique, which adjusts the TX power of neighboring nodes when the TX power of the interference source was 15 dBm, improves throughput by at least 195% compared to the conventional technique. In addition, the proposed technique can reduce latency by up to 0.52 s compared to the conventional method. Future research will focus on lightweight the proposed techniques to reduce latency and apply machine learning to environments with multiple interference sources to characterize and optimize the interference.

Acknowledgments. This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE) under the Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning and Evaluation (IITP).

References

1. Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., Mustaqim, M.: Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. IEEE Access 8, 23022–23040 (2020)
2. Eltholth, A. A.: Improved spectrum coexistence in 2.4 GHz ism band using optimized chaotic frequency hopping for wi-fi and Bluetooth signals. Sensors **23**(11), 5183 (2023)
3. Pirayesh, H., Zeng, H.: Jamming attacks and anti-jamming strategies in wireless networks: a comprehensive survey. IEEE communications surveys & tutorials **24**(2), 767–809 (2022)
4. Liu, Y., Zeng, Q., Zhao, Y., Wu, K., Hao, Y.: Novel channel-hopping pattern-based wireless IoT networks in smart cities for reducing multi-access interference and jamming attacks. EURASIP Journal on Wireless Communications and Networking 2021(1), 152 (2021)
5. Chen, D., Zhuang, Y., Huai, J., Sun, X., Yang, X., Javed, M. A., Brown, J., Sheng, Z., Thompson, J.: Coexistence and interference mitigation for WPANs and WLANs from traditional approaches to deep learning: a review. IEEE Sensors Journal **21**(22), 25561–25589 (2021)
6. Webber, J. L., Yano, K., Suga, N., Hou, Y., Nii, E., Higashimori, T., Mehbodniya, A., Suzuki, Y.: WLAN interference identification using a convolutional neural network for factory environments. Journal of Communications **16**(7), 276–283 (2021)

An Effective Labeling Model for Minimal Labeled Encrypted Network Traffic Data

Daeho Choi[1][0009-0005-1355-914X] , Minjeong Cho[2][0000-0002-7668-0933] and Kiwook Shon[3][0009-0000-5941-7319]

[1, 2, 3] Seoul National University of Science and Technology, Seoul 01811, Republic of Korea
[1]19101286@seoultech.ac.kr [2]chomj@seoultech.ac.kr [3]kiwook@seoultech.ac.kr

Abstract. With the increase in encrypted network traffic, encrypted traffic classification models have emerged. This paper investigates a model that can effectively label a large number of packets using a small set of labeled packets to generate training data for encrypted traffic classification models. In the labeling accuracy experiments of the proposed model, over 97% accuracy was achieved with more than 5,000 labeled packets. Furthermore, at labeling ratios of 5%, 10%, and 25%, the model demonstrated higher accuracy than previous studies, achieving 91.4%, 95.48%, and 99.12%, respectively.

Keywords: Encrypted Network Traffic, Labeling, Random Forest, Label Spreading

1 Introduction

Training encrypted traffic classification models requires a large amount of labeled data. However, the traffic data labeling process demands prior knowledge and considerable time. Previous studies have attempted to overcome this issue by using semi-supervised learning [1]. Semi-supervised learning is a machine learning technique that can classify large amounts of data with a small amount of labeled data, showing high performance in labeling encrypted traffic data.
Previous studies have primarily considered the ratio of labeled data without considering the performance differences according to the total number of data. In this paper, we propose an enhanced encrypted traffic data labeling model using the LabelSpreading algorithm and Random Forest algorithm. We also compare the performance differences according to the number of total packets and labeled packets. Experimental results show that the proposed model achieves a classification accuracy of 99.72% using 10,000 labeled packets.

2 Background

In Chapter 2, we introduce the machine learning techniques used in the model proposed in this paper.

2.1 Random Forest

Random Forest is a supervised learning algorithm that performs classification tasks using multiple decision trees [2]. The Random Forest algorithm works as a formula $y = \arg max_k \sum_{b=1}^{B} h_b^k(x)$ constructs B decision trees by randomly selecting n features from the total features x_i, and this process is repeated multiple times. For a given label k, the final prediction y is chosen as the most frequent prediction h_b^k from each decision tree b.

2.2 LabelSpreading

LabelSpreading is a semi-supervised learning algorithm that probabilistically propagates labels from labeled data to unlabeled data [3]. The LabelSpreading algorithm works as a formula $f^{(t+1)} = \alpha S f^{(t)} + (1-\alpha)Y$, uses the initial labels Y, predicted labels f, and similarities S between data points to predict the labels at iteration t. The parameter α is used to control the influence received from the initial labels.

3 Methodology

In Chapter 3, we describe the structure of the encrypted traffic labeling model proposed in this paper. The structure of the model is as follows (see Fig. 1).

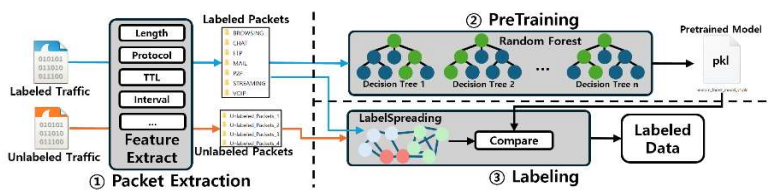


Fig. 1. Encrypted Traffic Labeling Model

(Phase 1 – Packet Extraction) The first step is the packet extraction step, which divides the traffic data extracted from the network into packets and extracts packet-level features. The types of features extracted are Length, Protocol, Time to live, Interval time, Identifier, Flow number, Flags, and Options.

(Phase 2 – Pretraining) In the pretraining phase, the model is trained using the Random Forest algorithm. During the pretraining process, three datasets (ISCXVPN2016 [4], CICIoT_2022 [5], USTC-TFC2016 [6]) are used to learn the characteristics of the labels. The Random Forest algorithm employed in this paper utilizes 100 decision trees for training.

(Phase 3 – Labeling) In the labeling phase, the labeling is performed using the LabelSpreading algorithm. The features of the unlabeled packets are extracted, and the predicted labels from the LabelSpreading algorithm are compared with the labels predicted by the pre-trained Random Forest model. If the predicted labels differ, the

parameters are updated. if the labels are the same, they are retained. This process is repeated for 10 iterations.

4 Experiments

In Chapter 4, experiments were conducted to evaluate the performance of the proposed model.

4.1 Datasets

For the experiments, the ISCXVPN2016 dataset and the ISCXTor2016 [7] dataset were used. The ISCXVPN2016 dataset is categorized into seven types of labels (browsing, chat, FTP, mail, P2P, streaming, and VoIP), and all data is encrypted traffic.

The ISCXTor2016 dataset is categorized into eight types of labels (Audio, Browsing, Chat, FTP, Mail, P2P, Video, VoIP), and all data is encrypted traffic.

4.2 Experiment Settings

The experiments in this paper were conducted by setting the total number of packets to 20,000, 50,000, and 100,000, and the number of labeled packets to 1,000, 5,000, and 10,000, respectively, to measure accuracy. Additionally, results were compared based on the total number of packets when the proportion of labeled pack-ets was the same. In the first experiment, both the labeled and unlabeled data were set using the ISCXVPN2016 dataset. In the second experiment, for cross-validation, the unlabeled data was set using the ISCXTor2016 dataset.

4.3 Result

The results of the experiments are as follows (see Table 1).

Table 1. Model Accuracy

Experiment	Experiment 1			Experiment 2		
Labeled # / Total #	1,000	5,000	10,000	1,000	5,000	10,000
20,000	88.63%	98.31%	99.72%	86.42%	97.97%	99.56%
50,000	81.91%	94.59%	97.80%	81.68%	93.55%	97.24%
100,000	74.07%	92.18%	93.92%	71.84%	91.40%	92.28%

In the first experiment, the model demonstrated an accuracy of over 92% with more than 5,000 labeled packets and a high accuracy of 88% with just 1,000 labeled packets. In the second experiment for cross-validation, the average accuracy decreased by only about 1%, maintaining a high level of accuracy. The experimental results also indicated that as the total number of packets increases, the accuracy decreases. This is due to the insufficiency of labeled packets when labeling a larger number of unlabeled packets, implying that enough labeled packets is required to label more packets effectively.

Additionally, the experimental results confirmed that a higher number of labeled packets leads to higher accuracy at the same labeling ratio. This indicates that the number of labeled packets is a crucial factor in measuring the performance of the labeling model.

5 Conclusion

In this paper, an enhanced encrypted traffic labeling model was proposed. The proposed model demonstrated an accuracy of 91.4% with 5% labeled packets and 99.12% with 25% labeled packets, showing higher accuracy compared to previous studies. It was also confirmed that the performance improves with a larger total number of packets at the same labeling ratio. Future research will aim to reduce the labeling time of the model for more efficient labeling.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2023-00235509, Development of security monitoring technology based network behavior against encrypted cyber threats in ICT convergence environment)

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Fahad, A., Almalawi, A., Tari, Z., Alharthi, K., Al Qahtani, F. S., Cheriet, M.: SemTra: A semi-supervised approach to traffic flow labeling with minimal human effort. In: Pattern Recognition, pp. 1–12. Publisher, Location (2019)
2. Rigatti, S. J.: Random forest. In: Journal of Insurance Medicine, pp. 31–39. Publisher, Location (2017)
3. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5070–5079. Publisher, Location (2019)
4. Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., Ghorbani, A. A.: Characterization of encrypted and VPN traffic using time-related. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414. Publisher, Location (2016)
5. Kahraman, Kostas, Just, M., Lones, M.A.: Externally validating the IoTDevID device identification methodology using the CIC IoT 2022 Dataset. In: Proceedings of [Conference Name], pp. [page numbers]. Publisher, Location (2022)
6. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: Proceedings of the 2017 International Conference on Information Networking (ICOIN), pp. 712–717. IEEE, Location (2017)
7. Lashkari, A. H., Gil, G. D., Mamun, M. S. I., Ghorbani, A. A.: Characterization of Tor traffic using time based features. In: International Conference on Information Systems Security and Privacy, Vol. 2, pp. 253–262. SciTePress, Location (2017)

5G Private Networks: A Survey on Architecture, Threats, Security Requirements, and Research Challenges

Changhyeon Woo[0009−0003−9458−1087], Jieon Oh[0009−0009−4123−4034], Yongho Ko[0009−0006−4905−6912], and Ilsun You[0000−0002−0604−3445]

Kookmin University, Seoul, Republic of Korea
{crow1104, jieon1006, koyh0911, isyou}@kookmin.ac.kr

Abstract. The emergence of 5G technology has fundamentally transformed users' daily lives and industrial structures by advancing various services and industrial sectors. Amid these changes, specific enterprises and organizations such as factories, airports, and hospitals have turned their attention to 5G private networks to provide specialized B2B services to multiple devices simultaneously. Unlike commercial networks, 5G private networks allow only authorized devices to access the network, enabling enterprises and organizations to optimize and efficiently deliver their services. The security requirements for private networks mentioned in the international standard TS 22.261[1] have become more complex and different from those of commercial networks due to the increasing emphasis on authorization. These issues necessitate the reestablishment of the security requirements of private networks. This paper analyzes the latest security threats and vulnerabilities of 5G private networks and aims to establish the security requirements for 5G private networks to enhance their security.

Keywords: 5G mobile communication · 5G Private Network · Security requirement.

1 Introduction

The 5th generation of mobile communications (5G), with its ultra-high speed, ultra-low latency, and massive connectivity, has facilitated its implementation in diverse service sectors alongside industrial development, fundamentally transforming users' daily lives and industrial structures. Amid these ongoing changes, the private 5G network is gaining significant attention in the industrial domain. Unlike commercial networks, private 5G networks provide 5G connectivity to specific enterprises or organizations, allowing only authorized individuals or devices to access the network. Through these characteristics, private 5G networks optimize security requirements, data processing, and communication connections, efficiently delivering the required services and finding applications across various fields. However, because private 5G networks can potentially lead to the leakage of critical data and services within enterprises due to security issues, a high level of security is required. Also, private Networks and commercial networks use the same authentication methods, but the security requirements demanded by the environment and each component are different. These are mentioned in the international standard TS 22.261[1]. Therefore, it is crucial to systematically analyze the security threats and vulnerabilities associated with private 5G networks and to identify the security requirements needed to address these issues effectively. This paper aims to analyze and organize the trends in security requirements for private 5G networks through the classification of each vulnerability by reviewing recent studies on the vulnerabilities and attacks associated with private 5G networks. Additionally, it seeks to propose network security requirements necessary and suggest future research directions based on these requirements to enhance the security of private 5G networks.

2 Private 5G Network

A private 5G network is a dedicated, small-scale network optimized for corporate environments, using allocated frequencies rather than commercial mobile networks. This setup enables non-telecom enterprises to build and operate their own networks, providing enhanced security and performance for applications in various fields, such as manufacturing, education, and healthcare.

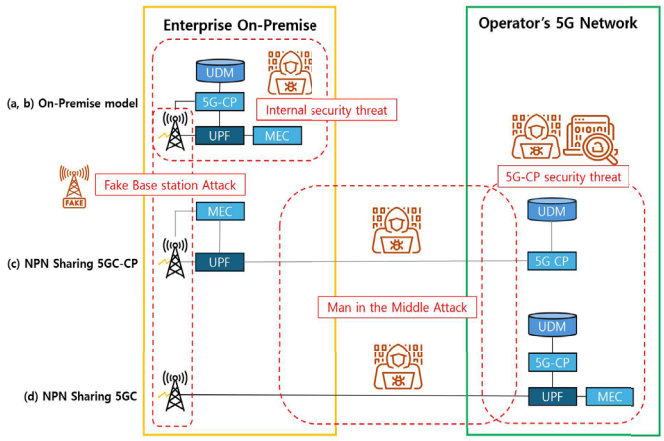


Fig. 1: Private 5G deployment models and threats

The types of 5G private networks and their associated threats are shown in Fig. 1:

(a) Self-Configured On-Premise Model: The company independently builds and operates its 5G network entirely within its premises. All components, including the User Plane Function (UPF) and Multi-Access Edge Computing (MEC), are onsite, ensuring high security and ultra-low latency.
(b) On-Premise Model: A private network operator sets up and manages all 5G components at the enterprise location. This model offers high security and performance, with UPF and MEC located on-premises and remote monitoring by the operator.
(c) NPN (Non-Public Network) Sharing 5GC-CP Model: The private network operator installs base stations, UPF, and MEC within the enterprise, while the Control Plane (CP) is hosted at the operator's central cloud. This requires a dedicated line for CP access. Internal data management ensures high security and excellent performance, but the external CP may expose device information and pose risks if the remote CP experiences issues.
(d) NPN Sharing 5GC Model: The private network operator installs base stations at the enterprise site, while other components are located at the operator's central facility. Data passes through the operator's Core and MEC, which can reduce security compared to other models. Increased communication distance can cause delays, and issues at the remote data center may impact operations.

These models differ based on the setup and management of network components, and how resources are shared between the enterprise and the private network operator.

3 Analysis of Vulnerabilities in Private 5G Networks

In this section, we compare and analyze recent research papers from 2021 to 2024 on private 5G networks based on key elements of information security. We divide these elements into confidentiality, integrity, availability, authentication, and privacy breaches, and organize them into a table according to the associated attacks. The table is as follows.

Security Requirements	[2]	[3]	[4]	[5]	[6]	[7]
Confidentiality	·Spoofing ·Timing Probe Transmission ·Switch Port Scan ·Malware/ Hardware Distribution ·Log Sniffing	·SUCI-Catchers	·User Linkability Attack Due to Authentication Failure	·Byzantine Attack ·MitM ·Spoofing ·Rogue Base Station ·Eavesdropping ·Tampering ·Side-Channel Attack	·AMF Spoofing ·Error in NAS Algorithm Usage ·UP Packet Manipulation and Leakage ·Key Reuse ·CP Packet Interception and Modification During Handover ·Rogue Base Station ·Replay Attack ·MitM ·JSON Parsing Exploitation	·Impersonation ·Data Sniffing ·Spoofing ·MitM ·Eavesdropping
Integrity	·Spoofing ·LLDP Packet Spoofing ·Flow Table Manipulation ·Malware/ Hardware Distribution		·Service Failure Due to Authentication Request Flooding ·Transmission of Supported Encryption List ·Exposure of MIMO and CA Parameters	·SIPDAS ·MitM ·NFV-based Tampering	·Error in NAS Integrity Algorithm Usage ·Exposure and Modification of CP Packets ·UP Packet Manipulation and Leakage ·CP Packet Interception and Modification During Handover ·Rogue Base Station ·Desynchronization Attack ·Misconfigured Routing Table ·Flooding ·MitM	·Impersonation ·Data Sniffing ·Spoofing ·MitM ·Eavesdropping ·Sybil
Availability	·DoS ·Flow Table Manipulation via Fake UPF Packet Injection ·Malware/ Hardware Distribution ·Physical Theft of Network Infrastructure ·Supply Chain Attack	·Service Downgrade Attack Using Fake Base Stations ·DoS Attack Using Fake Base Stations	·Lack of Security for NAS Signaling Message "REGISTRARTION REJECT" ·Service Failure Due to Flooding of Authentication Reject Requests ·Exposure of UE Category in Plaintext ·Exposure of ROHC Profile in Plaintext	·HX-DoS ·SIPDAS ·Byzantine Attack ·Jamming ·DoS ·Rogue Base Station ·Tampering ·Hijacking	·Registration Request Flooding ·Inaccurate SUCI Decryption ·NAS Attack Due to Authentication Failure ·Error in NAS Algorithm Usage ·Non-Emergency Bearer Not Activated ·RES Validation Failure ·Jamming ·Xn-Handover Bidding Down Attack ·Data Redirection ·Session Removal and Modification Requests ·Flooding with Fake PDU Session Establishment Requests ·JSON Parsing Vulnerability ·Rogue Container	·DDos/Dos ·Jamming ·Sybil
Authentication	·Spoofing ·Biometric Authentication Control Attack ·Log Sniffing		·User Linkability Attack Due to Authentication Failure	·Byzantine Attack ·MitM ·Spoofing ·Rogue Base Station	·AMF Spoofing ·NAS Attack Due to Authentication Failure ·Desynchronization ·Replay Attack ·Lack of Authentication and Integrity Protection for User Data ·MitM	·Impersonation ·Spoofing ·MitM ·Sybil
Privacy Violation	·Spoofing ·Malware/ Hardware Distribution	·SUCI-Catchers	·User Linkability Attack Due to Authentication Failure ·GUTI Persistence ·Transmission of Supported Encryption List ·Exposure of Dual Linkability in Plaintext	·Spoofing ·Eavesdropping	·5G-GUTI Reuse ·Key Reuse	·Spoofing ·Eavesdropping

Table 1: Table of Analysis of Vulnerabilities in Private 5G Networks

4 Conclusion

Based on the analysis, recent studies have mainly focused on availability issues among the vulnerabilities and threats affecting 5G specialized networks. This is because the potential damage from vulnerabilities affecting availability can be significant, with service disruptions or reduced system availability leading to substantial economic losses. Therefore, availability is a critical factor. However, existing studies has not sufficiently addressed vulnerabilities related to forward secrecy. If long-term keys are compromised, data encrypted with these keys could be easily decrypted by future quantum computers, or keys derived from the long-term key might also be exposed. Thus, forward secrecy is crucial. Forward secrecy is a critical component in ensuring the security of both current and future communications. It is imperative that encrypted data remains protected against potential future key compromises. As a result, private 5G networks must incorporate forward secrecy and adopt post-quantum cryptography. Both private and commercial 5G networks use exsisting standard 5G authentication protocols, but these protocols do not support forward secrecy. Consequently, private 5G networks are vulnerable to various security issues related to confidentiality, integrity, availability, authentication, privacy breaches, and forward secrecy. This vulnerability can significantly undermine the network's overall trustworthiness and security.

To establish a reliable private 5G network, a comprehensive and systematic approach to security is essential. Future research should focus on security considerations for the components of private 5G networks, such as base stations, UPF, MEC, and network slicing segments, based on existing vulnerability analyses. This approach is expected to play a crucial role in effectively addressing the diverse security threats that private 5G networks may encounter and in enhancing the overall resilience and integrity of the network.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00397469, Development of Private 5G Security Technology for Integrated Private 5G and Enterprise Network Security)

References

1. 3rd Generation Partnership Project(3GPP).: TS 22.261: Service Requirements for the 5G System. (2024).
2. Arpit Tripathi, Abhishek Thakur, Bheemarjuna Reddy Tamma.: Attack Graphs for Standalone Non-Public 5G Networks. 158–163 (2022). https://doi.org/10.1109/FNWF55208.2022.00036.
3. I-Hsien Liu, Meng-Huan Lee, Jung-Shian Li.: Securing 5G Non-Public Networks Against Fake Base Station. 10(2). 156–159 (2023). https://doi.org/10.57417/jrnal.10.2_156.
4. Haque, M.: Analysis of security vulnerabilities in 5G standalone network. (2021).
5. Pelin Angin, Manolya Atalay, Fatma Ceyda Gokce, Ilsun You.: A Survey on the Security of European 5G Private Networks. 8, 162–181 (2022). https://doi.org/10.56801/rebicte.v8i.149.
6. Mohammed Mahyoub, AbdulAziz AbdulGhaffar, Emmanuel Alalade, Ezekiel Ndubisi, Ashraf Matrawy.: Security Analysis of Critical 5G Interfaces. (2024). https://doi.org/10.1109/COMST.2024.3377161.
7. Jaya Preethi Mohan, Niroop Sugunaraj, Prakash Ranganathan.: Cyber Security Threats for 5G Networks. 446-454 (2022). https://doi.org/10.1109/eIT53891.2022.9813965.

Reliable Practical Byzantine Fault Tolerance Mechanism for High Throughput and Low Latency Blockchain Consensus

Namryeong Kim[0009−0004−0482−133X] and Ilgu Lee[0000−0002−5777−4029]

Sungshin Women's University, Seoul, Republic of Korea

Abstract. Practical Byzantine Fault Tolerance (PBFT) is a consensus algorithm that ensures high performance and finality in resolving faults. As blockchain ecosystem rises, a need for fast, secure consensus algorithm to handle high-frequency from users and IoT devices increased. Meanwhile, past studies that aimed scalability enhancement of PBFT has sacrificed signifiant portion of decentralization and security, due to the blockchain trilemma. This study introduces reliable PBFT (rPBFT), which identifies and responds to faulty nodes by aggregating messages during consensus and forming a small, reliable committee for decision-making. rPBFT targets decentralized and secure consensus algorithm, while still advancing performance of conventional PBFT.

Keywords: Blockchain · Consensus · Security.

1 Introduction

A blockchain is a distributed ledger system that records and stores transactions without the need for a trusted third party (TTP) in a peer-to-peer (P2P) network. [1] To ensure data reliability, blockchain uses consensus-based algorithms.

Practical Byzantine Fault Tolerance (PBFT) [2] is a notable consensus algorithm that guarantees liveness within a certain failure threshold. PBFT can reach stable consensus as long as fewer than one-third of the nodes are halted or operating maliciously. In the worst-case scenario, where more than one-third of the participants fail, the consensus process is likely to be deadlocked.

As the original PBFT has limitations in scalability [3], various methods [4,5,6,7] have been proposed to enhance its performance. However, due to the blockchain trilemma involving scalability, security, and decentralization, previous research often sacrifices significant portions of security and decentralization.

In this study, we propose a reliable Practical Byzantine Fault Tolerance (rPBFT) mechanism, which improves the existing PBFT's scalability in identifying and responding to faulty nodes in distributed systems. It assesses node reliability by collecting messages transmitted during the PBFT consensus process and forms a small committee without electing representatives. This approach reduces transaction overhead by offering priority-based selective consensus participation and improves availability by preventing delays caused by faulty nodes.

2 Proposed Method

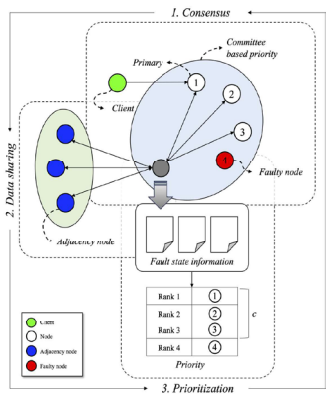


Fig. 1. An overview of rPBFT workflow.

rPBFT consists of prioritization, data sharing, and consensus. By maintaining records of consensus messages and their senders, rPBFT calculates the priority of each node. For example, if messages are received from nodes 3 and 5 out of 5 nodes, the fault state is recorded as [0, 0, 1, 0, 1]. This accumulated fault state is called Faulty State Information (FSI). The count of accumulated messages indicates the historical liveness of each node and is used to determine the priority for election to the block production committee.

Within rPBFT, nodes aggregate FSIs from adjacent nodes and prioritize them based on the shared FSIs. For instance, if a node has three fault states—[0, 1, 0, 1, 0], [0, 1, 1, 1, 1], and [0, 1, 0, 0, 1]—the summed value is [0, 3, 1, 2, 2]. Thus, the reliabilities of the nodes from highest to lowest are nodes 2, 4, 5, 3, and 1. Consequently, it is concluded that a higher summed value indicates higher node reliability.

Consensus involves nodes forming a committee to reach an agreement. This committee comprises the top c nodes (where c<n) based on their priority. Each node forms a unique committee according to its experience. Figure 1 shows how a node organizes its committee. Node D sends messages to the c highest-priority nodes and receives messages from nodes that have chosen D as one of their high-priority nodes.

3 Evaluation

We evaluated latency, throughput, and consensus success rates of conventional PBFT(cPBFT) and our method. Latency is defined as the time from the start to

the end of consensus formation. The latency of rPBFT was measured from when the time the client had sent a request message to when prioritization had been completed. Regarding the throughput, it refers to the amount of information that has been successfully processed; it can be measured by 1:

$$\text{Throughput (bps)} = \frac{\text{Block Height} \times \text{Block Size}}{\text{Transmission Time}} \tag{1}$$

The third is the consensus success rate, which is the rate of successful consensus formations among the total number of attempts to achieve a consensus; it can be measured using 2:

$$\text{Consensus Success Rate (\%)} = \frac{\text{Block Height}}{\text{Total Number of Consensus}} \tag{2}$$

The latency, throughput, and consensus success rates were evaluated using the reliability model. A reliability model is used to estimate, prevent, and manage device failures by predicting the lifetime of each device. It reflects failures due to aging, initial imperfections, and random failures, respectively. In this experiment, the infinite mortality, random failure, wear-out model, and a bathtub curve model that combined the three models are used. The model is randomly sliced and assigned to the nodes to reflect a practical reliability model. Experiments are performed under a 100 node / 1000 block generation scenario.

3.1 Evaluation Results Under the Reliability Model

The evaluation shows that cPBFT and rPBFT are effective in environments with low and high fault rates, respectively. Additionally, rPBFT is effective in processing the faulty nodes of the infant mortality and wear-out models, for which the fault rates tend to increase or decrease over time.

Figure 2 illustrates the latency, throughput, and consensus success rates of cPBFT and rPBFT for a reliability model. In the infant mortality model, cPBFT initially showed a high average consensus success rate of 99.6%. However, this rate decreased sharply due to the high fault rate in the early stages. Additionally, latency increased and throughput decreased because of the fault rate. In contrast, rPBFT maintained a 100% success rate, with constant latency and throughput. Its ability to tolerate a high fault rate at the beginning resulted in a reduced consensus group size, leading to low latency.

Similarly, in the wear-out model, cPBFT started with a high consensus success rate of 98.3%, which dropped sharply due to the high fault rate in the second half, causing increased latency and reduced throughput. In contrast, rPBFT achieved a 100% block consensus formation, with constant latency and throughput throughout the model.

3.2 Evaluation Results in a Faulty Environment

In a fault-free environment, all nodes operated normally. In a faulty environment, faults occurred based on each node's fault rate. By randomly slicing the bathtub

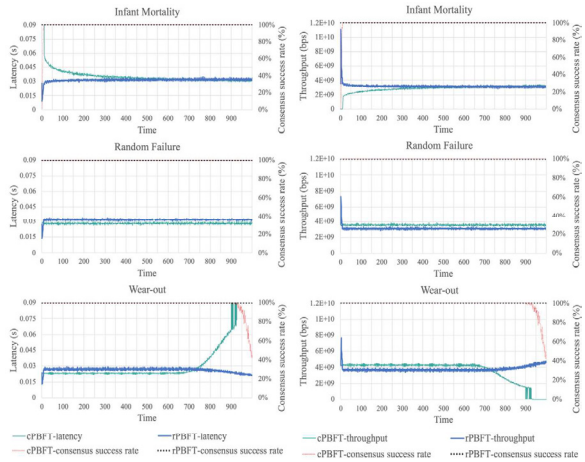


Fig. 2. Latency, Throughput, Consensus Rate Comparison of cPBFT and rPBFT.

curve reliability model into 300 units and assigning them to the nodes, a realistic reliability model reflecting various times and types of nodes was created. For instance, if the reliability model was biased towards the early, midterm, or end stages, it would result in the creation of infant mortality, random failure, and wear-out nodes, respectively. After conducting 100 experiments, each involving 100 nodes generating 300 blocks, the average results were calculated.

In a faulty environment, the consensus success rates of cPBFT and rPBFT were 96.99% and 100%. When cPBFT attempted to create 100 blocks in the faulty environment, it successfully created approximately 97 blocks. In the fault-free environment, the latency for cPBFT and rPBFT was 0.018 and 0.025 seconds, respectively; cPBFT was 0.007 seconds faster. Conversely, in the faulty environment, the latency for cPBFT and rPBFT was 0.05 and 0.024 seconds, respectively; rPBFT was 0.026 seconds faster. Regarding throughput in the fault-free environment, cPBFT and rPBFT achieved 2060.56 and 1484.58 Mbps, respectively; cPBFT was 575.98 Mbps faster. In the faulty environment, the throughput for cPBFT and rPBFT was 686.5 and 1519.5 Mbps, respectively; rPBFT was 833 Mbps faster.

4 Conclusion

In this study, we developed a method to identify and fix faulty nodes in a distributed system without intermediaries. We evaluated node reliability via fault data from cPBFT messages, sharing it with neighboring nodes. A priority-based consensus committee was formed to achieve consensus. The rPBFT method improved scalability through selective participation and enhanced availability by avoiding delays from faulty nodes while preserving decentralization by forming committees without separate representatives.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Insti-tute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Nakamoto, S.: . In: Bitcoin: A peer-to-peer electronic cash system. (2008)
2. Castro, Miguel and Liskov, Barbara and others: Practical byzantine fault tolerance. In: Operating Systems Design and Implementation (OSDI), pp. 173–186. USENIX, LA, USA (1999)
3. Gupta, S., Hellings, J., Rahnama, S., & Sadoghi, M.: An in-depth look of BFT consensus in blockchain: Challenges and opportunities. In: Proceedings of the 20th international middleware conference tutorials, pp. 6–10.(2019)
4. Gengrui Zhang and Hans-Arno Jacobsen.: Prosecutor: an efficient BFT consensus algorithm with behavior-aware penalization against Byzantine attacks. In: Proceedings of the 22nd International Middleware Conference (Middleware), pp. 52–63. Association for Computing Machinery, New York, NY, USA (2021)
5. A. Bessani, J. Sousa and E. E. P. Alchieri: State Machine Replication for the Masses with BFT-SMART. In: Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 355–362. Institute of Electrical and Electronics Engineers, Atlanta, GA, USA (2014)
6. Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham.: HotStuff: BFT consensus with linearity and responsiveness. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, pp. 347–356. Association for Computing Machinery, New York, NY, USA (2019)
7. G. Golan Gueta et al.: SBFT: A Scalable and Decentralized Trust Infrastructure. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 568–580. Institute of Electrical and Electronics Engineers, Portland, OR, USA (2019)

Cryptojacking Container Detection Based on Network Flow Analysis in Kubernetes

Sumin Kim[1], Soomin Lee[1], Jeongeun Ryu[1], Ri-Yeong Kim[1], and Seongmin Kim[1]

Sungshin Women's University, Seoul, Korea
{20221082, 20221121, 20221112, 220236036, sm.kim}@sungshin.ac.kr

Abstract. As the deployment of containers in cloud environments increases, cryptojacking attacks targeting containers are also on the rise. Cryptojacking attacks involve unauthorized use of someone else's computing resources to mine cryptocurrency. This can lead to resource abuse, affecting other legitimate containers and hosts. However, existing research on detecting cryptojacking attacks from a network perspective has not considered cloud environments. Therefore, this paper proposes a framework for detecting malicious cryptojacking containers through network flow analysis. By using an eBPF-based monitoring tool to capture network flow-based features, we analyzed these features and trained machine learning models to evaluate the detection model's performance.

Keywords: Cryptojacking Detection · eBPF · Network flow · Kubernetes.

1 Introduction

The adoption of containers has revolutionized cloud application deployment by encapsulating necessary components into a unified package and executing them within an isolated execution environment. Containers package the elements like libraries and configuration files into a single unit while sharing the host operating system. Meanwhile, attacks targeting distributed containers have risen, with cryptojacking being a common threat. Cryptojacking, also known as mining bot, is an attack where an attacker illicitly uses the victim's computing resources to mine cryptocurrency. According to the 2022 Sysdig report, cryptojacking images were the most frequently detected malicious images [1]. Furthermore, the 2024 SonicWall report indicated that exploiting a vulnerability in JetBrains TeamCity has led to the deployment of the XMRig cryptocurrency miner, demonstrating the persistence of cryptojacking attacks. [2].

Previous research has primarily focused on browser-based or general PC-based cryptojacking attacks by collecting network packets and utilizing traffic-based features to detect cryptocurrency mining activities [3, 4]. However, these methods have limitations in container environments. Containers communicate in a mesh pattern between pods, making it difficult to inspect all traffic from a single location [5]. In addition, when packets generated by containers are collected

after being transmitted to the host network, critical container packet source information, such as virtual network interfaces and network namespaces, is lost [6], making it challenging to identify the original container source from the packets. Therefore, effective network monitoring in container environments requires a lightweight approach for monitoring and analyzing traffic at the container level.

This study proposes a framework for detecting malicious cryptojacking containers in Kubernetes environments by leveraging the extended Berkeley packet filter (eBPF) to collect and analyze network traffic information at runtime. For data collection, eBPF programs are deployed as DaemonSets in Kubernetes, capturing and analyzing the network activity of pods on each node. The collected network traffic features are applied to a machine learning model, demonstrating the framework's effectiveness in detecting malicious cryptojacking containers.

Background. eBPF enables user-defined code to run within the Linux kernel, allowing real-time data processing with low overhead. Due to these properties, eBPF has been used in studies for system monitoring, network analysis, and security detection. Amin Sadiq et al. proposed a method for detecting DoS attacks in Kubernetes [7], while bpfFlowMon used eBPF to monitor network flows in virtualized environments. [5]. Inspired by pioneering studies that utilize eBPF in a containerized environment, we leverage it to detect cryptomining containers.

2 Proposed Methodology

In this study, we propose a framework to analyze malicious cryptojacking containers by leveraging eBPF-based network flow features in Kubernetes. Our goal is to apply these features to train a detection model to differentiate between malicious and benign containers. Figure 1 illustrates the overall architecture.

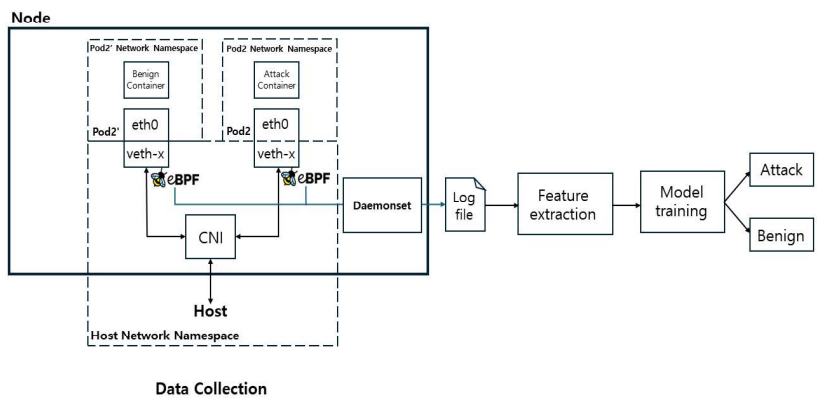


Fig. 1. eBPF-based Cryptojacking Detection Framework in Kubernetes Environment

Network Traffic Collection in Kubernetes. When a pod is created in Kubernetes, a network namespace and Veth pairs are allocated, providing each pod with a unique virtual IP address and an isolated network environment. Veth pairs connect the pod's network namespace to the host network namespace, enabling data transmission. The transmitted data is then routed through other network interfaces or container network interfaces (CNIs), facilitating communication between pods and between pods and external networks. Since the packets generated by each container lose their association with the original container once they transition to the host network namespace, our framework monitors the container's Veth interfaces to collect traffic before network encapsulation, using the container's IP and port. The eBPF monitoring tool, deployed as a DaemonSet, collects traffic information from pods running on each node.
Feature Extraction. We obtained 99 columns, including 5-tuple, traffic statistics, timing information, network-level and application-level information, and other metadata. Figure 2 shows the top 20 features selected after performing feature selection. Note that we exclude 5-tuple to achieve an intrinsic flow pattern for coping with unseen cryptojacking patterns. We used the SelectKBest module from scikit-learn to train the model with the most important features.

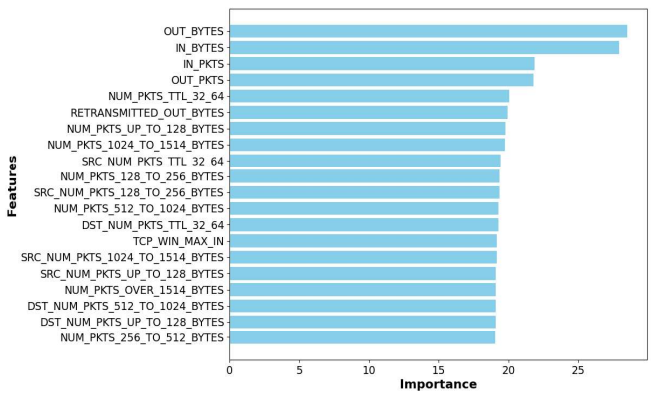


Fig. 2. The importance of the top 20 features

3 Evaluation

We set up a Kubernetes cluster using Minikube [8] on Ubuntu 20.04 and ran Docker images as pods. For malicious container images, we used XMRig and XMR-Stak-rx, commonly employed in cryptojacking to mine Monero. For legitimate containers, we used the media-streaming, data-caching, and web-serving benchmarks from Cloudsuite 4.0 [9], which provides benchmarks for cloud services. Figure 3 illustrates the network traffic patterns based on the extracted network traffic data, showing the relationships between *Out_Bytes* and *In_Bytes*,

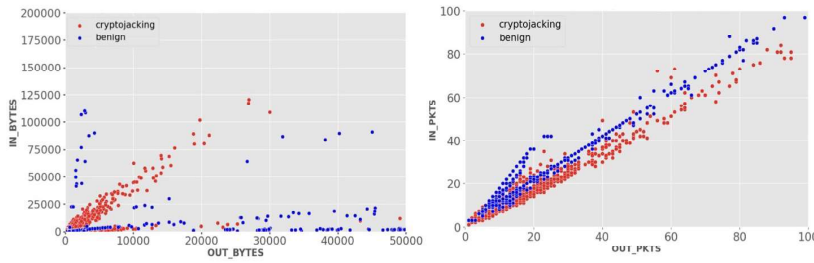


Fig. 3. Graph of Dynamic Feature Analysis for Containers

as well as *Out_Pkts* and *In_Pkts*. It can be observed that mining traffic has higher *In_Bytes* and *In_Pkts* compared to *Out_Bytes* and *Out_Pkts*. This is because the mining program continuously receives mining tasks from the server, and the client sends the results of the tasks back to the server.

We use four machine learning classifiers: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN). The dataset with 20 feature columns and 1 label column was split 7:3 into training and testing sets. Each model was trained using default hyperparameters. Table 1 summarizes the performance. Except for LR, all models deliver exceptionally high performance near 1. Notably, the RF model achieved the highest accuracy (0.99957). This superior performance is likely attributable to the effective selection of high-importance features, which facilitated the identification of significant patterns within the data.

Table 1. ML Performance evaluation

ML Model	Accuracy	Precision	Recall	F1-Score
RF	0.99957	0.99957	0.99957	0.99957
SVM	0.99871	0.99872	0.99871	0.99871
KNN	0.99592	0.99594	0.99592	0.99592
LR	0.74126	0.76776	0.74126	0.72964

In summary, network flow-based features monitored by utilizing eBPF in a dynamic cloud-native environment can be effective to distinguish mining bots.

Acknowledgment. This work is partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00351898), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Sysdig. https://sysdig.com/2022-cloud-native-security-and-usage-report/, last accessed 2024/07/26.
2. SonicWall. https://www.sonicwall.com/threat-report//, last accessed 2024/07/26.
3. Lijian Dong, Zhigang Li, Xiangrong Li, Xiaofeng Wang, and Yuan Liu. Identification technique of cryptomining behavior based on traffic features. *Frontiers in Physics*, 11:1269889, 2023.
4. Yebo Feng, Jun Li, and Devkishen Sisodia. Cj-sniffer: Measurement and content-agnostic detection of cryptojacking traffic. In *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 482–494, 2022.
5. Matteo Repetto and Alessandro Carrega. Monitoring network flows in containerized environments. In *Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools*, pages 32–55. Springer, 2022.
6. Jaehyun Nam, Seungsoo Lee, Phillip Porras, Vinod Yegneswaran, and Seungwon Shin. Secure inter-container communications using xdp/ebpf. *IEEE/ACM Transactions on Networking*, 31(2):934–947, 2022.
7. Amin Sadiq, Hassan Jamil Syed, Asad Ahmed Ansari, Ashraf Osman Ibrahim, Manar Alohaly, and Muna Elsadig. Detection of denial of service attack in cloud based kubernetes using ebpf. *Applied Sciences*, 13(8):4700, 2023.
8. Minikube. https://kubernetes.io/ko/docs/tutorials/hello-minikube/, last accessed 2024/07/26.
9. CloudSuite 4.0. https://github.com/parsa-epfl/cloudsuite, last accessed 2024/07/26.

Autoencoder-Based Face De-identification and Swapping for Enhanced Privacy Protection

So Yeon Kim, Seong Hyun Min, Ma Il Jo, and Won Gyum Kim

Aideep.Inc., South Korea
{sykim,shmin,mijo,wgkim}@aideep.ai

Abstract. In the digital era, privacy protection has become a critical issue, and face de-identification technology is emerging as a key solution. This paper proposes an autoencoder-based model for face de-identification and swapping, aiming to protect privacy while providing various face swap capabilities. The proposed model comprises an encoder and two decoders, which transform faces into different domains. The combination of perceptual loss and style loss significantly enhances the visual quality and consistency of the face transformations. The model's effectiveness in reducing identifiability while preserving face features was validated. The de-identified images reduce identification risk while maintaining key traits, and the swapped images alter appearance while preserving the original features.

Keywords: Face de-identification · Face swap · Autoencoder · Privacy.

1 Introduction

With the increasing use of social media and online platforms in modern society, privacy has become more important than ever. We routinely share numerous photos and videos, resulting in significant exposure of face data on the internet. However, this exposure can threaten our privacy, and cases of unauthorized collection and misuse of face images are on the rise.

Furthermore, there is a growing trend of companies and government agencies collecting and analyzing face data for security and surveillance purposes. While the advancement of these technologies offers significant convenience, it also raises increasing concerns about the infringement of privacy rights. This issue is not merely a technical one but is deeply intertwined with social, legal, and ethical concerns. Face de-identification technologies have gained increased prominence due to the growing emphasis on data privacy protection, and they are being employed across various industries to mitigate the risk of personal data leakage.

This paper proposes an efficient autoencoder-based model that uses two decoders to simultaneously address face de-identification and face swapping. By combining perceptual loss and style loss, the model enhances the visual quality of reconstructed images and enables effective transformations across various domains with minimal data and computational resources.

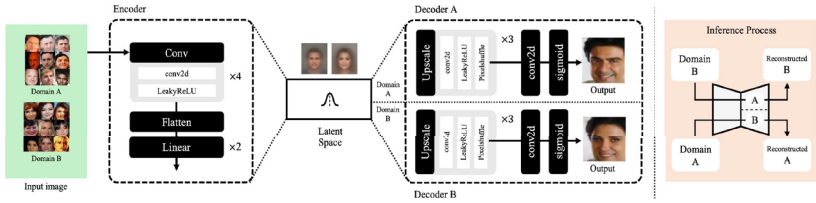


Fig. 1. Architecture of the Face De-Identification and Swapping Model - The model employs an encoder and two decoders, reconstructing images from different domains, and during inference, the decoders are switched to generate the final output.

2 Proposed Model

In this paper, we propose an autoencoder-based network that simultaneously provides privacy protection and face swapping capabilities. As illustrated in the Fig. 1, we define the input domains A and B for transformation as male and female, respectively. The overall architecture comprises a single encoder and two decoders, each designed to effectively reconstruct the input images from their respective domains. The encoder sequentially passes through convolutional layers, progressively reducing the input features. Subsequently, the 2D feature maps are transformed into 1D vectors through a flattening process, and then projected into the latent space via linear layers. The decoder reconstructs images using the latent space vectors generated by the encoder. The decoders pass through upscaling layers composed of convolution, LeakyReLU, and pixel shuffle, reconstructing the images to their original resolution, and finally converting them into RGB format through the last convolution layer. Each decoder learns to reconstruct images from the input domain during training, and during inference, reconstructs images through the decoder of the other domain.

To effectively perform face de-identification, we introduced two additional loss functions in our autoencoder model besides the traditionally used reconstruction loss: perceptual loss and style loss. The reconstruction loss measures the pixel difference between the input image and the image reconstructed by the model, effectively guiding the model to restore the input. Perceptual loss mimics the visual perception of the original image by comparing the feature maps between the two images. Lastly, style loss is used to enhance the quality of the reconstructed image by computing the correlations between feature maps using the Gram matrix, thereby incorporating the style elements of the original image into the reconstructed image. The total loss function is defined as Eq. 1.

$$L_{total} = \lambda_1 L_{L1} + \lambda_{pct} L_{perceptual} + \lambda_{sty} L_{sty} \tag{1}$$

$$L_{L1} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|_1, \quad L_{pct} = \sum_{l} \|\phi_l(x) - \phi_l(\hat{x})\|_2^2, \quad L_{sty} = \sum_{l} \|G_l(\phi_l(x)) - G_l(\phi_l(\hat{x}))\|_2^2 \tag{2}$$

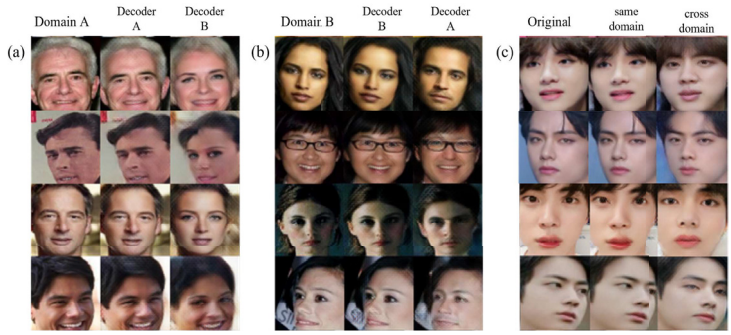


Fig. 2. Face De-identification and Swapping Results. (a) and (b) show face de-identification, while (c) shows face swapping. Each result is organized into three columns: original image, same-domain reconstruction, and cross-domain reconstruction.

Table 1. Performance comparison of different models for face de-identification and face swap tasks using metrics like Identity similarity, LPIPS, SSIM, and PSNR.

Task	Model	Identity↑	Identity↓	LPIPS↓	SSIM↑	PSNR↑
Face de-identification	DeepPrivacy [1]	-	0.080	0.310	0.791	21.3
	CIAGAN [2]	-	0.041	0.395	0.522	18.6
	IDeudemon [3]	-	0.012	0.213	0.898	25.9
Face swap	FaceShifter [4]	0.316	-	0.398	0.634	21.7
	E4S [5]	0.383	-	0.241	0.860	23.0
	StyleIPSB [6]	0.538	-	0.254	0.811	22.4
	Ours	**0.916**	**0.006**	**0.211**	**0.868**	**25.6**

3 Experiments

To validate the performance of the proposed model, we conducted evaluations using the CelebA-HQ dataset, randomly selecting 500 male and 500 female face images. To verify the face swapping performance, 150 face images of two randomly selected males were collected. The model was trained on a single GTX A6000 with a batch size of 64 for 100,000 iterations. The optimizer used was Adam, with a learning rate set to 5e-5.

The Table. 1 shows the performance evaluation results of the proposed model. To assess both face de-identification and face swapping, we compared the performance of the proposed model with representative face de-identification and face swapping models. Identity compares the similarity between images before and after de-identification in the case of de-identification models, and between the original source image and the generated image in the case of face swap models. The proposed model uses two decoders for each domain. LPIPS, SSIM, and PSNR are used to assess the quality of generated images. In the first domain, the proposed model exhibited higher identity values, demonstrating its effectiveness in restoring the domain accurately. Conversely, in the second domain, the proposed model achieved significantly lower identity values, indicating superior

de-identification, and showed similar or improved quality metrics compared to other models. The visual analysis of de-identified and face-swapped images is presented in Fig. 2. Panels (a) and (b) show de-identified images for males and females, respectively, while (c) displays face-swapping results for two specific male targets (V and Jin). Each row is organized as follows: left-original image, center-same domain decoder result, right-other domain decoder result. The same domain decoder effectively reconstructs faces from the original images, while the other domain decoder successfully performs de-identification. However, due to the age distribution of the dataset, the generated faces tend to exhibit fewer wrinkles and smoother skin.

4 Conclusion

This paper proposes an autoencoder-based model for face de-identification and swapping. A shared encoder compresses images from two domains, while two decoders transform the images into the opposite domain. By combining perceptual and style losses, the model enhances visual quality and consistency. The model's performance was validated through quantitative and qualitative evaluations. De-identified faces reduce identifiability while retaining key features, and swapped faces effectively alter appearance while preserving essential traits. Future research will explore additional face transformation functions, such as age progression and emotion expression changes, and develop lightweight models for real-time performance.

Acknowledgments. This work was supported by Korea Internet & Security Agency (KISA) grant funded by the Korea government(PIPC) (No.RS-2022-IS000004, Research&Development for Right to be forgotten in Video Contents provided by limited online service providers).

References

1. Hukkelås, H., Mester, R., and Lindseth, F.: Deepprivacy: A generative adversarial network for face anonymization. In: International Symposium on Visual Computing, pp. 1–12. Springer International Publishing (2019)
2. Maximov, M., Elezi, I., and Leal-Taixé, L.: Ciagan: Conditional identity anonymization generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2020)
3. Wen, Y., et al.: Divide and conquer: A two-step method for high quality face de-identification with model explainability. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2276–2285 (2023)
4. Li, L., et al.: Faceshifter: Towards high fidelity and occlusion aware face swapping. arXiv preprint arXiv:1912.13457 (2019)
5. Li, M., et al.: E4S: Fine-grained face swapping via editing with regional GAN inversion. arXiv preprint arXiv:2310.15081 (2023)
6. Jiang, D., et al.: StyleIPSB: Identity-preserving semantic basis of StyleGAN for high fidelity face swapping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7641–7650 (2023)

One-Pixel Attack Mitigation Technique

Yeon-Ji Lee[1][0000-0002-0482-2381] and Il-Gu Lee[1] [0000-0002-5777-4029]

[1] Sungshin Women's University, Seoul, 02844, Korea
{cselab.lyj, iglee19}@gmail.com

Abstract. As artificial intelligence (AI) technology is integrated into various fields, such as image classification and autonomous driving, new threats exploiting the vulnerabilities of AI have emerged. Among adversarial attacks, the one-pixel attack is a technique that disrupts the correct classification by distorting a single pixel in an image. This paper proposes an image processing technique to mitigate one-pixel attacks. Evaluation results show that the proposed method recovers 94 out of 100 one-pixel attack images when the image size is adjusted to 27x27, improving the confidence of the images by 68.89%.

Keywords: adversarial attack, one-pixel attack, image processing

1 Introduction

Recently, artificial intelligence (AI) technology has rapidly advanced in various fields, such as image classification, object detection, voice control, and autonomous driving [1,2]. These advancements have had a broad impact on our lives and society. They are expected to lead to even greater innovation and development as they are integrated into more areas in the future. However, alongside the development of AI-based image processing technologies, there has been an increase in attempts to exploit AI, with adversarial attacks using adversarial examples becoming a new threat.

Adversarial attacks involve generating adversarial examples by distorting images. Such attacks can have serious implications for autonomous driving and voice recognition [2]. Therefore, developing techniques to counter these intelligent attacks has become crucial. This study focuses on implementing and mitigating one-pixel attacks, which distort only a single pixel, through proposed image processing techniques. The main contributions of this study are as follows:

- Proposing an image processing technique to mitigate one-pixel attacks.
- Proposing an evaluation framework to assess the performance of one-pixel attacks and corresponding mitigation techniques.

The structure of this paper is as follows: Section 2 analyzes previous research on countering one-pixel attacks. Section 3 explains the proposed method, while Section 4 analyzes the experimental and performance evaluation results. Finally, Section 5 concludes the paper.

2 Related work

Various techniques have recently been studied to detect and defend against adversarial attacks. Muhammad A. H [3] proposed a robust Principal Component Analysis (RPCA) based image restoration model that applies the Accelerated Proximal Gradient approach to detect and mitigate one-pixel attacks. The proposed model demonstrated that the RPCA model could correctly restore images by optimizing the hyperparameter lambda. However, this model requires a recovery model and has limitations as the parameter values vary for each image.

Rahul Paul [4] proposed a multi-initialization-based ensemble model to mitigate adversarial attacks, including the Fast Gradient Sign Method (FGSM) and one-pixel attacks. The model trained with one-pixel attack images included in the test dataset showed about a 3% decrease in accuracy compared to the normal dataset. The proposed ensemble model, which combines three CNN models with seven weight initialization layers, improved performance by 8% compared to the one-pixel attack. However, the proposed method is complex due to the combination of three CNN models and does not achieve significant performance improvements relative to the increased complexity.

Existing studies propose complex algorithms to mitigate one-pixel attacks. However, Internet of Things (IoT) devices such as security cameras and autonomous vehicles, which are the primary targets of one-pixel attacks, are unsuitable for implementing complex algorithms. Therefore, there is a need for research on simpler mitigation techniques.

3 One-Pixel Attack Mitigation Technique

This section describes the proposed one-pixel attack mitigation technique and analyzes its performance evaluation results.

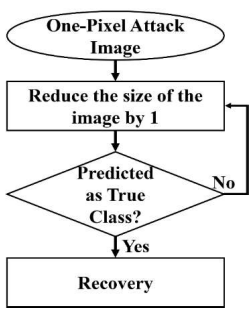


Fig. 1. Flowchart of One-Pixel Attack Mitigation Technique.

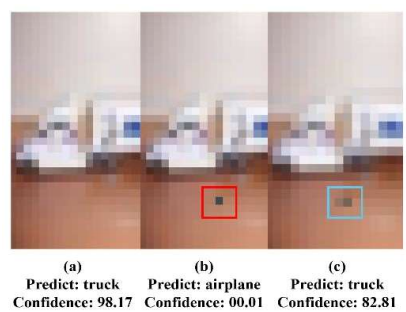


Fig. 2. Example of One-Pixel Attack Mitigation Technique.

Figure 1 illustrates the flowchart of the proposed technique. The model iteratively processes the image by lowering its resolution by one pixel at a time until it is correctly

classified into the actual class. Due to the nature of one-pixel attacks, the pixel affected by the attack significantly differs from its surrounding pixels. Thus, the proposed technique mitigates the one-pixel attack by lowering the image resolution so that the attacked pixel blends with its surroundings.

4 Evaluation and Analysis

This section analyzes the environment and performance evaluation results for implementing the proposed model. The CIFAR-10 [5] image dataset used in this study consists of 60,000 images, with 6,000 images per each of the ten classes. The mutually exclusive classes make the dataset suitable for testing one-pixel attacks and mitigation techniques. For performance evaluation, 50,000 images were used as training data and 9,900 images as test data. Additionally, 100 evaluation data with one-pixel attacks were generated to validate the one-pixel attack mitigation technique. A ResNet (Residual Neural Network) model was used for training and classification through transfer learning.

In this evaluation environment, the confidence level was assessed by gradually reducing the resolution of the one-pixel attack images until the model classified them into the correct class. Here, the confidence represents the probability that the model predicts the correct class, indicating attack success and mitigation.

Figure 2 shows examples of images with the proposed method applied. (a) is the original image without attack, correctly classified into the appropriate class. (b) is an image with a one-pixel attack, with the attacked pixel indicated in red. This image was misclassified into another class, resulting in a low confidence score. (c) is the same image as (b) but processed with the image technique, reducing the original size from 32x32 to 31x31. The blue area indicates where the attack noise has been mitigated. The (c) image was reclassified into the correct class, resulting in improved confidence. Although the confidence of the mitigated (c) image was 15.36% lower than that of (a), it improved by 82.80% compared to the one-pixel attack image.

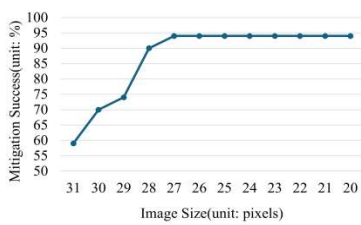


Fig. 3. Attack Mitigation Rates According to Image Size Adjustment Using the Proposed Method.

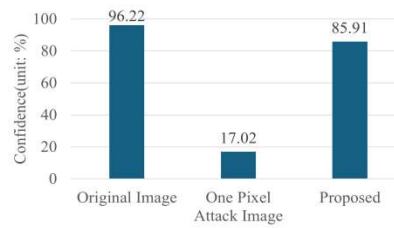


Fig. 4. Confidence Evaluation for One-Pixel Attack Mitigation.

Figure 3 presents a graph of the mitigation rates for one-pixel attacks according to image size adjustment, with the size adjusted in pixel units. When the image size was

adjusted to 31x31 pixels, 59 out of 100 one-pixel attack images were mitigated; at 30x30 pixels, 70 images; at 29x29 pixels, 74 images; and at 28x28 pixels, 90 images were mitigated. The point at which the mitigation success rate converges is at 27x27 pixels, where 94 out of 100 images were mitigated. The remaining six images could not be mitigated, even when the image size was reduced to 1x1 pixel.

Figure 4 shows the average confidence scores for the original image, the one-pixel attack image, and the image with the applied mitigation technique. This excludes the 6 unmitigated images, confirming that the one-pixel attack was successfully applied, and that the mitigation technique effectively improved the confidence of the attacked images by 68.89%.

5 Conclusion

This paper proposed a method for mitigating one-pixel attacks using image processing techniques. A base model was created using the ResNet model trained on the CIFAR-10 dataset, and the mitigation technique was evaluated using 100 one-pixel attack images. The experimental results showed that the proposed model could mitigate one-pixel attack images with a simple process, mitigating 94 out of 100 images when the image size was reduced from 32x32 to 27x27. Additionally, the confidence of images with one-pixel attacks improved by 68.89%, demonstrating the effectiveness of the proposed model.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of Artificial Intelligence Adversarial Attack and Defense Technologies. In: Applied Sciences, vol. 9, issue 5 (2019).
2. Xu, H., Ma, Y., Liu, H.-C., Deb, D., Liu, H., Tang, J.-L., Jain, A.K.: Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. In: International Journal of Automation and Computing, vol. 17, pp. 151–178 (2020).
3. Husnoo, M.A., Anwar, A.: Do not get fooled: Defense against the one-pixel attack to protect IoT-enabled Deep Learning systems. In: Ad Hoc Networks, vol. 122 (2021).
4. Paul, R., Schabath, M., Gillies, R., Hall, L., Goldgof, D.: Mitigating Adversarial Attacks on Medical Image Understanding Systems. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), USA (2020).
5. Tensorflow: cifar10. Last updated: 06 December, last accessed 22 April 2024. Available: https://www.tensorflow.org/datasets/catalog/cifar10.

Enhancing Alarm Systems Utilizing In-Network Computing for Mitigating Tampered Alarms

Junghyun Moon[1][0009-0005-9148-2168], Eunji Chang[1][0000-0002-8364-9213] and Kookheui Kwon[1][0000-0001-6104-6000]

[1] Korea Institute of Nuclear Nonproliferation and Control, 1418 Yuseong-daero, Korea
moon_aver@naver.com

Abstract. The acceleration of industrial automation technologies has led to an increase in the number of sensors deployed within Operational Technology (OT) environments. Additionally, the integration of emerging technologies such as IoT, AI, and 5G to enhance productivity has augmented the complexity. Consequently, to manage industrial operations effectively, strategies to minimize abnormal situations and false alarm generation are being implemented through alarm management standards such as IEC 62682 and ANSI/ISA 18.2. However, traditional alarm systems transmit data to central systems, making them susceptible to various interferences, including adversarial attacks, necessitating measures to secure the data. This paper proposes a secure alarm system based on In-Network Computing (INC), demonstrating through experiments that it can safeguard OT data while maintaining production efficiency.

Keywords: Operational Technology (OT), In-Network Computing (INC), Alarm system, Noise, Interference

1 Introduction

With the advancement of Information and Communication Technology (ICT), the equipment and devices in smart factories are increasingly connected and utilized through wireless communication. This interconnection of process data facilitates the optimization of the production environment and offers flexibility through automation. While wireless connectivity is convenient and enables mobility, it is increasingly being deployed in nuclear facilities, including Small Modular Reactors (SMRs), where its use was once fundamentally prohibited. Nonetheless, it remains vulnerable to signal interference, has a limited range, and is slower compared to wired connections. Consequently, in operational technology (OT) environments, monitoring and alarm systems, as well as defense-in-depth strategies, are employed to prevent severe damage from various threats. However, traditional alarm systems, predominantly operated on threshold-based, centralized architectures, are negatively impacted by equipment aging, failures, weather conditions, and attacks. This necessitates secure and efficient data collection and processing methods to address the limitations of data centralization. The main contributions of this study are as follows:

- We propose the alarm system utilizing In-Network Computing (INC) to effectively detect anomalies and mitigate performance degradation caused by interferences.
- We provide experimental results demonstrating that the proposed method protects sensitive data while ensuring data processing efficiency and reliability.

This paper is structured as follows: Chapter 2 reviews related work, Chapter 3 proposes the method to enhance the security of the alarm system based on In-Network Computing, Chapter 4 validates the performance of the proposed method through experiments, and Chapter 5 concludes the study.

2 Related work

In recent operational technology (OT) industry processes, the number of devices such as sensors and actuators is increasing to prevent large-scale accidents, such as those causing loss of life or property damage. Research is actively being conducted to develop methods for effectively responding to abnormal processes.

Emanuel Sousa Tomé et al. [1] proposed an online data-based anomaly detection method considering the state of sensors to address issues caused by failures and frequent false alarms in fire alarm systems. While their experiments demonstrated performance variations depending on environmental conditions, the study did not address the limitations of centralized systems. Kai Yang et al. [2] proposed a data recovery strategy for PLC devices to recover from abnormal states. Their experiments proved that security functions can be added via software updates alone without hardware modification, demonstrating practicality. However, their method processes communication data between PLC devices centrally, which can increase system latency and network load. Jan Vávra et al. [3] proposed an anomaly detection system considering the characteristics of industrial environments to defend effectively against attacks targeting industrial control systems. Their experiments confirmed the enhanced security of the proposed method based on four machine learning algorithms. However, the limitations of centralized systems were not resolved, indicating a need for efficient data processing methods, such as parallel computing.

3 Proposal method

In this chapter, we propose an INC-based alarm system to enhance the efficiency and reliability of OT alarm systems. Alarms indicate abnormal states and require prompt action from operators by using auditory and visual means when issues such as system malfunctions occur[4]. Therefore, it is crucial to protect raw data containing sensitive information while effectively detecting anomalies to maintain the performance of the alarm system.

Sensors and equipment in OT environments can experience data distortion and performance degradation due to various interferences during data generation and transmission. These interferences include not only unintentional noise such as aging, weather, and ambient noise but also malicious attacks like adversarial attacks and jamming

attacks. An adversarial attack involves adding noise δ to the input data x. Depending on the method of adding δ, techniques such as Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Boundary Attack can be classified. If the attacker is aware of the model's internal structure, parameters, and gradients, it is a White-box Attack. The system is compromised through a Black-box Attack method by executing multiple queries and updates based on the input and output information of the model.

Traditional centralized alarm systems incur unnecessary latency by transmitting all data to a central system for interference removal and anomaly detection. However, the proposed method leverages INC, allowing network devices to perform logical and arithmetic operations locally before sending the computation results to the central management system, rather than transmitting raw data. As illustrated in Fig. 1, this approach protects sensitive data from interference. INC involves network devices such as switches and routers along the data path utilizing their computing resources to perform computational functions. By processing and transmitting data in parallel, INC optimizes task execution speed, reduces latency, and decreases network congestion, thereby efficiently managing system malfunctions.

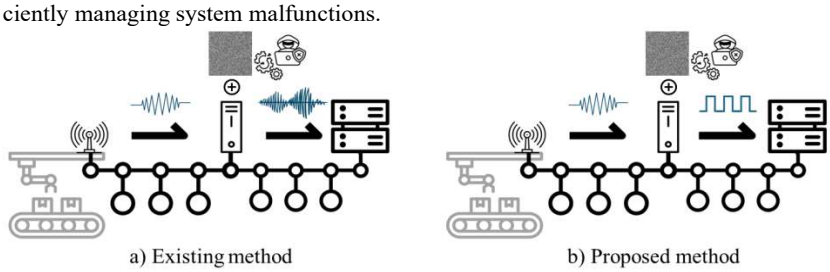


Fig. 1. In-Network Computing-based data transfer and processing structure

4 Experimental Environment and Performance Analysis

In this study, we assumed a data transmission environment with noise and adversarial attack interference to verify the effectiveness of the proposed method. We utilized various components of an in-network computing alarm system, analogous to neurons and connections between neurons in a neural network. Devices and nodes of the alarm system were mapped to neurons, while wired and wireless links were mapped to connections between neurons. A comparison was performed with a logistic regression model. Given the extensive and varied risk situations in the OT industry, encompassing sectors such as energy and manufacturing, we used a dataset consisting of 3,000 entries of five alarm types (danger, fire, gas, none, tsunami), which included scenarios of fire and natural disasters[5].

As depicted in Fig. 2, the accuracy and loss of the traditional alarm system and the proposed method were measured on training and validation data. Both the traditional and proposed methods show performance degradation proportional to the intensity of interference compared to normal conditions. However, under the same level of

interference, the proposed method demonstrated higher accuracy and lower loss compared to the traditional method. This confirms that in-network computing provides security and efficiency while showing robustness against interference with the proposed mechanism.

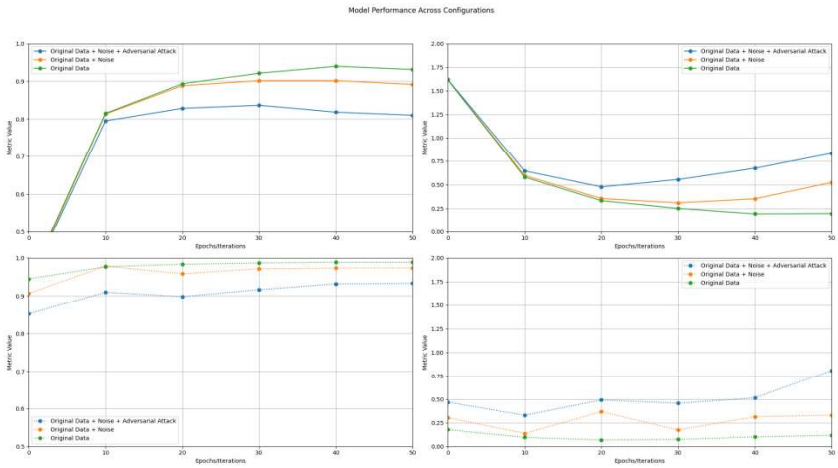


Fig. 2. Model performance across configurations

5 Conclusion

Availability and safety are paramount in OT systems, and they must continue to operate without system interruptions even in the event of failures, ensuring fault containment. Therefore, we proposed an OT security enhancement method utilizing INC and confirmed through experiments that the proposed method can protect data while efficiently collecting and processing it in an interference environment. The method studied in this paper can be considered by nuclear operators to enhance cybersecurity of their alarm system. Future research will focus on developing a fault-tolerant system that encompasses endpoint and network security, allowing for the isolation and management of failed components to maintain normal operations.

References

1. Sousa Tomé E, Ribeiro RP, Dutra I, Rodrigues A.: An Online Anomaly Detection Approach for Fault Detection on Fire Alarm Systems. Sensors 23(10) (2023)
2. Yang Kai, Haining Wang, and Limin Sun.: An effective intrusion-resilient mechanism for programmable logic controllers against data tampering attacks. Computers in Industry 138 (2022)

3. Vávra, J., Hromada, M., Lukáš, L., & Dworzecki, J.: Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment. International Journal of Critical Infrastructure Protection 34 (2021)
4. ISA, A.: Isa-18.2: Management of alarm systems for the process industries, International Society of Automation, Durham, NC, USA, ISO 690 (2009)
5. Kaggle.: Alarm dataset, https://www.kaggle.com/datasets/devisdesnug/alarm-dataset (2022)

Multi-channel-Based Physical Layer Security for Dynamic Key Update

Chae-Yeon Park[0009-0006-7677-4100] , Sun-Jin Lee[0000-0001-9114-2992] , Il-Gu Lee[0000-0002-5777-4029]

Sungshin Women's University, South Korea

{20221097,220237017,iglee}@sungshin.ac.kr

Abstract. Recently, wireless devices used in all industries are vulnerable to eavesdropping attacks due to the unique characteristics of wireless communication. To solve this problem, this study proposes a dynamic physical layer security technique using multi-channels. The proposed technique continuously updates the keys to increase the complexity of a brute-force attack by at least 2^{52} times compared to the conventional model to enhance security.
Keywords: Multi-channel, Key update, Wireless communication systems, Physical layer security

1 Introduction

Mobile services in use so far are vulnerable to eavesdropping due to the unique characteristics of wireless communication. As a security measure against this, physical layer security (PLS) technology is being studied. In the physical layer, the preamble is a unique value generated based on the timing, frequency, and phase of receiving the signal. The receiving terminal may authenticate using the preamble value and identify the transmitting terminal. However, if an attacker may eavesdrop on the preamble to extract channel state information (CSI) of the corresponding terminal, the confidentiality of channel reciprocity can no longer be guaranteed, and the probability of a key leakage increases.

Therefore, this study proposes a key update technique that is robust to subchannel analysis through eavesdropping. The proposed technique utilizes the characteristics of multi-channel and preamble to update keys. It uses a unique value based on the channel reciprocity of the auxiliary channel preamble as key update rule information on the main channel. The attacker cannot identify the key's update pattern because the key is updated every time the radio channel state changes. In addition, the proposed technique is highly resistant to brute-force attacks. In particular, the triple-channel key update technique using two auxiliary channels has a greater difficulty in key recovery than the dual-key update technique using one auxiliary channel. However, the triple-channel key update technique increases delay time compared to the dual-channel key update technique. Therefore, the proposed technique updates the key by dynamically adjusting the number of channels used in consideration of the communication channel quality of the wireless device.

2 Related Work

Recently, research has been conducted focusing on PLS to cope with wiretapping attacks in wireless communication. Long et al. [1] modified the watermarking of the existing spreading code authentication (SCA) technique to watermark only some of the preamble chips, improving authentication accuracy without deteriorating communication performance. However, if an attacker learns the pattern of the preamble watermarking rule using SCA, there is a limitation in that it can guess the preamble and cannot take into account the dynamic channel fluctuation characteristics of the wireless network. Zhang et al. [2] propose an Extensible Preamble Modulation (eP-Mod) compatible with the 802.11n/ac/ax protocol by converting a specific frame part into a data field. The proposed method can utilize the existing preamble to support coordination functions at the physical layer at no additional cost. However, eP-Mod is vulnerable to wiretapping attacks because it only considers the usability of the preamble. Li et al. [3] propose a decomposed and distributed modulation (DDM) modulation technique that divides data into two signals instead of simply transmitting it as a single signal. The DDM technique enhances security because the eavesdropper must acquire both signals to obtain the original data. However, there is a limitation that not only the eavesdropper but also the normal user must acquire both signals without error to receive data. In this paper, we propose a preamble-based key update technique that dynamically utilizes multi-channel to improve the limitations of previous studies.

3 Dynamic Key Update Mechanism using Multi-channel

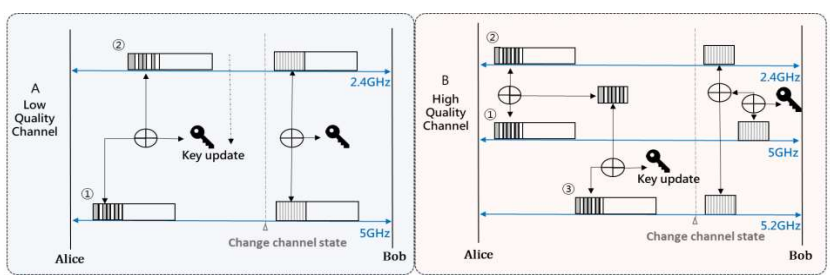


Fig. 1. Proposed Key Update Mechanism

Figure 1 shows the operation process of the proposed preamble-based key update technique. Dual channels are used at low channel quality, and triple channels are used at high channel quality. Two terminals (Alice and Bob) use the 2.4 GHz channel as the main channel and the 5 GHz and 5.2 GHz channels together as the sub-channel. In step 1, some subcarriers among 52 subcarriers constituting the sub-channel are selected and labeled with valid values. In step 2, the CSI value of the subcarrier labeled with a valid value in step 1 is transmitted to the main channel. In step 3, a part of the subcarriers of

the main channel preamble is selected based on the CSI value received from the sub-channel as updated rule information. In step 4, the key is updated after the subcarrier of the sub-channel is labeled with a valid value in step 1, and the subcarrier of the main channel selected in step 3 is XOR calculated. For key updates using the triple channel, the key is updated after the operation process of step 4 above proceeds in the same manner as the 5.2 GHz channel. The operation process is repeated whenever the channel state changes.

4 Numerical Analysis

To evaluate the security of the proposed multi-channel method in this study, after excluding the possibility of key leakage and error during communication between nodes, the attack complexity of the conventional single-channel method [1] and the proposed model was compared with an equation. The conventional model, the single-channel method, utilizes only one channel and does not have a process of updating the key based on the preamble. Table 1 shows the attack complexity of the triple-channel-based proposed model and the single-channel-based conventional model as an equation in each of the three cases.

Table 1. The attack complexity of the triple-channel-based proposed model and the conventional model in three environments

Case	Condition	Proposed model Attack Complexity	Conventional model Attack Complexity
Case 1	Eve knows Two Channel	2^{52}	2^{52}
Case 2	Eve knows only One Channel	$2^{52} \cdot 2^{52}$	2^{52}
Case 3	Eve Doesn't know all three Channel	$2^{52} \cdot 2^{52} \cdot 2^{52}$	2^{52}

Attack complexity refers to the number of cases in which an attacker attempts to succeed in a random substitution attack. Case 1 is when the attacker recognizes both the main channel and one sub-channel. Case 2 is when the attacker recognizes only the main channel. Case 3 is when the attacker does not recognize all three channels. Case 1 is the strongest attack environment, and Case 3 is a relatively weak one. In the proposed model, the attack complexity increases by 2^{52} times as it goes from Case 1 to Case 3. From Cases 1, 2, and 3 in Table 1, it can be seen that the attack complexity of the proposed model is proportional to the length of the key. On the other hand, the conventional model has a constant attack complexity of 2^{52} in all cases. Therefore, the proposed model increases attack complexity compared to the conventional model, responds to brute-force attacks, and strengthens security.

5 Conclusion

As the eavesdropping technique of wireless communication becomes intelligent and automated, it is difficult to protect the data privacy of wireless communication devices with conventional security methods. If the attacker succeeds in tapping the preamble, the confidentiality of the key can no longer be guaranteed. Therefore, in this study, a multi-channel-based dynamic key update technique was proposed to ensure the confidentiality of the key. Using the proposed model, the attack complexity increases in proportion to the length of the key, so the attack complexity increases at least 2^{52} times compared to the conventional model. Therefore, the proposed model can be used to increase the difficulty of brute-force attacks in environments where security is important. However, in this study, there is a limitation in that the security was evaluated without considering noise generated in the wireless network environment. Therefore, in future studies, we plan to evaluate the performance of the proposed model by considering noise generated in the actual wireless network environment.

Acknowledgments

This paper was conducted as a result of research on the Industrial Innovation Talent Growth Support Project (RS-2024-00415520) of the Ministry of Trade, Industry and Energy and the Korea Institute of Industrial Technology Promotion, and the ICT Innovation Talent 4.0 Project of the Ministry of Science and ICT and the Information and Communication Planning and Evaluation Institute in 2024 (No. IITP-2022-RS-2022-00156310)

References

1. Y. Leng, R. Zhang, W. Wen, P. Wu and M.Xia, "Physical-layer Authentication with Watermarked Preamble for Internet of Things," 2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Montreal, QC, Canada, 2023, pp. 212-217Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
2. Z. Zhang, H. Rahbari and M. Krunz, "Adaptive Preamble Embedding With MIMO to Support User-Defined Functionalities in WLANs," in IEEE Transactions on Mobile Computing, vol. 22, no. 2, pp. 691-707, 1 Feb. 2023, doi: 10.1109/TMC.2021.3095459
3. Z. Li et al., "Decomposed and Distributed Modulation to Achieve Secure Transmission," GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 2023, pp. 5378-5383,doi: 10.1109/GLOBECOM54140.2023.10437573.

The Paradox of Security Controls: Do Stricter Organizational Measures Improve Security Levels?

Jungjoo Oh[1] and Il-Gu Lee[1]

[1] Sungshin Women's University, 2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul, South Korea
winteroot@sungshin.ac.kr, iglee@sungshin.ac.kr

Abstract. While it may be thought that the enhanced security controls will improve organizational security levels, this is not necessarily the case. Strengthened security controls can cause stress for organization members, potentially resulting in Information Security Policy (ISP) violations. Based on security stress theory, this study discusses the impact of security stress factors on ISP violations.

Keywords: Security Stress, Information Security Policy (ISP), ISP Violation

1 Introduction

Does strengthening security controls enhance an organization's security level? While we might believe that the enhanced security controls will improve organizational security, these strengthened measures may not always lead to higher security levels. According to Proofpoint [1], organization members responded that they would prioritize security if it became easier (94%) or if rewards or recognition were offered (89%). In contrast, security experts believed that 'strengthening security training (83%)' and 'enhancing security controls (81%)' would effectively improve organizational security levels, showing a significant difference in views from organization members. Previous studies have examined why organization members violate security requirements from the perspective of security stress or fatigue [2-4]. According to these studies, organizational security requirements can induce stress, leading to violations of the Information Security Policy (ISP) by organization members.

Existing research has mainly focused on the coping responses of organization members due to security stress, limiting discussions on practical improvement directions. Additionally, discussions were based on specific aspects of security stress, such as security requirements or technologies or the attributes of security stress [4-6]. Among these, empirical analyses are few. Moreover, while not directly studied about security stress, there may be factors related to ISP violations that are associated with security stress. Therefore, this study restructures security stress factors according to security stress theory and discusses their relationship with ISP violation phenomena among organization members. The goal is to present relevant theoretical concepts based on literature review and propose a research model for future empirical analysis.

2 Literature Review

2.1 Security Stress

Stress occurs when an individual's resources are insufficient to cope with the demands and pressures of a situation [7]. Among these, the concept of security stress emerged when D'Arcy et al.[3] extended the concept of technostress to the field of information system security. Security stress arises as organization members strive to comply with ISP [8] and can lead to security fatigue, causing ISP violations [9]. D'Arcy et al. [3] categorized security stress into overload, complexity, and uncertainty. Too much security-related information and warning messages can induce security stress. Lee et al. [8] empirically proved that work overload can be a factor of security stress, particularly in organizations oriented towards technical security, where work overload has a greater impact on security stress. Security stress can also occur when security requirements are complex. Complexity can arise when the security technology used by organization members is difficult to understand or when the security procedures they must follow are long and complicated. Lastly, when the effects of complying with security requirements are uncertain, it can negatively stress organization members [3].

2.2 Antecedent Factors of ISP Violation

Security stress can be a precursor to ISP violations by organization members. Security stress can arise from various factors. It can be induced when security requirements are inconvenient, perceived as unreasonable or unfair, when security communication is excessive, or when the method of conveying security requirements is inappropriate [10]. Technical factors can also cause security stress. Security settings such as software updates and patches or website access blocking due to firewall configurations can cause inconvenience and fatigue for organization members in performing their tasks [2]. Frequent security threat warnings, account security alerts, regular password change notifications, and other recurring security-related alarms can also be factors of security stress [6, 10]. Additionally, although not directly studied about security stress, rewards and penalties for ISP violations can influence ISP compliance [11, 12].

2.3 ISP Violation

Previous studies have mainly focused on coping responses to understand why organization members violate ISP. Based on theories such as the Theory of Reasoned Action, Neutralization Theory, and Deterrence Theory, they emphasized the process from attitude to ISP violation behavior. Among these, studies on security stress and fatigue [2, 10, 13] revealed that excessive security requirements can lead to ISP violations by organization members. Olt & Mesbah [13] conceptualized security fatigue as a form of ego depletion and analyzed the characteristics of users who ignore security recommendations. According to their research, users who ignored security recommendations were those overwhelmed and exhausted by security-related information. Cram et al. [10] found that organization members may have reduced willingness to comply with ISP

when they need to make many decisions, which can result in ignoring or avoiding ISP or minimizing effort.

3 Research Model

3.1 ISP Violation Model

Previous studies [2, 10, 13] confirmed that organization members can intentionally or unintentionally violate ISP due to stress from organizational security requirements. However, there has been little comprehensive research on the factors constituting security stress. In particular, while factors such as ISP effectiveness and organizational rewards or sanctions may be related to security stress, they have mainly been studied about ISP compliance. According to neutralization theory, if the effect of an organization member's ISP compliance on the organization's security issues is unclear, the member may justify ISP violation behavior [14]. Previous studies have identified rewards and sanctions as effective ways to motivate organization members to comply with security [11, 12]. If these factors are insufficient or unsatisfactory, organization members may violate ISP, but before that, they may become dissatisfied or frustrated, leading to stress. Factors constituting security stress may have moderating effects on their relationships, potentially altering their impact on ISP compliance. In other words, systematic research, such as examining the relationship between security stress and ISP violations from a comprehensive perspective, is becoming increasingly important. Therefore, this study aims to reconstruct the factors constituting security stress from a comprehensive perspective and analyze the impact of these security stress factors on ISP violations.

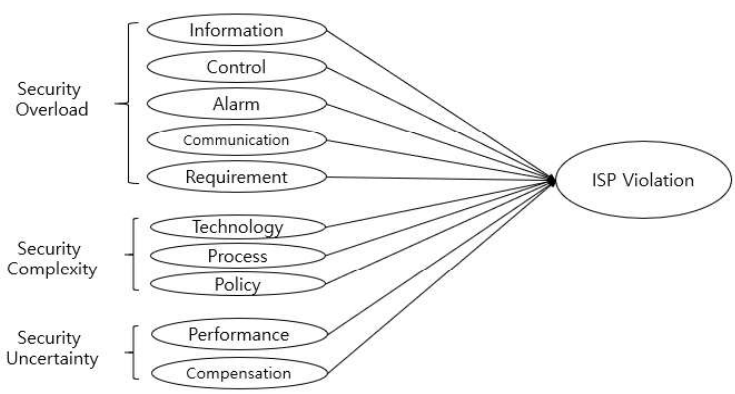


Fig. 1. Research Model

3.2 Security Stress Factors

The factors of the security stress model were reconstructed based on Nobles' (2022) [6] types of security fatigue. The reconstructed security stress factors are as follows:

Table 1. Security Stress Factors

Factors	Content
Information	Excess of information needed for security measures, news of security incidents, etc.
Control	Automatically performed security tasks, inconvenience in using work-related materials due to security measures, etc.
Alarm	Frequent security alerts, updates and patches, password change alarms, etc.
Communication	Frequent security-related emails, messages, security training, etc.
Requirement	Excessive number of security measures, excess of unnecessary or inefficient security requirements, etc.
Technology	New security technologies that are difficult to adapt to and complex, compatibility issues, etc.
Process	Complex authentication procedures, administrator approval procedures, etc.
Policy	Security regulations written in difficult-to-understand and complex terms etc.
Performance	The effect of employees' compliance with security regulations on solving security issues, etc.
Compensation	Provision of incentives for compliance with security regulations, whether it's reflected in performance evaluations, etc.

4 Conclusion

This study aims to reconstruct security stress factors based on a literature review and analyze their impact on intentions to violate the ISP (Information Security Policy). It focuses on the fact that in addition to technical factors, policy factors, and reward/punishment factors can also cause security stress, and these factors may influence ISP violations. This study has academic significance in reconstructing and refining stress models studied in specific dimensions. Practically, it is significant because it can suggest ways to minimize ISP violations by organizational members by empirically identifying security stress factors that directly affect ISP violations. For example, corporate security personnel can present security requirements focusing on factors with low psychological resistance among organizational members. This can reduce the phenomenon of organizational members rebelling against and circumventing ISPs, thereby increasing the effectiveness of ISPs.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Proofpoint: 2024 State of the Phish. (2024)
2. Furnell, S., Thomson, K.-L.: Recognising and addressing 'security fatigue'. Computer Fraud & Security 2009, 7-11 (2009)
3. D'Arcy, J., Herath, T., Shoss, M.K.: Understanding employee responses to stressful information security requirements: A coping perspective. Journal of management information systems 31, 285-318 (2014)
4. Chen, H., Liu, M., Lyu, T.: Understanding employees' information security–related stress and policy compliance intention: the roles of information security fatigue and psychological capital. Information & Computer Security 30, 751-770 (2022)
5. Parkin, S., Krol, K., Becker, I., Sasse, M.A.: Applying cognitive control modes to identify security fatigue hotspots. In: Twelfth Symposium on Usable Privacy and Security (SOUPS 2016). (Year)
6. Nobles, C.: Stress, burnout, and security fatigue in cybersecurity: A human factors problem. HOLISTICA–Journal of Business and Public Administration 13, 49-72 (2022)
7. Michie, S.: Causes and management of stress at work. Occupational and environmental medicine 59, 67-72 (2002)
8. Lee, C., Lee, C.C., Kim, S.: Understanding information security stress: Focusing on the type of information security compliance activity. Computers & Security 59, 60-70 (2016)
9. D'Arcy, J., Teh, P.-L.: Predicting employee information security policy compliance on a daily basis: The interplay of security-related stress, emotions, and neutralization. Information & Management 56, 103151 (2019)
10. Cram, W.A., Proudfoot, J.G., D'Arcy, J.: When enough is enough: Investigating the antecedents and consequences of information security fatigue. Information Systems Journal 31, 521-549 (2021)
11. Herath, T., Rao, H.R.: Protection motivation and deterrence: a framework for security policy compliance in organisations. European Journal of information systems 18, 106-125 (2009)
12. Bulgurcu, B., Cavusoglu, H., Benbasat, I.: Information security policy compliance: an empirical study of rationality-based beliefs and information security awareness. MIS quarterly 523-548 (2010)
13. Olt, C.M., Mesbah, N.: Weary of watching out?-cause and effect of security fatigue. (2019)
14. Siponen, M., Vance, A.: Neutralization: New insights into the problem of employee information systems security policy violations. MIS quarterly 487-502 (2010)

Optimized Implementation of SABER on resource-constrained 16-bit MSP430 MCU

MinGi Kim[1][0009−0008−4541−1088], DongHyun Shin[2][0009−0008−4847−5597], YoungBeom Kim[2][0000−0003−4715−8393], and Seog Chung Seo[1,2,*][0000−0001−8016−2808]

[1] Department of Information Security, Cryptology and Mathmatics, Kookmin University, Seoul, South Korea
[2] Department of Financial Information Security, Kookmin University, Seoul, South Korea
{mingi1684,dkldkl13,darania,scseo}@kookmin.ac.kr

Abstract. SABER is one of NIST PQC Round 3 KEM algorithms and is based on MLWR (Module Learning with Rounding) problem. Since the core operation of SABER is polynomial multiplication[1], accelerating it contributes to the overall performance improvement. Thus, in this paper, we present the first optimized implementation of SABER on resource-constrained 16-bit MSP430 MCU by optimizing the performance of the polynomial multiplication. In detail, to optimize the polynomial multiplication, we apply NTT(Number theoretic Transform)-based polynomial multiplication instead of the previously used Toom-Cook based one. For applying NTT, we found a optimal prime value q'(`0x18D 0001`) which not only satisfying the necessary conditions ($q' > nq^2$ and $q' \equiv 1 \bmod 2n$) for NTT, but also enabling efficient modular reduction computation due to its lower 16-bit (`0x0001`) on 16-bit MCUs. In addition, we optimize signed Montgomery modular reduction, heavily used on NTT-based polynomial multiplication, by taking full advantage of both MSP430's hardware multiplier and the newly chosen prime q'. The process of NTT and inverse NTT conversion is also optimized by merging some layers by fully utilizing registers, which saves a number of memory accesses. With the proposed optimization techniques, our NTT-based method outperforms reference implementation by 102% in polynomial multiplication. Moreover, our proposed implementation in SABER-PKE achieves speed improvements of 57% in Key Generation, 44% in Encapsulation, and 52% in Decapsulataion compared to the reference implementation.

Keywords: Post-Quantum Cryptography · MSP430 · NTT · Optimize Implementation · Saber

This work was supported by the Institute of Information and communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through Ministry of Science and ICT (MSIT) (A study on PQC optimization and security protocol migration to neutralize advanced quantum attacks in Beyond 5G-based next-generation IoT computing environments, 100%) under Grant 2022-00207416.

1 Introduction

The advancement of quantum computers threatens traditional public key cryptosystems, leading to active research in quantum-resistant cryptography. Various cryptosystems based on complex mathematical problems have been proposed, among which lattice-based cryptosystems have demonstrated superior performance and have been widely adopted. For example, out of the four final algorithms selected in the NIST competition, three are lattice-based, and among the eight second-round candidates in the domestic KpqC competition, four are lattice-based cryptosystems.

Lattice-based cryptosystems predominantly utilize the Number Theoretic Transform (NTT) algorithm to address the substantial computational load of polynomial multiplication. However, there exist lattice-based cryptosystems that are NTT-unfriendly, meaning they cannot utilize the NTT transform. Significant research efforts have been dedicated to enabling the use of NTT in such cryptosystems. Nevertheless, these optimization studies have predominantly focused on devices like Cortex-M4[2], M3[3], and AVX2, leaving low-specification devices relatively unexplored. Therefore, we provide the first implementation and optimization results for the MSP430, a low-specification device operating in a 16-bit environment.

2 Preliminaries

2.1 NTT

The Number Theoretic Transform (NTT) is a specialized form of the Discrete Fourier Transform (DFT) defined over finite fields. In finite fields like $F = Z/q$, primitive n-th roots of unity, ω_n, exist if n divides $p-1$. For efficient application in lattice schemes using the polynomial ring $R[x] = Z_q/f(x)$, $f(x)$ is typically chosen as $f(x) = X^n + 1$ in schemes like Kyber and Dilithium. Using the Chinese Remainder Theorem (CRT), elements in $R[x]$ are transformed into n integers, and their multiplication complexity is $\mathcal{O}(n \cdot \log n)$.

2.2 MSP430

The MSP430 is an ultra-low power Micro Controller Unit (MCU) designed by Texas Instruments, based on a 16-bit RISC CPU [4]. The MSP430 features 16 registers numbered `R0-R15`. Among these, `R0-R3` are reserved for special purposes: program counter, stack pointer, status register, and constant generator, respectively. The remaining registers, `R4-R15`, are general-purpose registers that can be utilized for various applications. The MSP430 is equipped with a hardware multiplier that supports both 16-bit and 32-bit multiplications. These multiplications are executed by writing values to dedicated hardware multiplier registers.

3 NTT Optimization on MSP430

To enable the use of the NTT transform in Saber, we select a new prime number $q' = 26017793$. This prime is greater than nq^2, thereby preserving the values of the original multiplication operations, and since $q' \equiv 1 \mod 2n$, it allows for the execution of the complete-NTT transform. Given that q' is a 25-bit number, the coefficients have to perform 32-bit operations.

3.1 Montgomery Reduction using q'

To perform the 32-bit NTT mentioned earlier, we need to compute a 32-bit modular reduction. In this paper, we use signed Montgomery reduction arithmetic. The new prime number $q' = 26017793$, which we selected to apply the NTT in Saber, has a lower 16-bit value of `0x0001`. This provides a significant advantage when performing Montgomery reduction. Algorithm 1 takes the product of two numbers a as input and performs modular reduction. First, the lower 32-bit of a is multiplied by q^{-1}. Since the lower 16-bit of our selected prime is `0x0001`, this can be processed with one 16-bit multiplication instead of a 32-bit multiplication. As proven in Sec. 3.1 of [5], signed Montgomery reduction efficiently computes the result by only calculating the signed high product, unlike traditional Montgomery reduction. Previously, this required a left-shift operation, but the MSP430 lacks a barrel shifter capable of moving multiple bits simultaneously. Therefore, performing a 32-bit shift would require a register the size of the coefficient and 32 instructions. By using signed Montgomery reduction, this paper achieves an efficient implementation. Algorithm 2 presents the code for signed Montgomery multiplication on the MSP430.

Algorithm 1 Signed Montgomery Reduction [5]

Require: $0 < q < \frac{\beta}{2}$ odd, $-\frac{\beta}{2}q \leq a = a_1\beta + a_0 < \frac{\beta}{2}q$ where $0 \leq a_0 < \beta$
Ensure: $r' \equiv \beta^{-1}a \pmod{q}, -q < r' < q$
1: $m \leftarrow a_0 q^{-1} \mod \pm\beta$ ▷ signed low product, q^{-1} precomputed
2: $t_1 \leftarrow \left\lfloor \frac{mq}{\beta} \right\rfloor$ ▷ signed high product
3: $r' \leftarrow a_1 - t_1$

3.2 NTT merging

In assembly language implementations, unlike in high-level languages such as C, research has been continuously conducted to minimize memory access during NTT transformations [6]. Each embedded device has a different number of general-purpose registers and coefficient sizes, necessitating efforts to find the optimal layer merging strategy. Generally, to merge n NTT layers, 2^n coefficients need to be stored in registers. Additionally, since our q' is 25-bit, two registers are required to store one coefficient on the MSP430. Currently, when performing NTT transformations, we reserve 4 out of the 12 general-purpose registers for the polynomial pointer, loop counter for iterations, and two registers for butterfly operations in NTT. Consequently, 8 registers are available for layer merging, leading to the conclusion that merging two layers at a time is the optimal approach. Therefore, our implementation is designed with the layer merging strategy of 8-7, 6-5, 4-3, and 2-1 layers.

4 Result

We benchmarked the 16-bit MSP430 environment using the IAR Embedded Workbench 9.1 simulator. We utilized the IAR EW compiler, with the optimization level set to high (speed). Since there is no existing implementation for

Algorithm 2 Signed Montgomery Multiplication on MSP430

Input: 32-bit $a = (a_1||a_0),\ b = (b_1||b_0)$
Output: $c = abR^{-1} \mod q$, where $R = 2^{32}$

```
1: mov  a0,       &MPYS32L
2: mov  a1,       &MPYS32H
3: mov  b0,       &OP2L
4: mov  b1,       &OP2H                          ▷ multiply a, b
5: mov  &RES0,    &MPYS
6: mov  &RES1,    &MPYS32H
7: mov  &RES2,    c0
8: mov  &RES3,    c1                             ▷ store multiplication result
9: mov  #0xFE73, &OP2
10: add  &RESLO,  &MPYS32H                       ▷ signed low product
11: mov  #0x0001, &OP2L
12: mov  #0x018D, &OP2H                          ▷ signed high product
13: sub  &RES2,   c0
14: subc &RES2,   c1
15: return c = (c1||c0)
```

a 16-bit environment, we compared the reference C code [7] and the C code with NTT. Additionally, we compared these implementations with our assembly implementation. Table 1 shows the results. Firstly, for 1-module multiplication, the NTT-based multiplication achieved a 16% performance improvement over the Reference implementation using Toom-Cook multiplication, while our implementation achieved a 102% performance improvement. Furthermore, in the entire Saber KEM process, our implementation achieved up to 57% (KeyGen), 44% (Encaps), and 52% (Decaps) performance improvements compared to the reference implementation. Compared to the NTT-based C implementation, our work achieved up to 35%, 37%, and 38% performance improvements, respectively.

Table 1. Performance Results in Clock Cycles on the MSP430

	1-module MUL	NTT	NTT^{-1}		Lightsaber	Saber	Firesaber
Ref (TC4) (C)	568K	-	-	K	4,430K	9,062K	15,114K
				E	6,307K	11,777K	18,578K
				D	6,917K	12,692K	19,890K
32 NTT (C)	490K	137K	170K	K	3,825K	7,485K	12,119K
				E	5,835K	10,833K	17,004K
				D	6,291K	11,516K	18,005K
This work (ASM)	281K (+102%) (+74%)	71K (+92%)	89K (+91%)	K	3,124K	6,007K	9,579K
				E	4,589K	8,340K	12,850K
				D	4,629K	8,400K	13,020K

References

1. LADS Ribeiro, José Paulo da Silva Lima, Ruy JGB de Queiroz, Amirton Bezerra, Jonysberg P Chagas, Fabio QB da Silva, Andre LM Santos, José Roberto, and Ribeiro Junior. Saber post-quantum key encapsulation mechanism (kem): Evaluating performance in mobile devices and suggesting some improvements. In *Third PQC Standardization Conference*, pages 07–09, 2021.
2. Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. NTT multiplication for NTT-unfriendly rings. Cryptology ePrint Archive, Paper 2020/1397, 2020. https://eprint.iacr.org/2020/1397.
3. Amin Abdulrahman, Jiun-Peng Chen, Yu-Jia Chen, Vincent Hwang, Matthias J. Kannwischer, and Bo-Yin Yang. Multi-moduli NTTs for saber on cortex-m3 and cortex-m4. Cryptology ePrint Archive, Paper 2021/995, 2021. https://eprint.iacr.org/2021/995.
4. T. Instruments. *MSP430x5xx and MSP430x6xx Family User's Guide*. Texas Instruments, 2023. https://www.ti.com/lit/ug/slau208q/slau208q.pdf?ts=1703730297957&refurl=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430F5510%253FHQS%253Dti-null-null-verifimanuf-manuf-manu-pf-octopart-wwe.
5. Gregor Seiler. Faster AVX2 optimized NTT multiplication for ring-LWE lattice cryptography. Cryptology ePrint Archive, Paper 2018/039, 2018. https://eprint.iacr.org/2018/039.
6. Erdem Alkim, Yusuf Alper Bilgin, Murat Cenk, and François Gérard. Cortex-m4 optimizations for {R,M}LWE schemes. Cryptology ePrint Archive, Paper 2020/012, 2020. https://eprint.iacr.org/2020/012.
7. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. *SABER. Submission to the NIST Post-Quantum Cryptography Standardization Project*, 2020. https://www.esat.kuleuven.be/cosic/pqcrypto/saber/.

TrustMark, can I trust you?

Su-Kyoung Kim, Yu-Ran Jeon, Jung-Hwa Ryu, Il-Gu Lee

Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
{20220115, 220247016, 220236039, iglee}@sungshin.ac.kr

Abstract. Recently, generative AI (Artificial Intelligence) technologies have been widely utilized across all industrial sectors. However, content created by generative AI has led to issues such as copyright infringement and misuse. To prevent these adverse effects, digital watermarking technologies have been employed to protect digital content. However, conventional digital watermarking technologies have not internalized security in the design and implementation of watermarks, and robustness evaluations against geometric attacks have not been considered. This study evaluates the robustness of TrustMark, a generative AI-based watermarking technology, and suggests directions for future research.

Digital watermark, Watermark attack, AI security

1 Introduction

Generative artificial intelligence (AI) is a technology that utilizes large datasets to create content such as conversations, images, videos, and music. While this technology has brought about technological innovations across various industries and changes in everyday life on a socio-cultural level, it has also resulted in adverse effects such as copyright infringement, the spread of misinformation, and deepfake crimes. To mitigate these side effects, digital watermarking technology is being employed. Digital watermarking involves embedding unique identification information into digital content to identify the outputs of generative AI. However, watermarks can be vulnerable to geometric attacks such as resizing, rotation, and compression, making robustness evaluation against these attacks essential. This study examines TrustMark, a generative AI-based digital watermarking technology, and evaluates the robustness of embedded digital watermarks by subjecting them to various attacks. Through this evaluation, we aim to verify whether digital watermarking technology can effectively function as a tool for copyright protection and maintaining trustworthiness.

2 Related Works

Adobe's TrustMark [1] is a GAN (Generative Adversarial Network) based watermarking technology. This technology applies a spatio-spectral loss design to balance image quality and watermark detection accuracy, demonstrating robustness against noise and resolution change attacks. Adobe maintains a high market share of 79.3% in the multimedia design and engineering software market [2]. By incorporating AI technologies,

Adobe enhances the accessibility of various software and editing programs. In the increasing demand for AI-generated digital content, advanced watermarking technology for digital content protection and authentication becomes essential. Adobe's TrustMark is expected to play a crucial role in ensuring the reliability of digital content. Based on this background, this study selected TrustMark as a representative digital watermarking technology and conducted a robustness evaluation.

3 Robustness evaluation of TrustMark

In this study, we evaluate the robustness of the TrustMark watermarking technology. Using the TrustMark open-source code [3], we set up the watermarking environment and assessed performance by applying geometric transformations such as JPEG compression, resizing, blurring, rotation, and cropping. Attacks were performed on a single 240x240 pixel image with an embedded watermark, and the watermark detection results were verified. To assess the degree of image quality degradation, the PSNR (Peak Signal-to-Noise Ratio) was measured according to the intensity of each attack. PSNR can be calculated using Equation (1), where MSE is determined by Equation (2).

$$PSNR = 10log_{10}\left(\frac{256^2}{MSE}\right)[dB] \quad (1)$$

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(D_i - O_i)^2 \quad (2)$$

In these equations, n represents the total number of pixels in the image, D_i is the value of the i-th pixel in the distorted image, and O_i is the value of the i -th pixel in the original image with the embedded watermark. MSE (Mean Squared Error) measures the average squared difference between the original and distorted images, so a higher PSNR indicates a higher similarity between the two images.

Table. 1. Evaluation of performance based on the strength of each geometric attack

Criteria	JPEG compression								
	50%	45%	40%	35%	30%	25%	20%	15%	10%
PSNR (dB)	31.743	30.258	29.011	29.519	29.010	28.442	27.493	26.707	25.422
Detection	Success	Success	Success	Success	Success	Failure	Failure	Failure	Failure
Criteria	Downsizing								
	98%	96%	94%	92%	90%	88%	86%	84%	82%
PSNR (dB)	45.107	42.127	45.734	46.354	46.550	46.959	46.896	47.368	47.327

Detection	Success	Success	Success	Success	Success	Failure	Failure	Failure	Failure
Criteria	Blurring								
	1σ	1.5σ	2σ	2.5σ	3σ	3.5σ	4σ	4.5σ	5σ
PSNR (dB)	27.631	24.914	23.628	22.713	22.158	21.491	21.016	20.579	20.195
Detection	Success	Success	Success	Success	Success	Failure	Failure	Failure	Failure
Criteria	Rotation								
	1º	1.5º	2º	2.5º	3º	3.5º	4º	4.5º	5º
PSNR (dB)	25.528	24.836	22.629	21.853	21.458	21.31	20.553	20.312	20.02
Detection	Success	Success	Success	Success	Failure	Failure	Failure	Failure	Failure
Criteria	Crop								
	4%	8%	12%	16%	20%	24%	28%	32%	36%
PSNR (dB)	26.234	25.4	25.133	24.34	24.245	23.289	23.767	22.93	21.109
Detection rate	100%	100%	100%	100%	100%	85.6%	84.8%	83.5%	61.2%

Table 1 shows the performance of TrustMark based on the intensity of each geometric attack. According to the experimental results, TrustMark failed to detect the watermark when JPEG compression was at or below 25%, resizing was at or below 88%, blurring was at or above 3.5σ, or rotation was at or above 3º.

When a cropping attack was performed, the watermark was detected after cutting the image to retain only the specified image area. The detection varied depending on the cropping location even when cropping to the same size. Therefore, the detection was carried out at all possible pixel positions within the image, and the success rate was calculated by comparing the number of successful detections to the total number of detection attempts. The evaluation results showed that the watermark was detected when 20% or less of the image area was cropped. However, when 24% or more of the image area was cropped, the detection success rate dropped to 85% or less. This indicates that the larger the cropped image area, the lower TrustMark's watermark detection accuracy.

In cases where detection failed, the PSNR values were significantly higher for resizing attacks, followed by JPEG compression, blurring, cropping, and rotation attacks. This demonstrates that resizing attacks had the most significant impact on the robustness of TrustMark.

4

4 Conclusion

In this paper, we reviewed the latest watermarking technologies to mitigate the adverse effects of generative AI and evaluated their robustness against watermark distortion attacks. The evaluation results indicated that the watermark could not be detected under the following conditions: JPEG compression at or below 25%, resizing at or below 88%, blurring at or above 3.5σ, rotation at or above 3°, and cropping 24% or more of the image. Additionally, considering the degree of image quality degradation, resizing attacks posed the highest risk to the robustness of TrustMark.
Future research will focus on evaluating the robustness of watermarking technologies across various types of images and developing high-reliability digital watermarking techniques that are resilient to geometric transformations.

Acknowledgments

This paper was conducted as a result of research on the Industrial Innovation Talent Growth Support Project (RS-2024-00415520) of the Ministry of Trade, Industry and energy and the Korea Institute of Industrial Technology Promotion, and the ICT Innovation Talent 4.0 Project of the Ministry of Science and ICT and the Information and Communication Planning and Evaluation Institute in 2024 (No. IITP-2022-RS-2022-00156310)

References

[1] Bui, Tu, Shruti Agarwal, and John Collomosse, "TrustMark: Universal Watermarking for Arbitrary Resolution Images," arXiv preprint arXiv:2311.18297,2023.
[2] J. Yoo and C. Kim, "Expanding AI Ecosystem, Expecting Stable Long-term Revenue Growth," KB Securities, 2023.
[3] Adobe, https://github.com/adobe/trustmark.git. “trustmark,”2023.

Partial Homomorphic Encryption for Privacy-Preserving Data Learning

Hye-Yeon Shim[1] [0009-0009-4978-2553], Yu-Ran Jeon[1] [0009-0001-4149-9288] and Il-Gu Lee[1] [0000-0002-5777-4029]

[1] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
220237062@sungshin.ac.kr, 220247016@sungshin.ac.kr, iglee@sungshin.ac.kr

Abstract. With the recent advancement of artificial intelligence technology, the number of services that converge on machine learning and big data has increased, and the risk of personal information leakage due to indiscriminate data collection and learning has increased. Therefore, a technology that can perform machine learning while protecting privacy is essential. This study proposes a technique that can perform machine learning with partially homomorphic encrypted datasets while maintaining the confidentiality of data subjects' personal information. Experimental results show that the proposed technique can improve the prediction accuracy by 59.4% and the training time by 63.6% compared to the conventional technique.

Keywords: Homomorphic Encryption, Machine Learning, Privacy-Preserving, Logistic Regression.

1 Introduction

With the advent of the Fourth Industrial Revolution, the number of services that combine big data and machine learning is increasing rapidly. In particular, in fields closely linked to users, such as healthcare and finance, learning using user information has made it possible to provide better quality services than before [1, 2]. However, due to the problem that personal information may be exposed to the server during the machine learning process, there is a growing concern about privacy leakage of data subjects [3]. To address these issues, studies have been conducted on learning techniques that utilize differential privacy to protect data while learning. However, differential privacy suffers from a trade-off between utility and privacy [4]. Another method for protecting data is cryptographic algorithms such as advanced encryption standard (AES) and Rivest Shamir Adleman (RSA), which have limitations that prevent them from being used for machine learning.

Homomorphic encryption techniques are being researched to address this utility-privacy trade-off to enable privacy-preserving data learning. Homomorphic encryption allows arithmetic operations such as addition and multiplication to be performed while encrypted, and machine learning that requires iterative operations can be performed [5].

However, encrypting information with homomorphic encryption causes the size of the ciphertext to be excessively large compared to the original information [6]. In addition, noise increases as the number of repeated ciphertext operations increases, so the larger the epoch of machine learning, the larger the noise contained in the ciphertext becomes [7]. The increased noise due to repeated computations affects the updating of weight and bias during model training, causing a decrease in model accuracy.

To address the speed and accuracy issues in machine learning when utilizing homomorphic encryption for privacy-preserving data learning, this paper proposes a technique to perform logistic regression learning on a dataset where some features used for training are encrypted with homomorphic encryption. The contributions of this paper are as follows:

- A partial encryption-based high-efficiency homomorphic encryption technique for privacy-preserving data learning is proposed.
- A framework for evaluating the performance of conventional homomorphic encryption techniques and partial encryption-based homomorphic encryption techniques is proposed.

This paper is structured as follows. In Section 2, related work is compared and analyzed, and in Section 3, a logistic regression learning technique for partially encrypted datasets is proposed. Section 4 describes the experimental environment and settings and analyzes the experimental results. Finally, in Section 5, conclusions are drawn.

2 Related work

In this section, the contributions and limitations of representative prior work on highly efficient machine-learning techniques for homomorphically encrypted datasets are compared and analyzed.

Sgaglione et al. [8] inspected signature-based data to successfully detect attacks embedded in homomorphically encrypted information. However, the time complexity was not optimized, which makes it unsuitable for real-world services.

Spathoulas et al. [9] proposed a homomorphic computation-based clustering model that utilizes alerts collected from intrusion detection systems. The proposed model improves the processing efficiency of homomorphic encryption operations by applying a cooperative computation method in which clients perform operations that the server cannot handle. However, there is a limitation that the amount of communication between the server and clients increases.

Alabdulatif et al. [10] proposed a fully homomorphic encryption-based machine learning technique for smart city technologies. The proposed model utilizes cloud computing resources to perform homomorphic encryption operations in parallel to reduce the model training time. However, the parallel operation has the limitation of increasing CPU and memory overhead.

3 Proposed technique

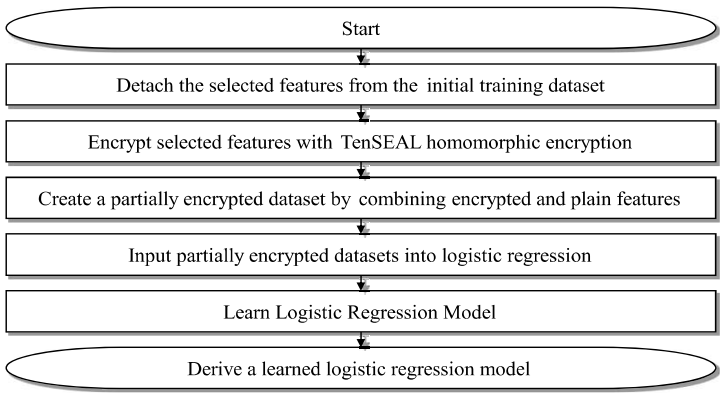


Fig. 1. Flowchart of the operation of the proposed technique.

This section proposes a technique to generate a predictive model using a training dataset where some features are encrypted with homomorphic encryption. In the proposed technique, the homomorphic encryption uses TenSEAL [12], a machine-learnable fully homomorphic encryption algorithm based on Microsoft's SEAL [13], and the simplest model is a logistic regression-based machine-learning model.

The overall operation flow of the proposed technique is shown in Fig 1. First, distinguish which features in the initial training dataset D are privacy-critical or confidential and which are not, as shown in Fig 2(a). The set of separated features that require confidentiality is encrypted using the homomorphic encryption TenSEAL to form the encrypted dataset D_{enc}, shown in Fig 2(b). Conversely, the set of features that do not require confidentiality among the separated feature set is separated into a plain dataset, D_{plain}, as shown in Fig 2(c). D_{enc} and D_{plain} are then combined to create a partially encrypted dataset D_{part}, as shown in Fig 2(d). Then, train a logistic regression model using D_{part} to generate a model that can predict labels.

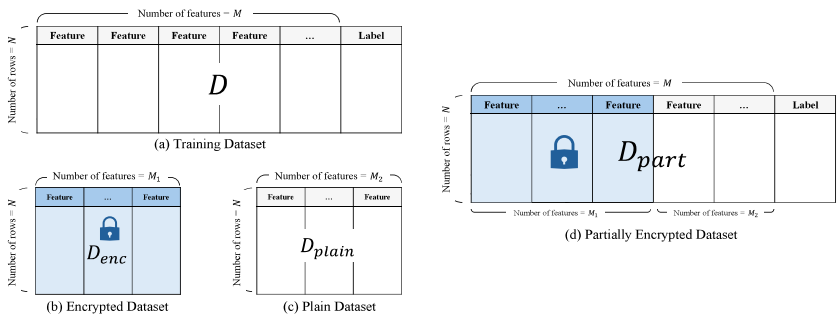


Fig. 2. Training dataset structure for the proposed technique.

4 Performance evaluation and analytics

Table 1. Performance evaluation for fully encrypted and partially encrypted datasets.

	Prediction accuracy (%)	Time (s)		Memory usage (MB)	
		Dataset encryption	Training	Dataset encryption	Training
Conventional	61.83	15.03	706.52	1,985.32	1.63
Proposed	98.56	11.67	257.13	1,714.50	0.05

In the experiments, the proposed technique with partial encryption is compared to the conventional model with full homomorphic encryption. The dataset used in the experiments is CIC-IDS2017, and 2000 rows with uniform label distribution were extracted. The performance metrics of the experiment are memory usage, model prediction accuracy, and latency, and the performance was measured over 100 iterations. The features encrypted with homomorphic encryption are 5, and the epoch of logistic regression training is 5.

Table 1 shows the performance comparison between the proposed and conventional techniques when the entire dataset is encrypted with homomorphic encryption. The experimental results show that the proposed technique achieves 59.4% higher prediction accuracy using the partially encrypted dataset than the conventional technique using the fully encrypted dataset. In addition, the time taken to generate the dataset for training is 22.36% less than the conventional technique. In particular, the proposed technique reduces the model training time by 63.61% compared to the conventional technique. Regarding memory usage, the proposed technique consumes less memory when encrypting the training data and when training the logistic regression model. Overall, the proposed technique consistently outperforms the conventional technique.

5 Conclusion

This paper proposes a technique for learning a logistic regression model from a partially encrypted dataset. Experimental results show that the proposed technique improves the prediction accuracy by about 59.4% and reduces the training time by about 63.61% compared to the case where the entire dataset is encrypted with homomorphic encryption. It also shows that the data encryption time and memory usage can be reduced.

The proposed technique was implemented through a logistic regression model, and the experiments were conducted under the assumption of a simple situation. Hence, the effectiveness of the proposed technique was not demonstrated. In future work, the proposed machine learning technique using partially encrypted datasets will be applied to deep learning, such as DNN and CNN, to demonstrate its effectiveness in complex models.

Acknowledgments. This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE) under the Training Industrial Security Specialist for High-Tech Industry (RS-2024-

00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning and Evaluation (IITP).

References

1. Shehab, M., Abualigah, L., Shambour, Q., Abu-Hashem, M. A., Shambour, M. K. Y., Alsalibi, A. I., Gandomi, A. H.: Machine learning in medical applications: A review of state-of-the-art methods. Computers in Biology and Medicine, 145 (2022)
2. Kumar, S., Sharma, D., Rao, S., Lim, W. M., Mangla, S. K.: Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research. Annals of Operations Research, 1-44 (2022)
3. Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., Lin, Z.: When machine learning meets privacy: A survey and outlook. ACM Computing Surveys (CSUR), 54(2), 1-36 (2021)
4. Rigaki, M., Garcia, S.: A survey of privacy attacks in machine learning. ACM Computing Surveys, 56(4), 1-34 (2023)
5. Podschwadt, R., Ghazvinian, P., GhasemiGol, M., Takabi, D.: Memory Efficient Privacy-Preserving Machine Learning Based on Homomorphic Encryption. In: International Conference on Applied Cryptography and Network Security (ACNS), pp. 313-339. Springer Nature Switzerland, Abu Dhabi, United Arab Emirates (2024)
6. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.: Homomorphic encryption standard. Protecting privacy through homomorphic encryption, 31-62 (2021)
7. Pulido-Gaytan, L. B., Tchernykh, A., Cortés-Mendoza, J. M., Babenko, M., Radchenko, G.: A survey on privacy-preserving machine learning with fully homomorphic encryption. Latin American High Performance Computing Conference, pp. 115-129. Springer International Publishing, Cuenca, Ecuador (2020)
8. Sgaglione, L., Coppolino, L., D'Antonio, S., Mazzeo, G., Romano, L., Cotroneo, D., Scognamiglio, A.: Privacy preserving Intrusion Detection via Homomorphic Encryption. 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 321-326. IEEE, Napoli, Italy (2019)
9. Spathoulas, G., Theodoridis, G., Damiris, G.-P.: Using homomorphic encryption for privacy-preserving clustering of intrusion detection alerts. International Journal of Information Security, 20(3), 347-370 (2021)
10. Alabdulatif, A., Khalil, I., Kumarage, H., Zomaya, A. Y., Yi, X.: Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities. Journal of Parallel and Distributed Computing, 127, 209-223 (2019)
11. Benaissa, A., Retiat, B., Cebere, B., Belfedhal, A. E.: Tenseal: A library for encrypted tensor operations using homomorphic encryption. arXiv preprint arXiv:2104.03152 (2021)
12. Chen, H., Gilad-Bachrach, R., Han, K., Huang, Z., Jalali, A., Laine, K., Lauter, K.: Logistic regression over encrypted data from fully homomorphic encryption. BMC medical genomics 11, 3-12 (2018)

ML-DSA Implicit certificates: PQC Migration for IEEE 1609.2 Standard

YoonSun Han[1][0009−0004−4550−5544] and Dongmin Kim[2][0000−0003−3861−2443] and Seog Chung Seo[1,*][0000−0001−8016−2808]

[1] Department of Financial Information Security, Kookmin University, Seoul, South Korea
[2] Mobile eXperience center, Samsung Electronics
{rlagksdbs12,scseo}@kookmin.ac.kr, kkomang03@korea.ac.kr

Abstract. For a secure and efficient Vehicle to Everything (V2X) transportation systems, a trust mechanism based on certificates is essential. IEEE 1609.2 is one of the security standard for V2X communications and defines the Security Credential Management System (SCMS) that uses Elliptic Curve Digital Signature Algorithm-P256 (ECDSA) as the standard algorithm. Research on migrating Post Quantum Cryptography (PQC) to SCMS using Butterfly Key Expansion(BKE) and Implicit certificates is just beginning. In this paper, we propose a secure certificate format and issuance process even in a pure-PQC environment, and present a ML-DSA based Implicit certificate (`MDIC`) for the first time. Through the analysis of security and performance, we show that the proposed certificate system is efficient regarding computation, and secure against the the corrupted Certificate Authority (CA)

Keywords: V2X · IEEE 1609.2 · PQC · Implicit certificates · Privacy

1 Introdution

Connected vehicles for intelligent transportation systems are one of the security requirements for autonomous driving systems [1]. In an intelligent transportation system, vehicles and Road Side Units (RSUs) that communicate with V2X Vehicle to Evrything (V2X) exchange traffic information through broadcast. For trust in communication, Basic Safety Message (BSM) is authenticated based on certificates. IEEE 1609.2, one of the V2X security standard, uses discrete logarithm-based digital signature algorithms such as Elliptic Curve Digital Signature Algorithm-P256 (ECDSA-P256) and ECDSA-P384.

National Institute of Standards and Technology (NIST) selected for standardization three PQC-DSA, Crystals-Dilithium (Dilithium), Falcon, and SPHINCS+ as alternatives to quantum vulnerabilities in July 2022. Accordingly, research on achieving quantum security in V2X communication is actively being conducted [2] [3] [4]. [2] proposed an ECDSA-PQC hybrid model that satisfies the IEEE 1609.2 standard data frame of 2,304 bytes. [3] proposed the Unified Butterfly Key (UBK) by setting a set of parameters for the qTELSA algorithm, and [4]

showed that designing an implicit certificate for Dilithium based on M-LWE is difficult. In this paper, we propose a pure-PQC technique for environments that will not allow existing classical cryptography after the 2030s. The M-LWE-based Implicit Certificates for efficient communication in V2V is first proposed, and the safety and performance of the proposed technique are analyzed.

2 Preliminaries

2.1 Security Credential Management System (SCMS)

In V2X communication, vehicles broadcast their certificates in real time, which can cause privacy issues. The SCMS standard issues pseudonym certificates to protect it. The anonymity of the vehicle is guaranteed throughout the issuance and use process and the cost of certificate issuance is reduced using Butterfly Key Expansion (BKE) [5] in the issuance process. For ML-DSA Implicit Certificates, we consider three entities, **Vehicle**, Registration Authority (**RA**), and Pseudonym Certificate Authority (**PCA**), to issue pseudonym certificates, and each entity performs the following roles. **Vehicle**: The entity requesting issuance of a certificate, and receives a valid pseudonym certificate from the PCA through communication with the RA. **RA**: Responsible for BKE, which expands each key received from multiple vehicles. After key expansion, all keys are shuffled to prevent PCA from knowing the source of the key. RA cannot know the contents of the pseudonym certificate issued to the vehicle. **PCA**: Responsible for creating pseudonym certificates using the key received from RA, and information about the vehicle that owns the pseudonym certificate is unknown.

2.2 ML-DSA

Federal Information Processing Standards Publication 204 (FIPS 204) standard Module-Lattice-Based DSA (ML-DSA) [6], defined in August 2023, uses the secret vector $\mathtt{s}$, noise vector $\mathtt{e}$ samped from rejection sampling as the secret key($\mathtt{sk}$). The public key($\mathtt{pk}$) corresponding to $\mathtt{sk}$ generated by computing $\mathtt{t} \leftarrow \mathtt{As} + \mathtt{e} \bmod q$ on public matrix $\mathtt{A}$. In the $\mathtt{pk}$ $\mathtt{t}$ of ($\mathbf{A}$, $\mathtt{t}$), $\mathtt{s}$ and $\mathtt{e}$ are protected by the M-LWE Problem.

ML-DSA' signature consists of the element $\mathtt{z} \leftarrow \mathtt{y} + \mathtt{cs}$, where $\mathtt{y}, \mathtt{s} \in R_q^{\ell}$ and $\mathtt{c} \leftarrow Hash[HighBits[\mathtt{Ay}]\mathtt{m}]$ on the message $\mathtt{m}$. Polynomial vector $\mathtt{y}$ is pseudorandomly sampled, s is also pseudorandomly sampled from the subset of polynomial vectors whose coeffocients are in range $[-\eta, \eta]$, $\mathtt{z}$ is at most $\gamma_1 - \beta$ (e.g., ML-DSA-65= $2^{19} - 196$). ML-DSA return $\mathtt{<c, z>}$, which is the notable signature result part. ML-DSA verify produce as output Boolean with respect to $Hash[HighBits[\mathtt{Az} - \mathtt{ct}]\mathtt{m}] \stackrel{?}{=} \mathtt{c}$.

3 ML-DSA based SCMS

Bindel et al. attempted to generate Implicit Certificates for Dilithium based on M-LWE [4]. The proposed Restructed Value (RV) is to include the CA's digital signature in the user sk. However, it has been revealed that RV with respect to dilithium is difficult because the maximum signature size is larger than the coefficient range of s and e ($\eta = 2$ or $\eta = 4$). In this paper, we propose M-LWE-based Implicit certificates MDIC format that can be used in V2X communication to achieve quantum resistance and method for Vehicle to issue a certificate from the PCA. Table 1. shows Vehicle → RA → PCA → Vehicle, which is the core of the issuance process. For MDIC, ML-DSA-65 with η=4 is used, and each entity in SCMS generates secret key(sk) coefficients of at most η=1. The confidentiality of the private key is guaranteed based on M-LWE [7]. sk with an η range of 1 are not used for digital signatures, and only the final generated key with η=4 is used for signing based on ML-DSA-65 standard.

Vehicle → RA: Vehicle transmits the pk generated with the caterpillar key $\mathtt{t}_i \leftarrow \mathtt{A}\mathtt{s}_V + \mathtt{e}_V$ and the seed ρ for synchronization with Vehicle to RA.

RA→PCA: For BKE, $\mathtt{s}_i$ and $\mathtt{e}_i$ are sampled from rejection sampling, $\mathtt{t}_i \leftarrow \mathtt{t}_V + \mathtt{A}\mathtt{s}_i + \mathtt{e}_i$ is calculated, and then all keys are shuffled to become a pk with η=2 by additive homomorphism.

PCA→Vehicle: For RV, sample s and e at η=1 and add the long-term key at η=1 to create sk with a coefficient range of η=2.

As shown in Table 1. $\mathtt{t}_{rv}$ is the pk U_i in MDIC, and is used as RV. To create u_i, the sk required for signing, Vehicle creates $(\mathtt{s}_{rv}, \mathtt{e}_{rv})$ by adding $(\mathtt{s}_{rand}, \mathtt{e}_{rand})$ received from PCA to $(\mathtt{s}_V + \mathtt{s}_i, \mathtt{e}_V + \mathtt{e}_i)$. u_i is a sk at η=4. Vehicle checks correctness by comparing whether the sum of t_{PCA} that is **PCA**'s pk and t_{rv} is equal to the $U_i \leftarrow \mathtt{A}\mathtt{s}_{rv} + \mathtt{e}_{rv}$.

4 Analysis

4.1 Security analysis

The safety of the ML-DSA-based SCMS can be achieved by checking the confidentiality of the sk of the **Vehicle** and **PCA** and whether anonymity is guaranteed during the issuance process. MDIC guarantee the confidentiality of s, e generated by rejection sampling with η=1 based on M-LWE [7]. Accordingly, RA cannot acquire $\mathtt{s}_V$ from $\mathtt{t}_V$. The **PCA**'s sk, $\mathtt{s}_{PCA}$, is protected through the $\mathtt{s}_{rv} \leftarrow \mathtt{s}_{rand} + \mathtt{s}_{PCA}$. Hence, the vehicle can only generate u_i using $\mathtt{s}_{rv}$, and **PCA**'s sk is unknown. Since the vehicle cannot know $\mathtt{s}_{rand}$ nor generate $\mathtt{t}_{rv}$, the certificate can only be issued by **PCA** and cannot be forged. Additionally, due to **RA**'s shuffling, the **PCA** cannot map the **vehicle** to which the certificate is subject. This ensures anonymity during the issuance process.

Vehicle	RA	PCA	Vehicle
$\mathtt{s}_V, \mathtt{e}_V \leftarrow S_1^{6\times 5}$ $\mathtt{t}_V \leftarrow \mathtt{A}\mathtt{s}_V + \mathtt{e}_V$ $\mathtt{A} \leftarrow Expand(\rho)$ $\mathtt{sk}(\rho, \mathtt{t}_V, \mathtt{s}_V, \mathtt{e}_V)$ $\mathtt{pk}(\rho, \mathtt{t}_V)$	$\mathtt{s}_i, \mathtt{e}_i \leftarrow S_1^{6\times 5}$ $\mathtt{t}_i \leftarrow \mathtt{t}_V + \mathtt{A}\mathtt{s}_i + \mathtt{e}_i$ $\mathtt{t}_i \leftarrow \mathtt{A}(\mathtt{s}_V + \mathtt{s}_i, \mathtt{e}_V + \mathtt{e}_i)$, $(0 \leq i \leq \beta)$ $\mathtt{sk}(\rho.\mathtt{t}_i, \mathtt{s}_{V+i}, \mathtt{e}_{V+i})$ $\mathtt{pk}(\rho, \mathtt{t}_i)$	$\mathtt{s}_{rand}, \mathtt{e}_{rand} \leftarrow S_1^{6\times 5}$ $\mathtt{s}_{rv} \leftarrow \mathtt{s}_{rand} + \mathtt{s}_{PCA}$ $\mathtt{e}_{rv} \leftarrow \mathtt{e}_{rand} + \mathtt{e}_{PCA}$ $\mathtt{t}_{rand} \leftarrow \mathtt{t}_V + \mathtt{t}_{rand}$ $\mathtt{t}_{RV} \leftarrow \mathtt{t}_i + \mathtt{t}_{rand}$ $sig_i = (\mathtt{t}_{rand}, \mathtt{s}_{rv}, \mathtt{e}_{rv})$ Pkg = Enc$(\mathtt{t}_i, sig_i)$ $\mathtt{sk}(\rho, \mathtt{t}_{RV} + \mathtt{t}_{PCA},$ $(\mathtt{s}_V + \mathtt{s}_i + \mathtt{s}_{rv}, \mathtt{e}_V + \mathtt{e}_i + \mathtt{e}_{rv}))$ $\mathtt{pk}(\rho, \mathtt{t}_{RV}, \mathtt{t}_{PCA})$	$\mathtt{s}_i, \mathtt{e}_i \leftarrow S_1^{6\times 5}$ $\mathtt{t}_i \leftarrow \mathtt{A}(\mathtt{s}_V + \mathtt{s}_i) + (\mathtt{e}_V + \mathtt{e}_i)$ Dec$(\mathtt{t}_i, Pkg)$ $\mathtt{t}_{RV} \leftarrow \mathtt{t}_i + \mathtt{t}_{rand}$ $U_i = \mathtt{t}_{RV}$ $u_i = (\mathtt{s}_V + \mathtt{s}_i + \mathtt{s}_{rv}, \mathtt{e}_V + \mathtt{e}_i + \mathtt{e}_{rv})$ $\mathtt{A}(\mathtt{s}_V + \mathtt{s}_i + \mathtt{s}_{rv}) + (\mathtt{e}_V + \mathtt{e}_i + \mathtt{e}_{rv})$ $\stackrel{?}{=} \mathtt{t}_{RV} + \mathtt{t}_{PCA}$ Implicit certificates (U_i, u_i)
$\mathtt{pk}(\rho, \mathtt{t}_V), \rho' \rightarrow$	$\mathtt{pk}(\rho, \mathtt{t}_i) \rightarrow$	Pkg $\rightarrow$	$\rightarrow$ Pkg $\rightarrow$ Verify

4.2 Performance analysis

In V2X communication, the size of `RV` is important because packet size affects communication overhead. `MDIC` reduce the explicit certificate size from 1,952 bytes + 3,293 bytes = 5,245 bytes to 1,952 bytes by generating `RV`. In addition, BSM verification of Implicit certificates guarantees a certificate-based trust system with only one verification using $\mathtt{t}_{rv}$ which is received and $\mathtt{t}_{PCA}$ together. The verification is as follows: $Hash[HighBits[\mathtt{A}\mathtt{z} - \mathtt{c}\mathtt{t}_{rv} - \mathtt{c}\mathtt{t}_{PCA}]\mathtt{m}] \stackrel{?}{=} \mathtt{c}$. The BSM verification is performed only once without certificate verification, so the reciver's operation overhead can be reduced by half.

5 Conclusion

In this paper, we propose a pure-PQC SCMS for environments that do not allow existing encryption systems after 2030. Implicit Certificates generation, which failed based on M-LWE, was solved by accumulating ($\mathtt{s}$, $\mathtt{e}$). Trust in V2X that uses `MDIC` is guaranteed because only Vehicle can generate u_i and only PCA can generate t_{rv}. Additionally, the overhead of communication and computation is significantly reduced. Future work include Implicit certificates that can be applied to various parameter sets.

Acknowledge This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) ((No. 2021-0-00796, Research on Foundational Technologies for 6G Autonomous Security-by-Design to Guarantee Constant Quality of Security, 50%) and Following are results of a study on the "Convergence and Open Sharing System" Project, supported by the Ministry of Education and National Research Foundation of Korea. (50%).

References

1. Khan, Manzoor Ahmed, et al. "Level-5 autonomous driving—are we there yet? a review of research literature." ACM Computing Surveys (CSUR) 55.2 (2022): 1-38.

2. G. Twardokus, N. Bindel, H. Rahbari, and S. McCarthy, “When cryptography needs a hand: Practical post-quantum authentication for V2V communications,” in Proc. Network and Distributed System Security Symposium (NDSS), San Diego, CA, Feb. 2024.
3. Paulo S. L. M. Barreto, Jefferson E. Ricardini, Marcos A. Simplicio, Jr., and Harsh Kupwade Patil. 2018. qSCMS: Postquantum certificate provisioning process for V2X. IACR Cryptol. ePrint Arch. (2018), 1247
4. Bindel, Nina, and Sarah McCarthy. "The need for being explicit: Failed attempts to construct implicit certificates from lattices." The Computer Journal 66.6 (2023): 1320-1334
5. Barreto, Paulo SLM, et al. "Schnorr-based implicit certification: Improving the security and efficiency of vehicular communications." IEEE Transactions on Computers 70.3 (2020): 393-399.
6. National Institute of Standards and Technology: Module-Lattice-Based Digital Signature Standard. Tech. Rep. NIST FIPS 204 ipd, National Institute of Standards and Technology, Gaithersburg, MD (Aug 2023). https://doi.org/10.6028/NIST.FIPS.204.ipd
7. Boudgoust, Katharina, et al. "On the hardness of module learning with errors with short distributions." Journal of Cryptology 36.1 (2023): 1.

Efficient Implementation of Falcon on Cortex-M4

Junhyeok Choi[1][0009−0004−8087−3940], Seung-Ju Wi[2][0009−0004−2289−0582], and Seog Chung Seo[1,2,*][0000−0001−8016−2808]

[1] Department of Financial Information Security, Kookmin University, Seoul, South Korea
[2] Department of Information Security, Cryptology and Mathematics, Kookmin University, Seoul, South Korea
{c981126,jhgf9630,scseo}@kookmin.ac.kr

Abstract. Falcon is a NIST standardized digital signature algorithm with a small key size and a small signature size. Compared to key generation and signature generation, signature verification is much more frequently used operation. Since embedded devices have limited computing and memory capacity, performance optimization is essential for its smooth usage. In this paper, we present the results of an optimized implementation of Falcon signature verification on Cortex-M4 which is the standard embedded device for performance comparison among PQC algorithms. For optimizing the performance of Falcon signature verification, we concentrate on optimizing the NTT-based polynomial multiplication process (composed of NTT conversion, point-wise multiplication, and inverse NTT conversion), which is the core operation of signature verification in Falcon. Compared to the latest Falcon implementation on pqm4 project, the proposed NTT implementation achieves performance improvements of 341%/180%/368% (Falcon-512) and 365%/213%/382% (Falcon-1024) for NTT conversion, point-wise multiplication, and inverse NTT conversion, respectively, which contributes to the overall performance improvement of 64% (Falcon-512) and 75% (Falcon-1024) for overall signature verification.

Keywords: Post-Quantum Cryptgoraphy · Cortex-M4 · Falcon · Optimization

1 Introduction

In a quantum computing environment, the Shor algorithm can solve prime factorization and discrete logarithm problems in polynomial time. Advances in quantum computers threaten traditional public-key cryptosystems like RSA and

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) ((No. 2021-0-00796, Research on Foundational Technologies for 6G Autonomous Security-by-Design to Guarantee Constant Quality of Security, 50%) and was partly supported by a study on the "Convergence and Open Sharing System" Project, supported by the Ministry of Education and National Research Foundation of Korea. (50%).

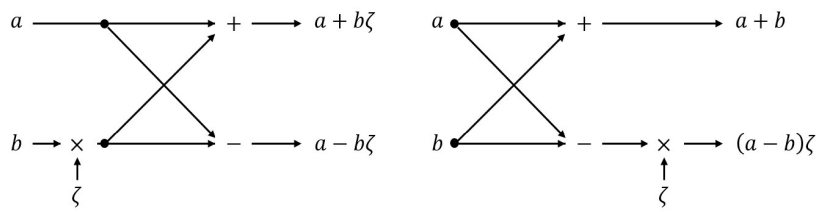


Fig. 1: CT (left) and GS (right) butterfly in NTT

ECC, which are based on these mathematical challenges. Research on Post-Quantum Cryptography (PQC) is currently underway, and the National Institute of Standards and Technology (NIST) has selected one Public Key Encryption/Key Encapsulation Mechanism (PKE/KEM), Crystals-Kyber, and three Digital Signature Algorithms (DSAs), Cryptals-Dilithium, SPHINCS+, and Falcon, for standardization in 2022.

Embedded devices used in environments such as autonomous driving, drones, and IoT have limited performance and memory, so PQC optimization is required. Among them, Falcon is an algorithm that uses NTRU lattice, Fast Fourier Transform (FFT), and Number Theoretic Transform (NTT), which has a shorter key length and faster signature verification speed than other DSAs [1]. Falcon was selected as the best digital signature algorithm for autonomous driving environments that require fast signature verification and for TLS 1.3, which supports FP operations [2, 3].

2 Preliminaries

2.1 Falcon

Falcon is a digital signature cryptographic algorithm based on the NTRU lattice problem. Its internal functions are mainly composed of Keygen, Sign, and Verify. In the Verify, the signature is verified by checking whether the signature is smaller than a certain range based on the message m, signature sig, and public key pk. Falcon uses a polynomial multiplication NTT with O(nlogn) computational complexity during Verify. Depending on the security strength, Falcon-512 performs the NTT operation on polynomials in $\mathbb{Z}_{12289}[x]/(X^{512}+1)$ and Falcon-1024 performs it on polynomials in $\mathbb{Z}_{12289}[x]/(X^{1024}+1)$. The process of transforming an existing polynomial by multiplying it with a twiddle factor is called Forward NTT (NTT), and the reverse process is called Inverse NTT (INTT). Figure 1 shows the Cooley-Tukey (CT) butterfly and Gentleman-Sande (GS) butterfly algorithms used for NTT operations.

2.2 Cortex-M4

The Cortex-M4 is a 32-bit Microcontroller Unit (MCU) that provides 16 general-purpose registers and 32 signle-precision floating-point registers (FP registers). Out of the 16 general-purpose registers, 14 can be used for actual computations. We use the FP registers to store the twiddle factors used in NTT operations.

Algorithm 1 CT Butterfly with Plantard multiplication

Input: a, b (a=$(a_t||a_b)$, b=$(b_t||b_b)$)
Input: twiddle(=ζ), tmp
Input: q(=12289), qa(=$q2^{\alpha}$)
Output: a = $(a_t + b_t\zeta \ || \ a_b + b_b\zeta)$
Output: b = $(a_t - b_t\zeta \ || \ a_b - b_b\zeta)$

```
1: smulwb  tmp, twiddle, b
2: smulwt  b, twiddle, b
3: smlabb  tmp, tmp, q, qa
4: smlabb  b, b, q, qa
5: pkhtb   tmp, b, tmp, asr#16
6: usub16  b, a, tmp
7: uadd16  a, a, tmp
8: return a,b
```

Algorithm 2 GS Butterfly with Plantard multiplication

Input: a, b (a=$(a_t||a_b)$,b=$(b_t||b_b)$)
Input: twiddle(=ζ), tmp, tmp2
Input: q(=12289), qa(=$q2^{\alpha}$)
Output: a = $(a_t + b_t\zeta \ || \ a_b + b_b\zeta)$
Output: b = $(a_t - b_t\zeta \ || \ a_b - b_b\zeta)$

```
1: ssub16  tmp, a, b
2: sadd16  a, a, b
3: smulwb  tmp2, twiddle, tmp
4: smulwt  tmp, twiddle, tmp
5: smlabb  tmp2, tmp2, q, qa
6: smlabb  tmp, tmp, q, qa
7: pkhtb   b, tmp, tmp2, asr#16
8: return a,b
```

It also provides Digital Signal Processing (DSP) extension instructions such as Single Instruction Multiple Data (SIMD) and Multiply Accumulate Instructions (MAC) for performance enhancement.

3 Proposed Optimized Implementation

3.1 Plantard multiplication on Cortex-M4

We use Plantard multiplication, which performs word size (32, 64-bit) modular multiplication between polynomial coefficients and twiddle factors. The twiddle factor used in the NTT operation can be precomputed by a combination of the primitive root of unity and bit-reversal. Since q' and the plantard constant $-2^{2l} \ mod^{\pm} q$ are fixed values, they can be stored as a 32-bit table by multiplying the twiddle factor, q', and $-2^{2l} \ mod^{\pm} q$. By utilizing this to perform the NTT operation, we can reduce the number of multiplications between Plantard multiplications by one and skip the domain transformation.

Suppose a pair $(a_b, b_b), (a_t, b_t)$ consist of 16-bit coefficients a_b, a_t, b_b, b_t performs a butterfly operation using the same twiddle factor ζ. We pack the factors to store $a_t||a_b, b_t||b_b$ in 32-bit registers a, b, respectively, and store ζ in register twiddle. Algorithm 1,2 is the process of performing CT butterfly and GS butterfly operations on the $(a_b, b_b), (a_t, b_t)$ pair by utilizing DSP instructions, respectively [4].

3.2 NTT, Pointmul, and INTT

We use CT butterfly/GS butterfly for NTT/INTT implementation respectively and Barrett reduction for reduction.

NTT Out of the 14 general-purpose registers available for actual operations, 8 registers are used to store coefficients. The remaining registers are used to store constants required for Plantard multiplication, Barrett reduction, or as temporary registers. We apply optimization techniques such as layer-merging and lazy reduction to minimize memory accesses and the number of reductions.

Table 1: Cycle counts for the Verify and core polynomial arithmetic

Scheme	Implementation	Verify	NTT	Pointmul	INTT
Falcon-512	This work	291,846 (+64%)	15,974 (+341%)	3,108 (+180%)	17,047 (+368%)
	pqm4	480,051	70,584	8,716	79,876
	reference	566,860	76,437	9,265	82,185
Falcon-1024	This work	576,668 (+75%)	33,138 (+365%)	6,210 (+213%)	35,803 (+382%)
	pqm4	1,011,895	154,169	19,499	172,751
	reference	1,150,098	167,510	18,475	177,868

We use 3-3-3 layer-merging for the 9 layers of Falcon-512 and 3-3-4 layer-merging for the 10 layers of Falcon-1024. And we perform reduction after 3,6-layer in the NTT.

Pointmul We perform point-wise multiplication using Plantard multiplication on the two polynomials after NTT. Since we use Plantard multiplication for modular multiplication, it is necessary to multiply the result of the operation by the Plantard constant $-2^{2l}\ mod^{\pm} q$. To avoid an additional multiplication, we merge this step with the operation of multiplying by $\frac{1}{N}$ after the INTT.

INTT We use GS butterfly in INTT. Due to the characteristics of GS butterfly, coefficients that are subjected to addition have their upper and lower bounds doubled, and coefficients that are multiplied with twiddle factors have an output range of $(-0.5q, 0.5q)$. If a coefficient with a range of $(-0.5q, 0.5q)$ is added three times in a row, only that part will be $4q$, which exceeds the range of values that can be stored. In this paper, we use loop unrolling to perform reduction only for the coefficients that need to be subtracted for each layer. Also, the last step of INTT involves multiplying by $\frac{1}{N}$ to transform to the integer domain. We combine this process with multiplying by the Plantard constant to handle the $-2^{-2l}\ mod^{\pm} q$ resulting from point-wise multiplication.

4 Result

The benchmarking results are obtained on NUCLEO-L4R5ZI board. The implementation and benchmarking is done using STM32CubeIDE. The comparison implementations are pqm4 [5], a PQC benchmark study targeting Cortex-M4 boards, and the Falcon code submitted to Round 3 of the NIST PQC competition [6]. It was run with a frequency of 20 MHz and compiled options of -O3.

Table 1 shows the overall performance of Falcon-512, Falcon-1024 Verify and the cycle counts for the NTT implementation NTT/Pointmul/INTT. Overall Verify shows a performance improvement of 63%/75% compared to pqm4, respectively. NTT/Pointmul/INTT shows performance improvements of 341%, 180%, and 368% for Falcon-512 and 365%, 213%, and 382% for Falcon-1024, respectively, compared to the pqm4 implementation.

References

1. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. Falcon: Fast-fourier lattice-based compact signatures over ntru. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5):1–75, 2018.
2. Nina Bindel, S McCarthy, H Rahbari, and G Twardokus 3rd. Suitability of 3rd round signature candidates for vehicle-to-vehicle communication. In *Workshop Record of the Third PQC Standardization Conference. Available at https://csrc. nist. gov/Presentations/2021/suitability-of-3rd-roundsignature-candidates-for*, 2021.
3. Dimitrios Sikeridis, Panos Kampanakis, and Michael Devetsikiotis. Post-quantum authentication in tls 1.3: a performance study. *Cryptology ePrint Archive*, 2020.
4. Junhao Huang, Jipeng Zhang, Haosong Zhao, Zhe Liu, Ray CC Cheung, Çetin Kaya Koç, and Donglong Chen. Improved plantard arithmetic for lattice-based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):614–636, 2022.
5. Matthias J Kannwischer, Joost Rijnveld, Peter Schwabe, and Ko Stoffelen. Testing and benchmarking nist pqc on arm cortex-m4. Radboud Repository, 2019. Available at: http://hdl.handle.net/2066/210214.
6. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. *Falcon Specifications and Supporting Documentation*. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

Repeated Embedding Techniques of Duplicated Least Significant Bits for Secure Digital Watermarks

Chae-Won Song[1], So-Hyun Park[1], Il-Gu Lee[1]

[1]Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
{20221106, 220227022, iglee} @sungshin.ac.kr

Abstract. Digital watermarking technology is used to hide data or identify information within digital multimedia. Watermark insertion technology is classified into transform domain techniques and spatial domain techniques. Among spatial domain techniques, the representative LSB (Least Significant Bit) technique hides information by replacing the least significant bit of a selected pixel with watermark information. This technique has little effect on the quality of images or audio but is vulnerable to distortion attacks such as noise. Therefore, to counter distortion attacks such as noise, this paper proposes the LSB-DR (Duplicate and Repeat) technique, which duplicates the watermark information embedded in the LSB within a pixel into multi-bits and embeds that pixel repeatedly into multiple pixels. This technique can mitigate the effects of noise attacks through multi-bit duplication and pixel repeat insertion.

Keywords: LSB, Data bits, Duplicated LSB, Repeated Embedding

1. Introduction

Watermarking is a technology that hides data or identifies information within digital multimedia. Digital watermarking inserts a watermark into media so that third parties cannot easily extract information without significantly degrading the visual quality. Watermark insertion techniques are classified into spatial domain techniques and transform domain techniques. The frequency transform technique, which is a representative transform domain technique, converts an image or audio file into the frequency domain and then inserts a watermark into the corresponding frequency component. The least significant bit (LSB) technique is a representative spatial domain technique. The LSB technique hides information by replacing the least significant bit of a selected pixel with watermark information [1].

Each pixel in an image is usually composed of RGB values, and each color channel (R, G, B) has a value between 0 and 255. Therefore, when extracting information, the least significant bit of each color channel is read to restore the original secret information [2].

Since the LSB technique changes the least significant bit of a pixel, it has little effect on the quality of images or audio. However, the LSB technique is sensitive to minor changes from the outside. In particular, distortion attacks such as noise easily damage data hidden in the LSB. Secret messages hidden using the LSB technique cannot be easily restored during a noise attack. Noise can

cause changes in the media and affect all bits including the least significant bit. Therefore, in this paper, to complement the vulnerability of the LSB-based watermarking technique, we propose the LSB-DR (LSB Duplicate and Repeat) technique, which duplicates the watermark information embedded in the LSB into multi-bits and repeatedly embeds it in multiple pixels.

2. Related Work

Digital watermarking technology plays an important role in copyright protection and information security of multimedia content. Various studies have been conducted in this field; among them, studies utilizing the LSB technique have been conducted steadily. Deepshikha Chopra [3] evaluated the performance of the LSB technique in various distortions such as Salt and Pepper Noise, Gaussian Noise, and Speckle Noise. This study emphasized the necessity and importance of digital watermarking and proved that the LSB technique can be used effectively for copyright protection.

Ranjeet Kumar Singh [4] conducted a study on extracting the original watermark by applying various noises to the watermarked image using the LSB technique. The significance of this study is that the original watermark can be extracted even when various noise attacks are performed. Still, there is a problem with the inserted watermark being identified by extracting the LSB in reverse.

Abdullah Bamatraf [5] proposed a new digital watermarking algorithm combining the LSB technique with the Inverse Bit. The proposed algorithm inverts the binary value of the watermark text and inserts it in a different order. The proposed technique is robust to attacks such as cropping and provides high security while maintaining image quality. However, the study focused on gray-scale images and did not examine other multimedia formats such as color images or videos, which limits its practical application scope.

This paper proposes a technique that duplicates the watermark information embedded in the LSB within a pixel into multi-bit and repeatedly embeds it in multiple pixels.

3. LSB-DR (LSB Duplicate and Repeat) Technique

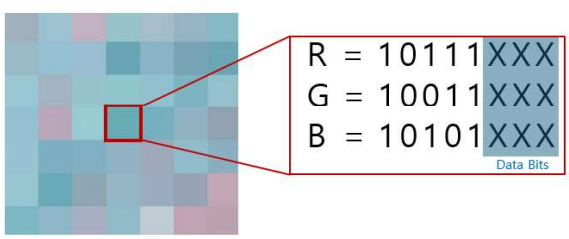


Fig. 1. Data Bits of one pixel in the image

In this paper, we propose a method to restore the original watermark even under watermark distortion attacks by duplicating the watermark information embedded in the LSB into multi-bits and repeatedly embedding it in multiple pixels. Since the conventional method inserts information data into the LSB of each image pixel, it is difficult to extract the entire watermark normally if a partial attack is applied to the image. When a malicious attacker tries to remove or damage data, it can damage the original watermark by adding noise or performing a compression attack. Distortion attacks such as noise weaken the robustness of the watermark [6].

In this paper, we propose a technique to repeatedly embed LSBs into multiple pixels of an image by utilizing multi-bits, including LSBs, to solve the problem that LSB techniques are easily damaged by attacks such as noise addition and compression.

Figure 1 shows an example of LSB-DR-based multi-bit replication in a pixel in a specific image. Three bits out of 8 bits of a pixel are set as data bits, and a watermark text is inserted into the set data bits. By duplicating the watermark information embedded in LSBs into multi-bits within a pixel, the inserted watermark text can be restored and extracted even in the case of a watermark neutralization attack.

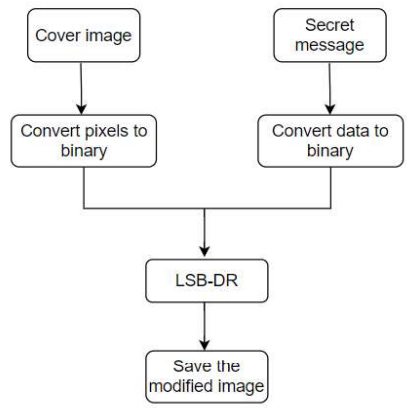


Fig. 2. Flowchart for LSB-DR insertion operation

Figure 2 is a flowchart of the insertion operation of the proposed LSB-DR technique. First, the size of the host image to be inserted into the watermark text is loaded, and the data of each image pixel is converted into a binary string. The secret message, which is the watermark information to be inserted, is also converted into a binary string and embedded into the host image using the LSB-DR technique.

The LSB-DR technique has two different features compared to the existing LSB embedding technique: 'Duplicated LSB' and 'Repeated Embedding'. Duplicated LSB duplicates the watermark information embedded in the LSB within a pixel to multi-bits. Repeated Embedding embeds the pixel duplicated in multi-bit into multiple pixels of the image. After going through the duplicated LSB and repeated embedding processes, the modified binary pixel data is converted into image pixels according to the inserted data, creating a new pixel array and a new image.

It is important to design the extraction algorithm by considering the length of information in the hidden data and the data bits set during encoding. Open the image file, load the pixel data, and extract the bits from the set multi-bit RGB value. Generate the binary data of the hidden message, and extract the file extension, message size, and actual message from the binary data.

The proposed LSB-DR watermarking technique can respond to distortion attacks that attackers maliciously attempt to damage data. In addition, inserting watermarks using duplicated LSB and repeated embedding can restore the original watermark even if the watermark is damaged due to watermark distortion attacks such as crop and noise.

4. Conclusion

This paper proposes a watermark embedding technique based on LSB-DR that is robust to watermark distortion attacks. The proposed technique applies duplicated LSB, which duplicates watermark information embedded in LSB into multi-bits, and repeated embedding, which repeatedly embeds duplicated LSB pixels into multiple pixels to cope with distortion attacks such as noise, crop, and blur attacks. In future work, we will conduct experiments on various distortion attacks to evaluate the performance of the proposed technique and enhance its security. In addition, we will apply the proposed technique to multimedia content and evaluate the watermark embedding and attack response performance.

Acknowledgments. This paper was conducted as a result of the research results of the 2024 Industrial Innovation Talent Growth Support Project (RS-2024-00415520) of the Ministry of Trade, Industry and Energy and the Korea Institute for Advancement of Technology (RS-2024-00415520) and the ICT Innovation Talent 4.0 Project of the Ministry of Science and ICT and the Korea IT Planning and Evaluation Institute (No. IITP-2022-RS-2022-00156310)

References

[1] Deepshikha, C., Preeti, G., Gaur, S., Anil, G.: Lsb Based Digital Image Watermarking For Gray scale Image. IOSR Journal of Computer Engineering, Volume 6, pp. 36-41, 2012

[2] Arun Kumar, S., Juhi, S.: Steganography in Image Using LSB Technique. International Journal of Latest Trends in Engineering and Technology, Volume 5, pp. 426-430, 2023

[3] Deepshikha, C., Preeti, G., Gaur, S., Anil, G.: Lsb Based Digital Image Watermarking For Gray scale Image. IOSR Journal of Computer Engineering, Volume 6, pp. 36-41, 2012

[4] Ranjeet Kumar, S., Dilip Kumar, S.: Experimental Studies of LSB Watermarking with Different Noise. Procedia Computer Science, Volume 54, pp. 612-620, 2015

[5] Abdullah, B., Rosziati, I.: A New Digital Watermarking Algorithm Using Combination of Least Significant Bit (LSB) and Inverse Bit. Journal of Computing, Volume 3, ISSN 21-51-9617, 2011

[6] Wenbo, W., Jun, W., Yunming, Z., Jling, L.: A Comprehensive Survey on Robust Image Watermarking. Neurocomputing, Volume 488, pp. 226-247, 2022

Practical Zero Trust Implementation for Low Latency and High Reliability Networks

Seo-Yi Kim[1][0009-0004-4890-1972] and Il-Gu Lee[1][0000-0002-5777-4029]

[1] Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul, Korea
cheonrang01@gmail.com, iglee@sungshin.ac.kr

Abstract. This paper proposes a dynamic access control and adaptive zero trust approach to address latency issues when implementing zero trust systems. The proposed method applies the zero-trust model and verifies all traffic data only when an attack is anticipated. Additionally, it performs machine learning-based dynamic access control to enable the use of the zero-trust model quickly and efficiently.

Keywords: Zero Trust, Dynamic Access Control, Low Latency

1 Introduction

Zero-trust is a cybersecurity model that designs, builds, and operates an information protection system based on the principle of "trust no one." It fundamentally does not trust any network activity and verifies and authenticates all access requests, both internal and external [1]. This contrasts with the traditional Perimeter Security Model, which divides the network into internal and external zones, considers the internal zone as trusted with fewer restrictions, and sets the external zone as untrusted, protected by firewalls and intrusion detection systems. However, the rise of cloud computing, mobile devices, the Internet of Things, and the increase in remote work have made boundary-setting difficult and heightened the risk of compromised single defense lines [2]. Additionally, the prominence of insider threats has underscored the need for the zero trust model. While zero-trust offers advantages in terms of security, the continuous authentication process can impact network performance, and its implementation and operation can be complex.

This paper proposes a dynamic access control and adaptive strategy for efficient zero trust implementation. Applying feature selection to network traffic data reduces the number of features used during verification, and machine learning algorithms are used to perform dynamic access control quickly and accurately. Additionally, instead of verifying all network traffic data under normal circumstances, only a portion is verified at appropriate intervals. When an attack is detected, an adaptive zero trust verification approach is proposed, which verifies all logs similar to the existing zero trust model.

2 Dynamic Access Control and Adaptive Zero Trust

2.1 Machine Learning-based Dynamic Access Control

After collecting traffic data, a data preprocessing step is conducted. Since zero trust performs continuous authentication for all access subjects, the verification time and resource usage increase compared to traditional perimeter-based methods. If the authentication process is not performed quickly, delays can critically impact network performance. Therefore, fast and accurate authentication and access control are essential [3]. The Mean Decrease in Impurity (MDI) embedded method of Feature Selection is employed to achieve this. Only data with feature importance above a certain threshold is retained, while features below the threshold are discarded. The remaining data undergoes encoding and normalization. The preprocessed data is then used with the Extreme Gradient Boosting (XGBoost) algorithm to determine whether access is legitimate. If access is deemed legitimate, it is allowed; otherwise, it is blocked.

2.2 Adaptive Zero Trust

While the zero trust model can enhance security levels compared to traditional perimeter-based models, its adoption and operational costs are relatively high, and implementation can be challenging. This is particularly problematic in environments where lightweight, low-power, and low-cost devices like those in the Internet of Things operate, as these devices have limited data processing and storage resources. Therefore, this paper proposes an adaptive zero trust approach that applies the zero trust security model only when an attack is suspected.

The adaptive zero trust approach performs verification on only a subset of log data during normal conditions when no attack is detected. When an attack is suspected, the full zero trust model is applied to verify all log data. This adaptive verification method can lower the barriers to adopting zero trust, allowing for fast and accurate attack detection at a reduced cost.

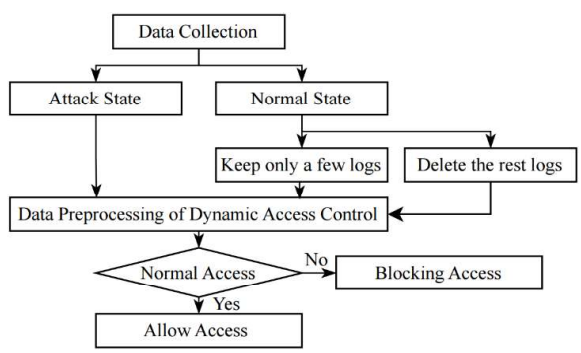


Fig. 1. Flowchart of Adaptive Zero Trust Method

3 Evaluation and Analysis

3.1 Evaluation Environment

For performance evaluation, the KDD-NSL dataset [4] was used. This dataset, created in 2009 by the New School of Information Security & Technology (NSL), is a large-scale dataset that includes various types of network attacks and is based on real network traffic, making it widely used in network-related research.

For feature selection, the Mean Decrease in Impurity (MDI) method was performed using the random forest algorithm, and features with an importance score lower than 0.02 were removed. In the case of the KDD-NSL dataset, out of a total of 42 features, only 17 features were used after excluding 25 features. One-hot encoding was performed to convert categorical data into numerical data, and min-max normalization was applied to integer data.

To validate the adaptive zero trust approach, the dataset was restructured. The KDD-NSL dataset is composed of a similar proportion of normal data and attack data. In a real environment, most verification requests are made by normal data, with malicious attack data being a minority. Reflecting this real-world scenario, the test dataset was structured to simulate adaptive zero trust verification. The test dataset maintained the same size as the original simulation but adjusted the ratio of normal logs to 55%, 75%, and 95%. Normal data was constituted according to the fixed normal log ratio, and the remaining data was randomly selected. The proposed method reduced false positives by verifying only 10% of the log data when no attack was detected and verifying all logs if five attacks were detected.

3.2 Experiment Result

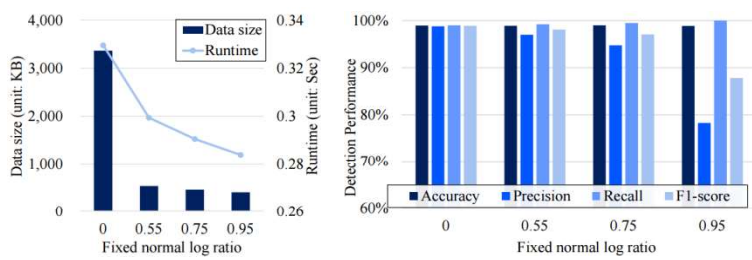


Fig. 2. (a) Comparison of data size and Runtime according to fixed normal log ratio (left), (b) Comparison of detection performance with fixed normal log ratio (right)

Fig. 2(a) is a graph comparing the dataset size and execution time for verification based on the proportion of normal logs. When the fixed normal log ratio is 0%, the adaptive zero trust method is not applied, and authentication is performed on all traffic. Comparing the size of the datasets used for actual verification, the dataset size decreased by an average of 86% when the adaptive zero trust method was applied. Additionally, the execution time decreased by an average of 12%.

In the graph of Fig. 2(b), the detection performance based on the fixed normal log ratio was compared. For the KDD-NSL dataset, when the fixed normal log ratio was set to 95%, the detection performance deteriorated by an average of 7.8%. When the fixed normal log ratios were 55% and 75%, the detection performance deteriorated by an average of 0.6% and 1.3%, respectively. However, a high detection performance of over 95% was observed across all evaluation metrics.

4 Conclusion

While the zero trust model can achieve a high level of security, it is challenging to operate while meeting performance requirements. We proposed an adaptive zero trust approach to address the difficulties of adopting the zero trust model. When applied with a dynamic access control method based on network traffic, it achieved significant improvements in dataset size and verification execution time while maintaining detection performance similar to that of traditional methods.

Acknowledgments

This work was partly supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean Government (MOTIE) (P0008703, The Competency Development Program for Industry Specialists) and MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310), supervised by the Institute of Information Communication Technology Planning and Evaluation (IITP).

References

1. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R., "Zero trust architecture (zta): A comprehensive survey," IEEE Access, 10, 57143-57179, 12 May 2022.
2. TN, N., Pramod, D., Singh, R., "Zero trust security model: Defining new boundaries to organizational network," In Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing, pp. 603-609, August. 2023
3. So-Hui Kim, Il-Gu Lee, Ye-Eun Lee, "Practical Zero Trust Technology and Policy for 5G Network Convergence Industry," Korean Journal of Industrial Security, Vol. 14, pp.203-222, Jan. 2024
4. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009

A Study on the Strengthening of the Research Security System: Focusing on the Comparative Analysis of Laws and Systems in Major Countries

Hyun-Che Song[1], Hye-In Lee[1] and Il-Gu Lee[1]

[1] Sungshin University, 2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul, Republic of Korea
{ssongzz, lhynee, iglee}@sungshin.ac.kr

Abstract. This study proposes strategies to enhance the security of the research security system for national R&D projects by benchmarking. It identifies domestic limitations and recommends adopting robust systems from the U.S., Japan, and the U.K., enhancing responsibilities for researchers and institutions, and creating a comprehensive case compendium and training portal. The goal is establishing a reliable research environment and protecting national research assets.

Keywords: Research Security, R&D, Laws and Systems

1 Introduction

The first national R&D project in Korea where the government provided substantial research funding for advancing science and technology was the "Specific Research and Development Project," launched in 1982 with a budget of 13.3 billion KRW under the Technology Development Promotion Act. This project has significantly contributed to developing core strategic technologies and advancing the R&D system.

Science and technology drive economic growth and industrial development, necessitating investment in R&D. Economic growth requires capital, labor, and productivity advancements, with science and technology being crucial for improving productivity.

In the era of technological hegemony, a research security system is essential to protect national research assets and align with international standards. To create a secure environment for researchers, it is crucial to benchmark international trends and enhance research security laws according to the "National Research and Development Innovation Act." This paper proposes measures to strengthen the research security system.

2 Research Security Management

Research security management (hereafter 'Research Security') refers to all activities undertaken by research institutions and researchers to prevent unauthorized leakage of research and development outputs from the preparation stage through the execution

process and after the completion of the research. The components of Research Security can be categorized into three main areas.

Security Management Subjects, which include the research institutions and researchers conducting R&D projects; Security Management Targets, which encompass the researchers involved, the research facilities required, and the research data and outputs generated; and Security Management Methods, which involve measures to prevent both intentional and accidental damage or leakage of security targets, such as security regulations, security devices, and security systems [1].

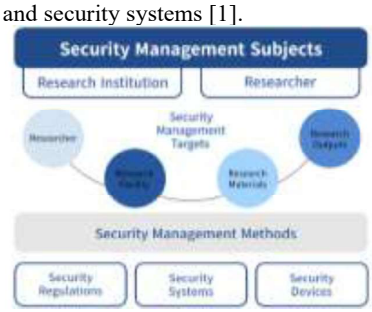


Fig. 1. Components of Research Security.

3 Trends in Research Security Overseas

Examining the research security situations in three major countries—the United States, Japan, and the United Kingdom—reveals common measures such as establishing dedicated organizations or mandating the reporting of foreign funding sources. These countries distribute field guides to researchers and research institutions, establish strong and diverse research security systems to prevent conflicts of interest, enhance research ethics, and proactively protect their researchers and key research assets.

3.1 United States

The U.S. has a comprehensive legal framework, with dedicated organizations under the CHIPS and Science Act and roles defined in NSPM-33. Researchers must report foreign funding when applying for government grants [2].

3.2 Japan

Japan emphasizes voluntary risk management by researchers rather than strict regulations. Legally and institutionally, the Ministry of Education, Culture, Sports, Science and Technology (MEXT) and research institutions have defined roles. Researchers must report any receipt of foreign research funds, and the reporting system has been revised to include domestic and foreign sources.

3.3 United Kingdom

The UK focuses on advisory roles for research sites. A Research Collaboration Advice Team provides security advice, and UKRI's "Trusted Research and Innovation

Principles" mandate reporting of personal and financial interests, including foreign relationships.

4 Domestic Trends in Research Security

In Korea, research security regulations are included in the Framework Act on Science and Technology, the National Research and Development Innovation Act, and the Research Outcome Evaluation Act. Additionally, the Ministry of Science and ICT establishes security measures under the Innovation Act [3]. Various ministries and research institutions are required to develop security measures, classify security tasks, and implement appropriate security measures. Compared to international trends, the limitations of domestic research security can be categorized into three areas: legal/institutional frameworks, security task management, and support systems, which are summarized below.

Table 1. Limitations of Domestic Research Security

Legal/Institutional Framework	Security Task Management	Support Systems
• Inadequate Legal Framework	• Simplistic Classification Criteria	• Absence of Dedicated Support Systems
• Divergent Regulations		• Lack of International Collaboration
• Across Departments	• Inadequate Classification Procedures	• Limitations in Situational Assessments
• Vague Grounds for Sanctions		• Personnel Management
• Unreasonable Obligations	• Unreasonable Performance Management	• Insufficient Security Awareness
• Insufficient Institutional Duties		• Inadequate Advisory Channels
• Uniform Management	• Divergent Management Systems	

5 Proposal for Benchmarking Research Security Laws and Systems of Major Countries

As demonstrated by the cases of major countries, achieving a balance between research integrity and autonomy requires effective responsibility and collaboration among researchers, research institutions/support organizations, and the government. Beyond rationalizing security task management, building a research ecosystem that aligns with evolving global standards and addresses the challenges of technological hegemony is essential. This proposal suggests establishing a research security system that balances these needs and reduces the burden on research environments [4].

Table 2. Proposal for Establishing a Research Security System

Legal/Institutional Framework	Security Task Management	Support Systems
• Establishing a Foreign Funding Information Management System • Establishing Comprehensive Security Regulations Across Agencies	• Refinement of Classification and Categorization • Expansion of Patent Non-Disclosure Categories	• Establishment of Dedicated Support Systems
• Strengthening Research Security Obligations for Institutions	• Introduction of Partial Disclosure of Results • Standardization of System Management	• Government-Level Personnel Management Support

6 Conclusion

Even if regulations and guidelines are well-established, they lose their significance if researchers are unaware of or do not utilize them in their work. Therefore, periodic training on guidelines for protecting research assets and preventing conflicts of interest in national R&D projects is essential.

Additionally, it is recommended that a compendium of domestic and international research security cases be created and distributed for researchers and institutions, and a research security training portal be established and operated. This portal would provide integrated guidance on research security regulations, case studies, guidelines, educational materials, and relevant agencies.

By strengthening the management of security tasks for national research projects and research assets, it is possible to build a trustworthy research environment that considers the autonomy, openness, and balance of research while protecting national research assets.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. National Research Council of Science & Technology and National Intelligence Service (Industrial Secret Protection Center). *Guidelines for Research Security Management for Researchers*, p. 6. (March 2022).

2. Korea Institute of Science and Technology Evaluation and Planning. *Science and Technology Talent Policy Trends Brief 2021 No. 08: New Approaches to Research Security in the U.S.*, pp. 1-6. (2021).
3. Ministry of Science and ICT. *Policy Case Studies for Preventing Research Asset Leakage in International Research Cooperation*, pp. 2-11. (June 2023).
4. Hwang, I. Y., & Jeong, J. K. *Key Contents and Implications of the National R&D Foreign Funding Reporting System*, pp. 1-3. (March 7, 2024).

Forensic Analysis of Trust Wallet Artifacts in Browser Extension Environment

JiWon Bae[1][0000-0002-0469-1035] and Kiwook Sohn[2][0009-0000-5941-7319]

[1] Seoul National University of Science & Technology, Korea
[2] Seoul National University of Science & Technology, Korea
jwonb0116@seoultech.ac.kr, kiwook@seoultech.ac.kr

Abstract. The rapid expansion of the cryptocurrency market, especially through Bitcoin and Ethereum transactions, has significantly strengthened the DeFi (Decentralized Finance) sector. However, this growth has also increased the risks of illegal activities due to the high level of anonymity these platforms provide. In this study, we present a pioneering analysis of Trust Wallet's artifact collection and encryption processes. By extracting wallet addresses and asset IDs from Chrome cache data, we have identified critical artifacts that can trace illicit fund flows. Additionally, we uncovered the password-based encryption mechanism Trust Wallet uses to protect user information, enabling the development of decryption methods for password recovery during criminal investigations.
Our research marks a significant advancement in detecting and recovering criminal proceeds, offering a valuable resource for law enforcement agencies monitoring illegal financial activities in the cryptocurrency ecosystem. Moreover, our findings provide a foundation for forensic investigations aimed at preventing unauthorized withdrawals of illicit funds.

Keywords: Trust Wallet, Digital Forensic, Key Derivation, Encryption Process .

1 Introduction

As blockchain technology expands, the cryptocurrency market has rapidly grown, especially in Bitcoin and Ethereum transactions. While this growth has greatly benefited the DeFi (Decentralized Finance) sector by reducing intermediaries, it has also facilitated illegal activities due to the inherent anonymity of cryptocurrencies.

In Korea, the 'Nth Room' case highlighted the challenges of recovering illicit gains without private keys.[1] As cryptocurrency-related crimes increase, tracing illicit transactions has become crucial but remains difficult due to wallet security. Recovering criminal proceeds is particularly challenging without private keys. Since 2022, Korea has intensified efforts against drug-related offenses, increasing investigations into illegal drug transactions involving virtual assets and expanding specialized teams, leading to significant asset seizures. However, successful recovery often depends on the accused providing passwords or private keys, making decryption difficult without cooperation.[2][3]

This paper examines the artifact collection and encryption mechanisms of Trust Wallet to support cryptocurrency crime investigations. This analysis aids law enforcement in monitoring illicit funds, preventing withdrawals, and recovering criminal proceeds.

2 Related Work

Son Ji-Hoon et al. [3] analyzed MetaMask's levelDB to elucidate the mnemonic code decryption mechanism via user passwords, developing a tool for artifact extraction using levelDB decryption and regular expressions.
Holmes et al. [4] proposed cryptocurrency wallet forensic procedures using the NIST digital forensic model, covering memory process data collection, local file system artifacts, and key recovery techniques.

3 Collection and Analysis of Artifacts in Trust Wallet

3.1 Trust Wallet

Launched in 2017, Trust Wallet is a web3-based cryptocurrency wallet with over 100 million users. It supports browser extensions and mobile platforms (Android, iOS) and is compatible with Chrome, Edge, Brave, and Opera. Users can manage multiple cryptocurrencies like Bitcoin, Ethereum, Litecoin with a password and mnemonic phrase.

3.2 Analysis Environment and Tools

The experiments utilized the Sepolia Testnet provided by Ethereum. Table 1 summarizes the tools and analysis environment used.

Table 1. Summary of Tools and Analysis Environment Used

No.	Item	Detail	Version	No.	Item	Detail	Version
1	Target	Trust Wallet	2.17.0	5	Memory Dump Tool	Magnet RAM Capture	1.2.0.0
2	Operating System	Windows 11, RAM 32GB	-	6	IDE	Chrome Developer Tools	Chrome 126
3	Web Browser	Chrome	126.0.6478.128	7	Hex Editor	HxD	1.7.7.0
4	Cache Viewer	ChromeCacheView	2.47				

3.3 Cache Data Analysis

By analyzing cache data, the wallet address and asset ID involved in a transaction can be identified. The wallet address can be retrieved from Chrome's cache files located at '%USERPROFILE%\AppData\Local\Google\Chrome\UserData%PROFILE%\Cache\Cache_Data'. These .json files contain the recipient wallet address and asset ID. Trust Wallet uses HTTP GET requests, embedding information such as the wallet address and asset ID in the URL. Therefore, by analyzing cache data, it is possible to identify the wallet and asset involved in the transaction.

3.4 Web Storage Data Analysis

Further, to gather encryption-related data, it is necessary to acquire storage data from the web browser. By examining the web storage data, encryption-related data can be identified as shown in Fig. 1.


"4f8ca828-58df-4374-b646-c265c868c5b8":
activeAccounts: (124) [{…}, {…}, {…}, {

Upon password verification, the wallet's mnemonic phrase is generated according to the BIP standard. This mnemonic phrase is then used to derive the wallet's private and public keys. The password also encrypts the private key using the scrypt algorithm, with the derived key serving as the AES-128-CTR key for encryption. The detailed process is illustrated in Fig.4.

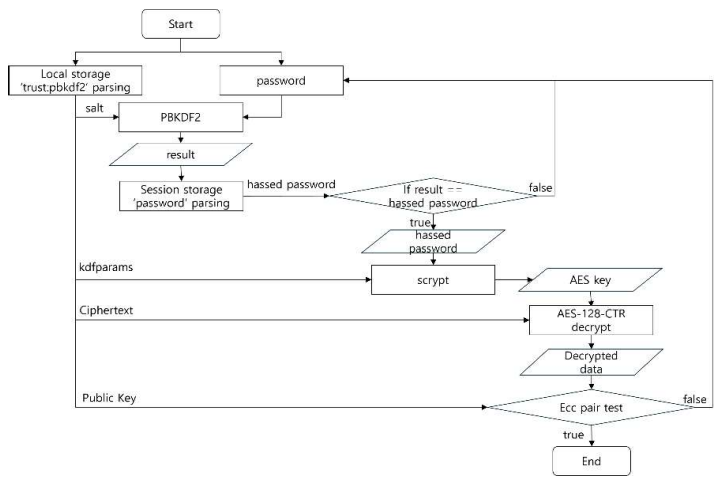


Fig. 4. Password verification process

4 Forensic Applications

The collected artifacts can be utilized in the investigative process to infer user behavior and assist in reconstructing the password cracking and verification processes for Trust Wallet. The collected artifacts can be utilized in the investigative process to infer user behavior and assist in reconstructing the password cracking and verification processes for Trust Wallet. To enhance the accuracy of the password verification process and increase the likelihood of identifying the correct key values, MAC value verification can be employed. Trust Wallet follows the BIP32 standard, where derived keys from each wallet undergo checksum verification to ensure integrity. This checksum verification, along with public key pair verification and MAC value verification, can be utilized to develop a decryption mechanism.

Such a decryption mechanism could be instrumental for law enforcement agencies in gaining access to the cryptographic wallets used by criminals, thereby facilitating investigations and the recovery of illicit funds.

5 Conclusion

In this study, we conducted cache data analysis, web storage data analysis, memory data analysis, and static analysis to collect artifacts from the Trust Wallet. These methods enabled us to identify critical data, including wallet addresses, asset IDs, encryption

keys, and mnemonic codes. Leveraging these findings in password cracking and verification processes can significantly enhance forensic investigations. This comprehensive approach will facilitate monitoring the flow of cryptocurrency involved in criminal activities and prevent the withdrawal of illicit funds, thereby improving forensic capabilities in cryptocurrency-related cases.

Future work will focus on collecting artifacts from Trust Wallet in mobile environments and developing an accelerated decryption process for derived private key generation.

Acknowledgments. A third level heading in 9-point font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Maeil business newspaper, `n번방의 지갑` 가상화폐...자백 없으면 환수 산넘어 산' https://www.mk.co.kr/news/society/9323014, last accessed 2024/07/27
2. News1, https://www.news1.kr/articles/?5180854 '다크웹 이용 마약거래 작년 한해만 1097명 적발...5년 전보다 13배 급증', last accessed 2024/07/27
3. Safetimes, '경찰청, 마약 사범 집중 단속에 5702명 검거' https://www.safetimes.co.kr/news/articleView.html?idxno=200722, last accessed 2024/07/27
4. Ji-hoon, S, Jeong-heum, P.: Analysis of Artifacts in the MetaMask Cryptocurrency Wallet, 16(4), 151-165. Digital Forensics Research (2022)
5. Holmes, A.: A framework for live host-based Bitcoin wallet forensics and triage", 44, 301486, Forensic Science International: Digital Investigation (2023)
6. Kim, J. E., Seo, S. H., Seok, B. J., Byun, H. S., & Lee, C. H. (2023). A Study on the Digital Forensics Artifacts Collection and Analysis of Browser Extension-Based Crypto Wallet. Journal of the Korea Institute of Information Security & Cryptology, 33(3), 471-485.

Optimizing HQC on Cortex-M4

DongCheon Kim[0009−0008−4541−1088] and
Seog Chung Seo*[0000−0001−8016−2808]

Department of Financial Information Security, Kookmin University,
Seoul, South Korea
{kindongsy,scseo}@kookmin.ac.kr

Abstract. To ensure quantum-safe communication, in 2022, NIST decided to continue PQC competition to select additional KEMs and DSAs and currently it is undergoing. Currently, the three algorithms under review in Round 4 are all code-based cryptosystems. Among them, HQC is based on the syndrome decoding problem through quasi-cyclic code and offers moderate advantages in terms of key size and performance compared to the other two algorithms. However, from an implementation perspective, there has not yet been an implementation on embedded devices, making accurate performance benchmarking on Cortex-M4 challenging. Additionally, the HQC reference code relies on specific external libraries (NTL, GMP, etc.), necessitating the removal of these dependencies for broader applicability across various environments. Therefore, this paper presents methods for removing such dependencies and optimizing key operations, specifically polynomial multiplication in binary fields. This resulted in performance improvements of ×11.02, ×11.20, and ×10.58 for HQC128, HQC192, and HQC256, respectively, compared to PQClean's multiplication operations. Furthermore, we applied FAFFT, the optimized polynomial multiplication method for binary fields used in the BIKE optimization research, achieving an 11.18% performance improvement over the NTL multiplication method.

Keywords: HQC · Cortex-M4 · Optimization · Binary Field

1 Introduction

To establish a secure communication environment in the quantum computing era, the NIST (National Institute of Standards and Technology) initiated the PQC (Post-Quantum Cryptography) competition in December 2016. In 2022, NIST selected four algorithms that passed Round 3 and proceeded with the standardization process for these algorithms in 2023 [1–3]. Additionally, to diversify the hard problems underlying PQC algorithms, NIST commenced Round 4 in 2022 [4]. The remaining three algorithms (HQC, BIKE, and Classic McEliece) are all based on error-correcting codes and are currently under review in Round

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) ((No. 2021-0-00796, Research on Foundational Technologies for 6G Autonomous Security-by-Design to Guarantee Constant Quality of Security, 100%) and Korea Evaluation Institute of Industrial Technology(KEIT) grant funded by the Korea government.

4. Research on PQC for resource-constrained embedded devices is actively being conducted. Until now, several PQC algorithms have been implemented and benchmarked on PQM4 project [5,6]. However, there has been no optimization research on HQC for the Cortex-M4. Therefore, this paper introduces a study on implementing the HQC submission, analyzing methods to make it applicable in diverse environments, including the Cortex-M4. Furthermore, it discusses and applies optimization techniques for polynomial multiplication over binary fields, which is a key operation in HQC, and presents performance benchmarks for these optimizations.

2 HQC

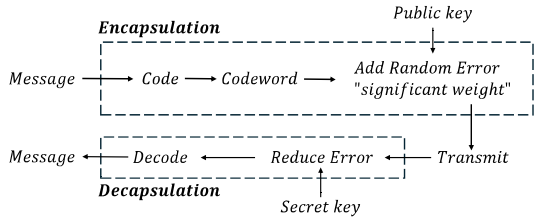


Fig. 1: Overview of HQC [7]

HQC (Hamming Quasi Cyclic) is a KEM (Key-Encapsulation Mechanism) algorithm based on the hardness of the syndrome decoding problem. HQC employs a concatenated code structure, which places two distinct codes (Reed-Muller code and Reed-Solomon code) in series. As illustrated in Fig. 1, the algorithm encrypts an input message using a codeword with errors based on the public key, and subsequently recovers the message by correcting the errors through decoding operations with the private key [8–10]. Additionally, the current HQC implementations submitted to NIST include a reference implementation using the Number Theory Library (NTL) for efficient computation, an optimized implementation using AVX2 to enhance key operations, an additional implementation applying constant-time operations, and a hardware implementation for the Artix-7 FPGA environment.

3 Investigation of HQC Implementation on Cortex-M4

PQM4 The PQM4 project is a framework for benchmarking implementations of PQC KEM and DSA schemes on Arm Cortex-M4 based embedded systems [11–13]. It is designed to compare various PQC algorithms under the same criteria, such as memory usage and performance. However, it does not include all the algorithms submitted to the NIST PQC competition. For example, HQC is not integrated into the current PQM4 project because it uses external libraries such as NTL, OpenSSL, and GMP (GNU Multi-Precision) [11]. Therefore, to analyze HQC using the PQM4 framework, one must use the PQClean project, which organizes NIST PQC implementations [14,15].

Table 1: Performance Comparison of HQC Algorithm (Mega Clock Cycle, -O3)

Alg.	HQC128			HQC192			HQC256		
	I	II	III	I	II	III	I	II	III
Mul.	51.50	51.50	**4.67** (×**11.02**)	158.25	158.25	**14.13** (×**11.20**)	289.32	289.32	**27.39** (×**10.56**)
*KG	52.23	51.96	**5.70** (×**9.12**)	160.01	169.38	**17.04** (×**9.35**)	293.24	291.91	**33.15** (×**8.81**)
KGH	0.22	0.23	**0.23**	0.42	0.43	**0.43**	0.65	0.66	**0.66**
ENC	104.71	104.12	**11.59** (×**8.98**)	320.25	318.94	**34.25** (×**9.31**)	586.45	583.82	**66.29** (×**8.81**)
ENCH	0.80	0.80	**0.80**	1.57	1.58	**1.58**	2.47	2.48	**2.48**
DEC	158.15	157.38	**18.59** (×**8.47**)	481.82	480.03	**52.99** (×**9.06**)	883.04	879.41	**103.11** (×**8.53**)
DECH	0.84	0.84	**0.84**	1.63	1.63	**1.63**	2.55	2.55	**2.55**

* KG : Key Gen, ENC : Encapsulation, DEC : Decapsulation, suffix H : Hash function
I PQClean implementation
II Latest Reference & PQClean main operations
III Latest Reference & NTL main operations

PQClean Currently, PQClean includes the version of the HQC implementation submitted on April 30, 2023, which differs from the latest submission on February 23, 2024. The differences between the PQClean version and the latest version arise from changes in the modulo operation (Barrett reduction) and the process of computing random data, leading to the generation of different test vectors.

Therefore, for an accurate performance comparison, the latest version was first integrated into the PQM4 and PQClean frameworks, and the operations that could not be performed due to external libraries were implemented using PQClean. All performance analyses targeted the NUCLEO-L4R5ZI board [16], selected as the default board for NIST Round 4, which uses an ARM 32-bit Cortex-M4 MCU (Micro Controller Unit). This board has a maximum clock speed of 120MHz, 2048KB of Flash memory, and 640KB of SRAM. The clock configuration for the computations was set to 20MHz, as suggested by the PQM4 framework.

4 Optimization of HQC on Cortex-M4

NTL To alleviate the computational load of polynomial multiplication in binary fields, which performance profiling results show accounts for over 97% of the performance overhead in HQC algorithm, HQC applies the multiplication method from the NTL (Number Theory Library) [17]. The NTL achieves optimization for polynomial multiplication through branching statements based on the input length of the polynomials, using the Karatsuba 2-way recursive algorithm. However, due to the dependency on external libraries, the current PQM4 does not support this operation. In this study, to achieve optimal implementation on Cortex-M4, we analyzed and debugged the HQC reference code to remove the library dependency and incorporate this multiplication method. To apply the NTL multiplication method to HQC on Cortex-M4, we removed operations that

Table 2: Comparison of Multiplication Performance (Clock Cycle, -O3)

Alg.	PQClean Mul.	NTL Mul.	FAFFT
HQC128	51,505,609	4,676,551	**4,206,319**
Improvement	-	×11.02	×**12.25**

do not affect the multiplication results and selected the multiplication with the lowest computational load among various base multiplications to achieve additional performance improvements. Based on the comparison of multiplication algorithms, we performed a performance analysis for all security levels of HQC algorithms. For a fair comparison, we used the PQM4 framework to measure performance, analyzing three versions as shown in Table 1. This table illustrates the comparison of multiplication implementations and the performance improvements of the algorithms achieved through these optimizations.

FAFFT FAFFT (Frobenius Additive Fast Fourier Transform) is an optimization method for polynomial multiplication algorithms, developed to utilize the widely known FFT in binary fields through iterative XOR (addition) [18, 19]. It is an advanced version of Additive FFT, enhanced with Frobenius mapping. FAFFT first uses the Cantor basis to convert the basis, leveraging the recursive nature of FFT that reduces the overall computational load by repeating transformation layers. Subsequently, it performs FFT transformations using the butterfly technique, adapted for operations in binary fields. Frobenius mapping is then applied to implement Truncated FFT for specific evaluation points of FFT. Finally, pointwise multiplication completes the FAFFT transformation, and the inverse process is carried out to finish the polynomial multiplication. This FAFFT optimization method has been applied to the BIKE algorithm, which is under review in Round 4, and is implemented in the PQM4 framework [20, 21]. Similarly, noting that HQC also operates in the binary field like BIKE, we applied the optimized FAFFT used in BIKE to HQC and successfully implemented it at security level 1. As shown in Table 2, this implementation achieved an 11.18% improvement compared to NTL multiplication.

5 Conclusion

As a result, we achieved performance improvements of ×11.02, ×11.20, and ×10.58 for HQC128, HQC192, and HQC256, respectively, compared to the PQClean multiplication. Consequently, key generation, encapsulation, and decapsulation operations saw performance gains of up to ×9.35, ×9.31, and ×9.06, respectively. Furthermore, by applying FAFFT, which was used in the optimization of BIKE, we further optimized the binary field polynomial multiplication in HQC, achieving an 11.18% improvement over the NTL implementation. In future research, we aim to apply FAFFT to all security levels of HQC and perform low-level optimizations using assembly to achieve maximum efficiency on the Cortex-M4. Ultimately, the goal of this research is to compare the Karatsuba and Toom-Cook algorithms presented in the HQC submission with the additional optimization method, FAFFT, to determine the most efficient algorithm for HQC operations across different environments.

References

1. National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard (initial public draft). Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 203 (Initial Public Draft), August, 2023, U.S. Department of Commerce, 2023.
2. National Institute of Standards and Technology. Module-lattice-based digital signature standard (initial public draft). Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 204 (Initial Public Draft), August, 2023, U.S. Department of Commerce, 2023.
3. National Institute of Standards and Technology. Stateless hash-based digital signature standard (initial public draft). Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 205 (Initial Public Draft), August, 2023, U.S. Department of Commerce, 2023.
4. NIST, the US National Institute of Standards and Technology. Post-quantum cryptography standardization project. https://csrc.nist.gov/Projects/post-quantum-cryptography.
5. Jieyu Zheng, Hong Zhang, Le Tian, Zhuo Zhang, Hanyu Wei, Zhiwei Chu, Yafang Yang, and Yunlei Zhao. Espm-d: Efficient sparse polynomial multiplication for dilithium on arm cortex-m4 and apple m2. *arXiv preprint arXiv:2404.12675*, 2024.
6. Saeed Aghapour, Kasra Ahmadi, Mila Anastasova, Mehran Mozaffari Kermani, and Reza Azarderakhsh. Puf-kyber: Design of a puf-based kyber architecture benchmarked on diverse arm processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.
7. Guillaume Goy, Antoine Loiseau, and Phlippe Gaborit. Estimating the strength of horizontal correlation attacks in the hamming weight leakage model: A side-channel analysis on hqc kem. In *WCC 2022: The Twelfth International Workshop on Coding and Cryptography*, page WCC_2022_paper_48, 2022.
8. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. Hamming quasi-cyclic (hqc). *NIST PQC Round*, 2(4):13, 2018.
9. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. Hqc. https://pqc-hqc.org/index.html.
10. Maximilian Schöffel, Johannes Feldmann, and Norbert Wehn. Efficient hardware implementation of constant time sampling for hqc. *arXiv preprint arXiv:2309.16493*, 2023.
11. Matthias J Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and benchmarking nist pqc on arm cortex-m4. 2019.
12. Matthias J Kannwischer, Markus Krausz, Richard Petri, and Shang-Yi Yang. pqm4: Benchmarking nist additional post-quantum signature schemes on microcontrollers. *Cryptology ePrint Archive*, 2024.
13. Matthias J Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. Pqclean project. https://github.com/mupq/pqm4.
14. Matthias J Kannwischer, Peter Schwabe, Douglas Stebila, and Thom Wiggers. Improving software quality in cryptography standardization projects. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 19–30. IEEE, 2022.
15. Matthias J Kannwischer, Peter Schwabe, Douglas Stebila, and Thom Wiggers. Pqclean project. https://github.com/PQClean/PQClean.

16. STMicroelectronics. Stm32 nucleo-144 development board with stm32l4r5zi mcu, supports arduino, st zio and morpho connectivity. https://www.st.com/en/evaluation-tools/nucleo-l4r5zi.html.
17. Victor Shoup. Ntl: A library for doing number theory. https://libntl.org/.
18. Shuhong Gao and Todd Mateer. Additive fast fourier transforms over finite fields. *IEEE Transactions on Information Theory*, 56(12):6265–6272, 2010.
19. Wen-Ding Li, Ming-Shing Chen, Po-Chun Kuo, Chen-Mou Cheng, and Bo-Yin Yang. Frobenius additive fast fourier transform. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 263–270, 2018.
20. Ming-Shing Chen, Tung Chou, and Markus Krausz. Optimizing bike for the intel haswell and arm cortex-m4. *Cryptology ePrint Archive*, 2021.
21. Ming-Shing Chen, Tim Güneysu, Markus Krausz, and Jan Philipp Thoma. Carryless to bike faster. In *International Conference on Applied Cryptography and Network Security*, pages 833–852. Springer, 2022.

A Study of V2V Certificate Management for PQC Migration

Jung-Hun Kang[1][0009−0002−3075−9440], Soojin Lee[1][0000−0003−1690−8577], and Seung-Hyun Seo[12][0000−0002−1150−7080]*

[1] The Department of Electronic & Electrical Engineering, Graduate School, Hanyang University, kjh980922@hanyang.ac.kr, tssn195@hanyang.ac.kr
[2] The Department of Electrical Engineering, Hanyang University (ERICA), Ansan, 15588, KOREA
seosh77@hanyang.ac.kr

Abstract. Intelligent Transportation Systems (ITS) enable communication between various entities, such as vehicles and Roadside Units (RSUs), to enhance safe and efficient transportation systems. To ensure secure vehicle-to-everything (V2X) communication, the IEEE 1609.2 standard for V2X communication, which relies on Public Key Infrastructure (PKI)-based certificates, has been proposed. In preparation for the era of quantum computing, Studies focused on applying quantum-resistant cryptography to V2V certificates have been conducted recently. This paper introduces recent research trends on the integration of PQC with V2X certificate management, including Butterfly key expansion, certificate transmission, and certificate revocation.

Keywords: Post Quantum Cryptography · PQC migration · V2X certificate management.

1 Introduction

Intelligent Transportation Systems (ITS) integrate information and communication technology into vehicles and transportation infrastructure to provide various services such as accident prevention and automated traffic control systems, thereby enhancing traffic efficiency and safety. The ITS market is growing at an annual rate of over 8.6% and is expected to reach $26.31 billion by 2026.[3] Institute of Electrical and Electronics Engineers (IEEE) has proposed the IEEE 1609 family of standards to ensure efficient communication among various ITS entities, including autonomous driving, platoon vehicles, and charging stations.[4] Specifically, IEEE 1609.2 provides Public Key Infrastructure (PKI)-based security services for ITS applications.

* This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS202300225201, Development of Control Rights Protection Technology to Prevent Reverse Use of Military Unmanned Vehicles)

On the other hand, the growing power of quantum computers is a serious threat to computer security. Discrete logarithm problem (DLP)-based public-key algorithms are at risk from shor's algorithm[12] with improvement of quantum computers. To address these threats, the National Institute of Standards and Technology (NIST) has conducted a competition to select Post-Quantum Cryptography (PQC) algorithms. Ultimately, NIST has chosen four algorithms and recommended a timeline for migration. Furthermore, NIST recommends completing the migration by the mid-2030s to ensure systems are secure against quantum threats.[6]

However, according to [15], adopting the larger public key and signature sizes of the NIST finalist PQC algorithms in V2X systems could lead to increased communication overhead and decreased efficiency. This is a significant concern, given the intricate nature of V2X environments, which involve numerous interactions between entities and applications.

We focus on certificate management in V2X security, particularly on Butterfly key expansion, certificate transmission, and certificate revocation. We provide a comprehensive introduction to certificate management for V2X, review various studies proposed for PQC migration, and offer insights and improvements needed for effective PQC migration.

2 V2X Certificate management standards

For secure V2X operations, IEEE and ETSI have proposed standards for V2X certificate management.[4][2] Specifically, the recent final standard released by the EU defines the roles of participants (i.e., AT, AA) in a PKI-based authentication system.[1] It specifies the lifecycle of public key-based certificates and uses anonymous certificates to protect privacy, ensuring that the certificates do not contain the actual identity of the vehicle. Additionally, butterfly key expansion and certificate revocation are utilized to efficiently and securely manage pseudonym certificates.

3 Processes in certificate management for PQC migration

In this chapter, we briefly explain three key processes in certificate management according to the design of EU standard[1] and introduce studies on PQC migration for each process. A summary of the papers we reviewed is shown in Table 1.

3.1 Butterfly key expansion for PQC certificate

V2X certificate management standards aim to provide secure and reliable communication in V2X communication. However, there are potential privacy issues associated with V2X certificates, as malicious parties can track vehicles and individuals by using certificate information such as unique certificate identifiers.

To address these concerns, pseudonym certificates are used to preserve the privacy of the owner of a certificate. According to [1], which is a certificate policy for the deployment and operation of European cooperative intelligent transport systems (C-ITS), users should not use issued certificates for more than one week and issue new certificates. Also, the standard applies the Butterfly key expansion to facilitate efficient key generation when creating multiple certificates for privacy protection. The Butterfly key expansion enables an entity to efficiently generate the required number of certificates with a single request. The existing Butterfly key expansion was designed based on Elliptic Curve Cryptography (ECC). Recently, research has been conducted to develop a butterfly key expansion protocol based on PQC to protect against potential quantum computing attacks.

Salman et al. [10] implemented NTRU-based Butterfly key expansion process, which is one of the PQC candidates. They compared it with the existing ECC-based process in terms of time and energy consumption by implementing it on embedded devices.

Barreto et al. [7] proposed a new Butterfly key expansion process utilizing PQC signature algorithms selected in the NIST Final Round. Additionally, since the existing Butterfly key expansion required two public/private key pairs, the proposed scheme applies unified butterfly key expansion, which unifies this into a single pair. It achieved up to a 50% improvement in performance compared to the Security Cre-dential Management System (SCMS), one of the vehicular public key infrastructure standards of the U.S.

3.2 Certificate transmission improvement

According to the [9], PQC candidates have larger key, signature, or ciphertext sizes than classical cryptography like RSA or ECC. As the volume of data required for V2X increases, the communication overhead also rises, leading to increased network latency and potentially causing the entire authentication system to malfunction. However, transitioning to PQC is essential to counter quantum computing attacks. Therefore, various studies are being conducted to apply PQC efficiently in the certificate transmission process.

Twardokus et al. [13] showed that unnecessary certificate data is transmitted redundantly in current V2V (Vehicle-to-Vehicle) communication. To solve the issue, the authors proposed a machine learning-based certificate transmission scheduling technique considering PQC migration. Additionally, they designed a partially hybrid authentication protocol using ECDSA and NIST standardized signature algorithms (Falcon, Dilithium, Sphincs+) and XMSS.

Kim et al. [8] proposed a method to integrate PQC signature algorithms into the WAVE protocol. Since the size of the Dilithium signature exceeds the WAVE protocol's maximum payload size of 2000 bytes, WAVE messages need to be split for transmission. They proposed a signature split method that eliminates redundant metadata during this splitting process.

3.3 Certificate revocation process

Standards in the U.S. and Europe introduce certificate revocation processes to address misbehaving entities in PKI-based ITS by revoking the credentials of those delivering incorrect information.[5][1] Both ITS standards employ CRL (Certificate Revocation Lists) and efficiently handle revocation via a blacklist with each signature verification. However, existing standards face challenges as anonymous certificates have fixed validity periods. Expired certificates are added to the CRL, leading to a significant increase in CRL size over time. This large size can impact network latency and, in the worst case, may result in the revocation of certificates that do not necessarily need to be revoked.

Verheul et al. [14] proposed a pre-issued single pseudonym certificate for authentication when an activation code is triggered. They decrease the size of the stored pseudonym certificates for five years by 22.2%. Compared to [1].

Scopelliti et al. [11] introduces a self-revocation mechanism to ensure credential revocation with low network overhead and good scalability in a virtual 5G-Edge deployment scenario. Unlike previous approaches, this mechanism does not require synchronized time and addresses delays or drops in revocation requests. It guarantees a fixed upper bound on revocation time. Additionally, it provides formal verification through the TAMARIN prover and deployable artifacts.

Table 1. Summery of works for key process of certificate management in V2X

Title	Motivation	PQC supported	Method
[10]	Butterfly key expansion	Y	Implement PQC butterfly key expansion process on embedded device
[7]		Y	Unify public/private key pair for butterfly key expansion
[8]	Certificate transmission improvement	Y	Divide the large signature data into smaller blocks
[13]		Y	Certificate transmission scheduling technique, Partially hybrid authentication protocol
[14]	Certificate revocation	N	Issue certificates first and activate later
[11]		N	Asynchronized self-revocation mechanism

4 Conclusion

This paper reviewed techniques such as butterfly key expansion, certificate revocation, and certificate transmission for applying PQC to the pseudonym certificates used in V2X operations. We found a lack of research and experiments on applying PQC to certificate revocation, which is essential for the continuous operation of V2X networks. Various studies indicate the need for PQC migration. It is clear that introducing PQC algorithms, which exceed the size supported by the current WAVE protocol, is challenging to implement on existing hardware. This can significantly impact the resource-constrained V2X communication environment, necessitating optimization efforts. Therefore, it is crucial to develop optimization solutions for PQC migration from a network perspective.

References

1. Certificate policy for deployment and operation of european cooperative intelligent transport systems (c-its), https://cpoc.jrc.ec.europa.eu/data/documents/E01941_C-ITS_Certificate_Policy_Release_3_0_FINAL.pdf, last accessed 2024/07/27
2. ETSI TR 103 415, https://www.etsi.org/deliver/etsi_tr/103400_103499/103415/01.01.01_60/tr_103415v010101p.pdf, last accessed 2024/07/27
3. Intelligent transportation systems market size, trend and analysis by region, system type, application and segment forecast to 2026, https://www.globaldata.com/store/report/intelligent-transportation-systems-market-analysis/, last accessed 2024/07/27
4. Ieee approved draft standard for wireless access in vehicular environments–security services for applications and management messages. IEEE Std 1609.2-2022 (Revision of IEEE Std 1609.2-2016) pp. 1–349 (2023). https://doi.org/10.1109/IEEESTD.2023.10075082
5. J2735 Dedicated Short Range Communications (DSRC) Message Set Dictionary (2023), https://www.sae.org/standards/content/j2735_202309/, sAE International Standard
6. Attema, T.: Post-quantum cryptography (pqc) in vehicular communication (2023), https://publications.tno.nl/publication/34641918/oicFLj/attema-2023-pqc.pdf, last accessed 2024/07/27
7. Barreto, P.S., Ricardini, J.E., Simplicio Jr, M.A., Patil, H.K.: qscms: Post-quantum certificate provisioning process for v2x. Cryptology ePrint Archive (2018)
8. Kim, Y., Seo, S.C.: Signature split method for a pqc-dsa compliant with v2v communication standards. Applied Sciences **13**(10), 5874 (2023)
9. Lonc, B., Aubry, A., Bakhti, H., Christofi, M., Mehrez, H.A.: Feasibility and benchmarking of post-quantum cryptography in the cooperative its ecosystem. In: 2023 IEEE Vehicular Networking Conference (VNC). pp. 215–222. IEEE (2023)
10. Salman, A., Blankinship, Z.: Implementing butterfly key expansion using post-quantum algorithms. In: Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 1. pp. 507–516. Springer (2022)
11. Scopelliti, G., Baumann, C., Alder, F., Truyen, E., Mühlberg, J.T.: Efficient and timely revocation of v2x credentials. In: Proceedings of the 2024 Network and Distributed System Security (NDSS) Symposium, ser. NDSS. vol. 24 (2024)
12. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. Ieee (1994)
13. Twardokus, G., Bindel, N., Rahbari, H., McCarthy, S.: When cryptography needs a hand: Practical post-quantum authentication for v2v communications. Cryptology ePrint Archive (2022)
14. Verheul, E., Hicks, C., Garcia, F.D.: Ifal: Issue first activate later certificates for v2x. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 279–293. IEEE (2019)
15. Yoshizawa, T., Preneel, B.: Post-quantum impacts on v2x certificates–already at the end of the road. In: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring). pp. 1–6. IEEE (2023)

Security Evaluation on the KpqC Round 2 Lattice-based Candidates*

Sumin Park, Eunmin Lee, Hyun-A Noh, Hyojung Yoon, Joohee Lee

Sungshin Women's University, Seoul, Republic of Korea
{20200917, 20211089, 20211056, 20200940, jooheelee}@sungshin.ac.kr

Abstract. In Republic of Korea, the KpqC (Korean PQC) competition to develop new algorithms for efficient Post-Quantum Cryptography (PQC) as national standards is ongoing and in its second round. In this paper, we evaluate the security of four lattice-based algorithms {NTRU+, SMAUG-T, HAETAE, NCC-Sign} in the second round of the KpqC competition with Albrecht's lattice estimator. We compare the estimated security in bits for the respective parameters of the four schemes with the claimed security in their official document and report some discrepancies.

1 Introduction

The KpqC competition [5] has been held since 2022 with the aim of establishing Post-Quantum Cryptography (PQC) standards in the Republic of Korea. The KpqC competition Round 1 started in November 2022 and ended in November 2023. As of Round 2, which began in April 2024 and is ongoing, there are two lattice-based candidates in the digital signature category and two in the Public-key Encryption and Key-establishment (KEM) category, respectively, described as follows.

NTRU+. NTRU+ is an algorithm that enhances the performance of the current NTRU scheme. [9]. It employs the approach of creating NTT (Number Theory Transform)-compatible settings for NTRU, as introduced in NTTRU [12] and NTRU-B. [8]. The security of NTRU+ relies on the NTRU and RLWE problems.

SMAUG-T. SMAUG-T is built on the hardness of the MLWE and MLWR problems. It is developed by merging two schemes SMAUG and TiGER of the first round of the KpqC competition. Its design goals are achieving small ciphertext and public key, computational cost, and resistance to side-channel attacks.

HAETAE. HAETAE is designed based on MLWE and MSIS. It uses the Fiat-Shamir with Aborts paradigm [10, 11] and a bimodal distribution introduced in the rejection sampling of BLISS signatures [7]. Additionally, it utilizes a hyperball uniform distribution to decrease the size of the signature, albeit at the cost of speed compared to Dilithium.

* This work is the result of commissioned research project supported by the affiliated institute of ETRI [2024-035].

NCC-Sign. NCC-Sign is an algorithm that puts together the design rationale of CRYSTALS-Dilithium and NTRU prime [4]. NCC-Sign also employs the Fiat-Shamir with Aborts paradigm, but it relies on a non-cyclotomic polynomial ring rather than using a cyclotomic polynomial ring of $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$. Polynomial multiplication is performed using the Toom-Cook method, which is a type of divide-and-conquer strategy.

In this paper, We analyze four lattice-based schemes {SMAUG-T, NTRU+, HAETAE, NCC-Sign} that were selected in the second round of the KpqC competition based on LWE/LWR utilizing the Lattice Estimator proposed by M. Albrecht [2]. We compare the outcomes of our security analysis with the asserted security in the official proposal documents under the same assumptions.

2 The LWE and LWR Problem

LWE. [14] Let $m, n, q \in \mathbb{N}$. An LWE distribution $A^{LWE}_{m,n,q,\chi}(s)$ consisting of m samples $\{(\vec{a}_i, b_i)\}_{i=1}^m$ can be obtained by computing the equation

$$b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i \mod q$$

where $\vec{a}_i \in \mathbb{Z}_q^n$ is an uniform random vector, $\chi \in \mathbb{Z}$ is an error distribution for a small error $e_i \in \mathbb{Z}$ and $\vec{s} \in \mathbb{Z}_q^n$ is a secret vector.

The LWE problem has two versions: The decision LWE problem asks to distinguish $\{(\vec{a}_i, b_i)\}_{i=1}^m$ either from $A^{LWE}_{m,n,q,\chi}$ or from the uniform distribution. The search LWE problem asks to find $\vec{s} \in \mathbb{Z}_q^n$ sampled from $A^{LWE}_{m,n,q,\chi}(s)$.

LWR. [13] The LWR problem hides the secret by replacing errors sampled from discrete Gaussian distributions with deterministic errors by rounding procedure $(\lfloor \cdot \rceil)$ to the nearest integers. Let $m, n, q, p \in \mathbb{N}$. An LWR sample $\{(\vec{a}_i, b_i)\}_{i=1}^m$ from distribution $A^{LWR}_{m,n,q,p}(\vec{s})$ can be obtained by computing

$$b_i = \lfloor (p/q) \cdot (\langle \vec{a}_i, \vec{s} \rangle \mod q) \rceil.$$

It also has decision and search versions.

3 Security Analysis Methods

3.1 Time Complexity Estimation of the BKZ algorithm

The BKZ algorithm [6] is a lattice basis reduction algorithm used to find a short basis within a lattice. The principle of the BKZ algorithm is to repeatedly apply the SVP solver, an algorithm that finds the shortest vector, for a sub-lattice of dimension β, smaller than that of a given lattice. At this time, when the dimension of the sub-lattice to which the SVP solver is applied is β, the BKZ is called β-BKZ, and the sub-lattice is called a block.

The Core-SVP model [3] serves as a measurement framework for estimating the time complexity of the BKZ algorithm from a conservative standpoint.

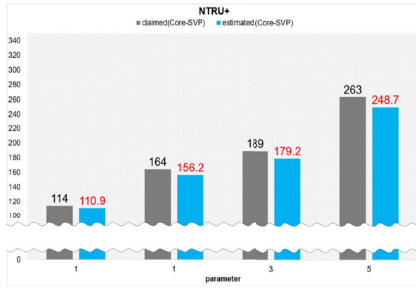
When estimating the time complexity of the BKZ algorithm using the Core-SVP model, the time complexity of β-BKZ is given as $2^{c \cdot \beta}$. This represents a lower bound for the time complexity of a single application of the SVP solver, which is approximately $2^{c \cdot \beta + o(\beta)}$, where c is a constant within the range $[0, 1]$. We use $c = 0.292$ for the classical security.

3.2 Lattice Estimator

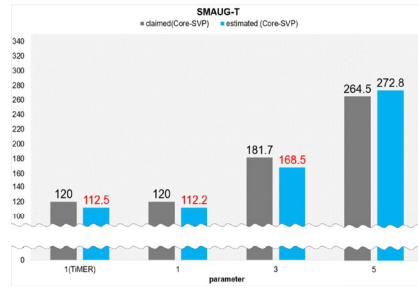
For the security analysis of LWE/LWR, we use M. Albrecht's Lattice Estimator [1]. The Lattice Estimator is an open-source Sage tool that determines the attack complexities and additional parameters necessary for an attack by using LWE/LWR parameters as input values. The lattice estimator shows the costs of LWE primal attack (`usvp`, `bdd`), LWE dual attack (`dual`), coded-BKW for LWE (`bkw`), Arora-GB (`arora-gb`) and hybrid attacks. The recent attacks such as Reconciliation Attack, May's Meet LWE attack and noise amplification attack are not considered in the lattice estimator.

4 Security Analysis on the KpqC Round 2 LWE/LWR-based Algorithms

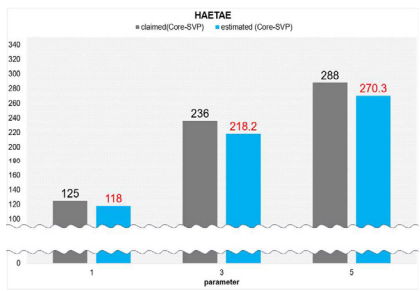
- For NTRU+1152, the estimated value was 248.7-bit, which is 14.3 bits less than the claimed value, showing the largest discrepancy. NTRU+576, NTRU+768, and NTRU+864 also show lower values in security by 3.1 bits, 7.8 bits, and 9.8 bits than the claimed security, respectively.
- The estimated values for SMAUG-T parameters TiMER, SMAUG-T128, and SMAUG-T192 are 112.5-bit, 112.2-bit, and 168.5-bit, respectively, which are 7.5∼13.2-bit less than the claimed. In the case of SMAUG-T256, the estimated result was 272.8-bit, which is 8.3-bit larger than the claimed security of 264.5-bit.
- HAETAE120 was measured at 118.0-bit, which is 7 bits less than the claimed value. HAETAE180 and HAETAE260 have estimated values of 218.2-bit and 270.3-bit, respectively, showing a slightly larger discrepancy of 17.7∼17.8 bits less than the claimed values.
- In the case of NCC-Sign, when measured without using Core-SVP, it was observed that the estimated values were larger than the Claimed values for three types of parameters ($\eta = 1$, $\eta = 2$, and Trinomial).
- When measured using Core-SVP, for NCC-Sign (η =1), Security Level V showed a value of 282.7-bit, which is 30.5 bits larger than the claimed value, while Security Levels I and III showed values of 125.4-bit and 181-bit, respectively, which are less than the claimed values. For NCC-Sign (η =2), the claimed values for each security level are 143-bit, 207-bit, and 279-bit, with the estimated values being smaller at 133.2-bit, 192.4-bit, and 257.6-bit, respectively. In the case of the Trinomial, the estimated values were 6.8 to 18.6 bits less than the claimed values. Specifically, the Security Level V parameter shows a significant discrepancy between the estimated value and the claimed one by 18.6 bits.



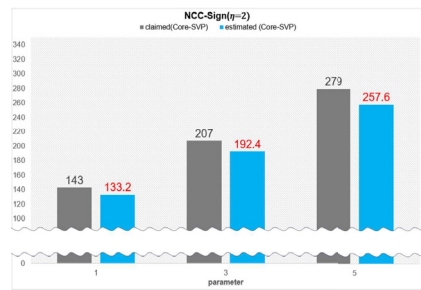
(a) Comparison of the claimed security and estimated results measured with and without the Core-SVP model for NTRU+ parameters



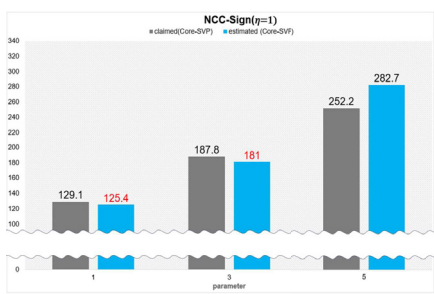
(b) Comparson of the claimed security and estimated results for SMAUG-T parameters



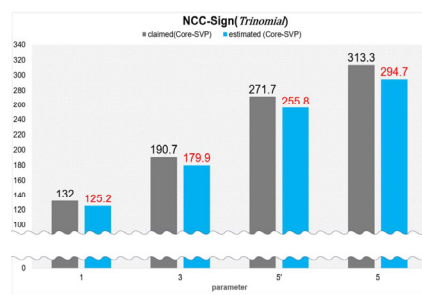
(c) Comparison of the claimed security and estimated results for HAETAE parameters



(d) Comparison of the claimed security and estimated results measured with and without the Core-SVP model for NCC-Sign ($\eta = 2$) parameters



(e) Comparison of the claimed security and estimated results measured with and without the Core-SVP model for NCC-Sign ($\eta = 1$) parameters



(f) Comparison of the claimed security and estimated results measured with and without the Core-SVP model for NCC-Sign Trinomial parameters

References

[1] Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving usvp and applications to lwe. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 297–322. Springer International Publishing, Cham (2017)

[2] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Cryptology ePrint Archive, Paper 2015/046 (2015), https://eprint.iacr.org/2015/046, https://eprint.iacr.org/2015/046

[3] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key Exchange—A new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343. USENIX Association, Austin, TX (Aug 2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[4] Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: Ntru prime: reducing attack surface at low cost. Cryptology ePrint Archive, Paper 2016/461 (2016), https://eprint.iacr.org/2016/461, https://eprint.iacr.org/2016/461

[5] Center, K.R.: Kpqc competition round 1, available from: https://www.kpqc.or.kr/competition.html [last accessed June 2023]

[6] Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. pp. 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[7] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Annual Cryptology Conference. pp. 40–56. Springer (2013)

[8] Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient ntru instantiations. In: IACR International Conference on Public-Key Cryptography. pp. 65–94. Springer (2023)

[9] Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International algorithmic number theory symposium. pp. 267–288. Springer (1998)

[10] Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 598–616. Springer (2009)

[11] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)

[12] Lyubashevsky, V., Seiler, G.: Nttru: truly fast ntru using ntt. Cryptology ePrint Archive (2019)

[13] Pointcheval, D., Johansson, T.: Advances in cryptology – eurocrypt 2012 : 31st annual international conference on the theory and applications of cryptographic techniques, cambridge, uk, april 15-19, 2012. proceedings. Lecture notes in computer science, 7237, EUROCRYPT (31st : 2012 : Cambridge, England), Springer, Berlin (2012)

[14] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324

Comparison of Voice Feature Extraction Techniques for Deepfake Detection

Seong-Bean Park[1], Jeong-Hyun Sim[2], and Hyun-Min Song[3]*

[1] Dankook University, Student
seongbeanpark@dankook.ac.kr
[2] Dankook University, Student
tlawjdgus0425@dankook.ac.kr
[3] Dankook University, Professor
hyunminsong@dankook.ac.kr

Abstract. With the rapid advancement of generative AI and the release of various generative AI services, the accessibility to creating deepfake media content has increased. Sophisticatedly crafted deepfake audio content has become difficult for humans to distinguish, leading to various misuse cases such as audio phishing and copyright infringement, thereby heightening the need for deepfake audio detection technology. In this study, we performed signal analysis using the CQCC, LFCC, and MFCC and extracted the characteristic features from the human's speaking audio data and the fake speaking audio data generated by generative AI models. We utilized an LSTM based detection model, known for its robust performance in time-series analysis, to conduct comparative experiments on the efficacy of signal analysis techniques and the detection performance of two forms of manipulated speech: TTS and VC, based on voice feature extraction techniques.

Keywords: AI security · Deepfake audio detection · Audio signal analysis

1 Introduction

The advancement and widespread use of generative AI technology has made it easy for individuals to create compelling deepfake content. Deepfake audio content, in particular, poses a significant risk in today's society. It has been linked to various social issues, including financial fraud, copyright violations, and the proliferation of misinformation [1]. Consequently, there is a growing demand for robust deepfake audio detection technology to address this challenge.

Our study focuses on using CQCC (Constant Q Cepstral Coefficient), LFCC (Linear-Frequency Cepstral Coefficients), and MFCC (Mel-Frequency Cepstral Coefficients), widely used voice feature extraction techniques in speech recognition, to effectively identify advanced deepfake audio. We harness the LSTM (Long Short-Term Memory) structure, known for its exceptional performance

* Corresponding author.

in time series analysis, as the key architecture for the deepfake audio detection model. Our proposed model establishes the effectiveness of CQCC, LFCC, and MFCC in detecting deepfake audio through two types of tampered audio detection experiments using the LSTM-based deepfake audio detection model.

2 Related Research

Yishuang Ning *et al.* demonstrated through experiments that models utilizing artificial neural networks such as DNN (Deep Neural Network), RNN (Recurrent Neural Network), and Transformer can produce more sophisticated deepfake audio data compared to traditional models like HMM (Hidden Markov Model) [2]. Zaynab Almutairi *et al.* discussed cases of financial fraud, privacy invasion, defamation, and public safety threats caused by deepfake audio. They emphasized the necessity of deepfake audio detection technology, reviewed current deepfake audio detection methods being researched, and suggested future directions [3]. Xinhui Zhou *et al.* compared the performance of MFCC and LFCC for speaker recognition, showing that LFCC is more effective in most environments and particularly excels in recognizing female speakers [4].

3 Deepfake Audio Detection Model

In this paper, CQCC, LFCC, and MFCC features are extracted from both genuine samples and deepfake audio samples. Using these features, an LSTM-based detection model performs binary classification to determine whether the input audio sample is a human-recorded genuine or a fake generated by deepfake audio.

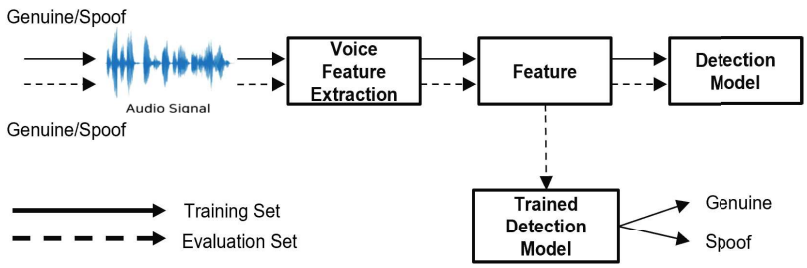


Fig. 1. Model structure proposed in the study.

During the experiments, the deepfake audio detection model architecture is illustrated in Fig. 1. The voice features are obtained from the training dataset using frequency analysis methods. These extracted features are then input into an LSTM-based deepfake audio detection model for the purpose of classifying a sample as genuine or spoof.

3.1 Voice feature extraction

In this study, we use three frequency analysis techniques, which are CQCC, LFCC, and MFCC, for voice feature extraction from audio data. The frequency analysis techniques are commonly used in signal processing applications to extract meaningful information from signal data.

The CQCC captures high-detail frequency characteristics across both low and high-frequency bands using a logarithmic scale, demanding significant computational resources. LFCC treats all frequency components equally on a linear frequency scale, which is a key feature that effectively utilizes high-frequency band information for feature extraction. MFCC mimics human auditory perception by compressing high-frequency information using the Mel scale, providing a detailed representation of low-frequency bands.

3.2 Deepfake Audio Detection

In this study, we used an LSTM structure as the deepfake audio detection model. LSTM, a type of RNN structure, has the advantage of being able to learn both short-term and long-term sequence information. Traditional deepfake audio detection models mainly used the GMM (Gaussian Mixture Model) structure, which processes each frame independently and thus fails to learn the sequence information of the speech signals. Therefore, by using the LSTM model to learn speech signals, it is possible to learn the sequence characteristics of the speech signals, and we can expect better performance than GMM when dealing with complex data like speech signal characteristics [5].

4 Experiment

4.1 Experimental Environment

The experiments were conducted on Google Colab, and the specifications of Colab include an Nvidia Tesla T4 15GB GPU, an Intel(R) Xeon(R) CPU @ 2.00GHz, and 12.7GB of RAM.

4.2 Dataset

In this paper, we used the ASVspoof 2019 LA dataset. The training dataset consists of 2,580 genuine samples from 8 male speakers and 12 female speakers, and 22,800 spoof samples generated by deepfake audio. The evaluation dataset used for model assessment consists of 7,355 genuine samples from 21 male speakers and 27 female speakers, and 63,882 deepfake audio samples. The deepfake audio samples were created using two techniques: TTS (Text-To-Speech) and VC (Voice Conversion).

4.3 Performance Evaluation and Experimental Results

The performance evaluation of the deepfake audio detection model according to feature extraction can be seen in Table 1. When comparing the models' performance using the F1-score on the total dataset, MFCC shows high precision but has many false negatives, resulting in low recall. On the other hand, CQCC has slightly lower precision than MFCC but higher recall. Overall, the CQCC model demonstrates the best performance, with an F1-score of 0.89 and an accuracy of 0.88.

Table 1. Results of male, female and total voice comparison experiments according to voice feature extraction.

Feature Extraction method	Dataset	Precision	Recall	F1-score	Accuracy
CQCC	Total	0.91	0.88	0.89	0.88
	Male	0.91	0.89	0.90	0.89
	Female	0.92	0.88	0.89	0.88
LFCC	Total	0.91	0.86	0.87	0.86
	Male	0.90	0.86	0.87	0.86
	Female	0.91	0.86	0.88	0.86
MFCC	Total	0.93	0.82	0.85	0.82
	Male	0.92	0.83	0.86	0.83
	Female	0.93	0.82	0.85	0.82

When the data is separated by gender, Precision for female data is slightly higher, while Recall for male data is somewhat higher. However, overall, the performance difference between male and female data is not significant.

5 Conclusion

This paper demonstrates the effectiveness of deep voice detection techniques using CQCC, LFCC, MFCC, and LSTM. While MFCC, commonly used for analyzing human audible frequency bands, was expected to be effective for voice analysis, experimental results showed that CQCC and LFCC, which consider other frequency bands, performed better. The model using CQCC demonstrated the best performance. Therefore, the deepfake audio generation model appears to leave significant traces in frequency bands other than the audible frequency during the creation of deepfake audio. Further analysis will be conducted in future research.

References

1. Zahra Khanjani, Gabrielle Watson, and Vandana P Janeja. How deep are the fakes? focusing on audio deepfake: A survey. *arXiv preprint arXiv:2111.14203*, 2021.

2. Yishuang Ning, Sheng He, Zhiyong Wu, Chunxiao Xing, and Liang-Jie Zhang. A review of deep learning based speech synthesis. *Applied Sciences*, 9(19):4050, 2019.
3. Zaynab Almutairi and Hebah Elgibreen. A review of modern audio deepfake detection methods: challenges and future directions. *Algorithms*, 15(5):155, 2022.
4. Xinhui Zhou, Daniel Garcia-Romero, Ramani Duraiswami, Carol Espy-Wilson, and Shihab Shamma. Linear versus mel frequency cepstral coefficients for speaker recognition. In *2011 IEEE workshop on automatic speech recognition & understanding*, pages 559–564. IEEE, 2011.
5. Awais Khan, Khalid Mahmood Malik, James Ryan, and Mikul Saravanan. Battling voice spoofing: a review, comparative analysis, and generalizability evaluation of state-of-the-art voice spoofing counter measures. *Artificial Intelligence Review*, 56(Suppl 1):513–566, 2023.

The architectural difference of Consumer, IoT and M2M eSIM RSP security

Hoseok Kwon, Taeho Won, Bonam Kim, and Ilsun You

Kookmin University, Seoul, 02707, Republic of Korea
{hoseok1997, xoghdnjs12, kimbona, isyou}@kookmin.ac.kr

Abstract. With the benefit of enhancing physical security, ability to manage multiple profiles from a remote system, and enabling device miniaturization, eSIM(embedded Subscriber Identity Module) is being introduced to the market. In order to adapt eSIM to Korean environment for remote machine identification, we first analyzed the consumer, IoT, M2M RSP(Remote Sim Provisioning) architecture specified by the GSMA(Global System for Mobile Communications Association). And by comparing its differences, we have analyzed the key element and its security requirement to operate safe eSIM environment. Based on this work, we plan to conduct future research on security requirement that are required on Korean domestic environment.

Keywords: eSIM · RSP · security requirement.

1 Introduction

Traditionally SIM(Subscriber Identity Module) profile was managed with removable USIM(Universal SIM) card. However, eSIM(embedded SIM), which integrates eUICC(embedded Universal Integrated Circuit Card) is embedded directly onto the device's board[1]. Since the card used to store SIM information is embedded, GSMA(Global System for Mobile Communications Association) developed the RSP(Remote SIM Provisioning) protocol to remotely manage the SIM profile. And eUICC with SIM profile on it is called eSIM(embedded SIM). eSIM brings benefits such as enhanced physical security, the ability to manage multiple profiles from a remote system, and enables device miniaturization[5]. The RSP protocol is categorized into three types depending on the usage environment: Consumer, IoT(Internet of Things), and M2M(Machine-to-Machine). Given that the RSP protocol is crucial for the secure operation of eSIM ecosystems, it is essential to analyze the differences in each structure to ensure proper implementation across various industrial environments. In this paper we have analyzed the Consumer, IoT and M2M RSP protocols architecture. While analyzing the each architecture, we have compared its security requirement based on its architectural difference. And third, we discussed considerations that needs to be further discussed while adapting eSIM to Korean IIoT environment.

2 RSP Architecture

RSP protocol offers secure manner to provision the SIM profile to the secure element(eUICC) in remote devices. It authenticates the eUICC, encrypts the profile to protect while transmission and uses integrity verification method to make sure the data is not tampered on the process. The architecture of this technology is defined to three types: Consumer RSP, IoT RSP and M2M RSP. Architectural deference of each type is described in figure 1. Common element for every architecture are operator and SM-DP(Subscription Manager-Data Preparation). Operator is the mobile network operator which are able to create profile that can be used to authenticate device onto the network. SM-DP is responsible for preparing profiles and authenticate the device before provisioning them to eUICC. Once eUICC is authenticated profile is encrypted and sent to the the eUICC. Interface between Operator and SM-DP+ is defined as ES2+ interface and it is assumed to be safe with mutual authentication using the agreed upon method between two elements. RSP protocol specifies the ES2+ interface but the authentication between method is out of bound.

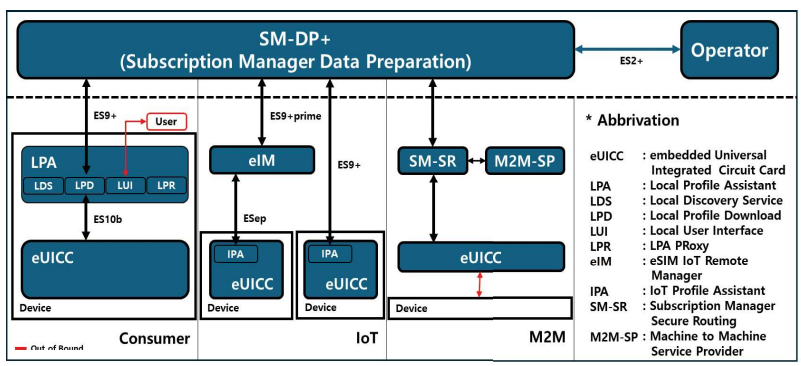


Fig. 1. Consumer, IoT, M2M RSP Architectures

2.1 Consumer RSP

Consumer RSP is designed to provision the SIM profile to the devices which consumers can interact with. Such as smartphone, tablet, wearable etc[2]. Consumer, or user, initiates the RSP protocol(Pull model) by interacting with LPA(Local Profile Assistant). RSP consists of two phases. First, mutual authentication between SM-DP+ and eUICC. Second, encrypting the profile and transmitting to the eUICC from the SM-DP+. Since eUICC is just a circuit chip with security credentials, it does not have a capability to connect directly to the SM-DP+. LPA connects user, eUICC and SM-DP+ in the middle. User performs profile management, such as initiating the RSP for profile download or enabling or disabling the profiles with LUI(Local User Interface) function in the LPA.

And LPD(Local Profile Download) function connects eUICC and SM-DP+ and proceeds RSP procedures.

In consumer RSP architecture, LPA is the biggest attack surface to the attacker. And it is important to implement the LPA function as specified to minimize the possibility of attack. GSMA guides 73 requirements for LPA. And among these, requirements related with security can be summarized to three parts. First LPA should protect profile data from unauthorised access through LUI. Second session between SM-DP+ and LPA is through open channel, so it must be protected using TLS. Third, profile management procedure must be proceeded as specified with user interaction involved.

2.2 IoT RSP

The development of IoT RSP was motivated by the need to use eSIM technology in IoT devices, which are normally NCD(Network Constrained Device) and/or UICD(User Interface Constrained Device)[3]. IoT RSP architecture is also a pull model and based on consumer RSP. However, since the user interaction is impossible in UICD and network abilities are constrained in NCD, IoT RSP architecture mitigates this constraints with eIM(eSIM IoT Remote Manage). eIM performs as LPA in consumer RSP. It is not only a primary element to connect with SM-DP, but also IoT device managers could manage fleet of IoT devices with eIM. eIM is outer element that are separated from the IoT devices. Because of eIM, IoT devices profiles could be safely managed. However, it also opens a new attack surface to the attacker. Compromised eIM[1] and malicious eIM[2]is the two main threats described in the specification. To mitigate such threats, specification guides to reference GSM SAS and IEC/ISO 27001:2022[4]

2.3 M2M RSP

M2M RSP was the first RSP architecture developed by the GSMA. Unlike consumer and IoT RSP, it is push model where operator pushes the profile to the remote eUICC. It is developed to use in a situation when the device is not easily reachable[3]. As its name, human interactions are not much required in the M2M architecture. In the eUICC manufacturing process, EUM(eUICC Manufacturer) must register the eUICC to the SM-SR(Subscription Manager Secure Routing). Once the eUICC is registered to the SM-SR, when the operator wants to initiate the RSP protocol, SM-DP+ can connect with SM-SR and SM-SR will reach the registered eUICC and relay the authentication and provisioning.

3 Conclusion

In this paper we have analyzed three RSP architecture. Based on architectural differences we have chosen the key element in each architecture and analyzed its

[1] An authorized and legitimate eIM is manipulated and compromised.

[2] A malicious eIM is both not authorized and illegitimate, e.g. malicious eIM acts as man-in-themiddle.

Table 1. Characteristics of three RSP architecture

	Consumer	IoT	M2M
Description	Pull model, which user requests the RSP to initiate via application such as LPA	Pull model, which IoT device manager manages profiles via eIM	Push model, which mobile service operator initiates the RSP to reach the devices that are hard to reach
Used devices	Smartphone, Wearable and Tablets etc	IoT devices (Sensors, Robots etc)	Utility meter, Car and Camera etc
Key element	LPA	eIM	SM-SR
Requirements	LPA needs to provide appropriate security measure to three components(User, SM-DP+, eUICC).	Setting the connection between eIM with IoT device and managing its connection in safe manner is crucial to IoT RSP architecture	Mutual authentication is required between SM-SR and SM-DP+(Operator)

security requirements. However, to adjust the eSIM technology to the domestic environment or private 5G environment, additional requirements could arise. Such as cryptographic algorithm that is only used in specific realm or governmental policy compliance. These considerations must be more discussed and its impact on security must be formally verified as a future research.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2024-00398312,Development of Quantum Security Based Device Identification and Test Verification Technologies for 5G Non-Public Network).

References

1. GSMA: esim whitepaper. Tech. rep., Global System for Mobile Communications Association (2018), https://www.gsma.com/solutions-and-impact/technologies/esim/wp-content/uploads/2018/12/esim-whitepaper.pdf
2. GSMA: Sgp.21 rsp architecture v3.1r. Tech. rep., Global System for Mobile Communications Association (2023), https://www.gsma.com/solutions-and-impact/technologies/esim/?attachment_id=16357/
3. GSMA: esim iot architecture and requirements version 1.2. Tech. rep., Global System for Mobile Communications Association (2024), https://www.gsma.com/solutions-and-impact/technologies/esim/?attachment_id=16357/
4. ISO: Iso/iec 27001:2022 information security, cybersecurity and privacy protection — information security management systems — requirements. Tech. rep., the International Organization for Standardization (October 2022), https://www.iso.org/standard/27001
5. Vesselkov, A., Hammainen, H., Ikalainen, P.: Value networks of embedded sim-based remote subscription management. In: 2015 Conference of Telecommunication, Media and Internet Techno-Economics (CTTE). pp. 1–7. IEEE (2015)

Analysis of Security Threats and Countermeasures of Artificial Intelligence-Based Digital Education

Hye-In Lee[1], Hyun-Che Song[2], and Il-Gu Lee[3]

Sungshin Women's University, Seoul, Korea
{lhynee, ssongzz, iglee}@sungshin.ac.kr

Abstract. The advancement of AI technology has raised the possibility of personalized education tailored to individual levels. In line with this trend, AI digital textbooks incorporating data-driven learning management systems are set to be introduced next year. However, there are significant security concerns given the nature of these services, which involve extensive data collection and utilization. This study analyzes the potential security threats inherent in these services by referencing the development guidelines of AI digital textbooks and proposes countermeasures to address these threats.

Keywords: AI Digital Textbook, Security Threats, Countermeasures.

1 Introduction

Providing 'personalized education' that considers the levels and characteristics of different learners to meet their individual needs has long been a challenge in the educational field. With the rapid development of digital technologies such as generative AI, new possibilities for 'personalized education' have been proposed, exemplified by AI digital textbooks. AI digital textbooks utilize intelligent information technology, including artificial intelligence, to support a variety of personalized learning opportunities tailored to individual students' abilities and levels. These textbooks are equipped with various learning materials and support functions. The Ministry of Education announced that AI digital textbooks will be introduced in subjects such as mathematics, English, and information starting in 2025, with a phased expansion until 2028. However, there are significant concerns about introducing digital textbooks at a national level for all students without clear guidelines or regulations related to AI usage and security. AI digital textbooks involve data-driven learning management systems that may include sensitive student information, presenting various security threats. Despite the high level of safety and accuracy required, discussions on these issues remain insufficient. Therefore, this paper aims to analyze the potential security threats of AI digital textbooks and propose corresponding countermeasures.

2 Characteristics of AI Digital Textbooks

AI digital textbooks support individual learning functions using intelligent information technology and are equipped with various learning materials. They enable

learning diagnosis and analysis through AI and provide personalized learning that reflects each student's learning level and pace. The services include learning diagnosis and analysis for students, optimal learning paths and content recommendations for each student, and data-based learning management functions for teachers, such as lesson design, customized prescriptions, and student learning history. Common services include analysis of student learning data via dashboards and support for communication among educational stakeholders. To achieve this, AI digital textbooks extensively collect and utilize data from users, particularly students.

Private companies handle the development of AI digital textbooks, while the integrated learning data hub and portal are managed publicly. Private companies provide AI digital textbooks for each subject and customized services to authenticated users via the AI digital textbook portal. The public sector manages and analyzes the data linked to the developers. The platform must be built on a cloud basis to allow direct access through web browsers without installing additional programs. Services must be developed in conjunction with the integrated authentication system provided by the AI digital textbook portal. Developers can receive user information necessary to provide customized services through authentication system integration [1].

3 Security Threats and Countermeasures of AI Digital Textbook

3.1 Policy Aspect

Developers of AI digital textbooks collect user data to provide personalized education services. According to the AI digital textbook development guidelines published by the Korea Education and Research Information Service, the data collected for services includes academic level, learning attitude, learning interest, learning status, and learning emotion. However, this data is likely to include sensitive information. The EU's AI Act classifies AI systems used to assess learning outcomes or appropriate education levels for individuals as high-risk. Stricter regulations apply particularly when using AI systems to infer human emotions in educational environments, such as learning emotions. However, the guidelines are vague or insufficient in specifying key regulations on data security, such as the types of data collected, the subjects and methods of data collection and management oversight, and the means of obtaining and refusing consent for data collection and use. For example, while it mentions collecting data on 'learning emotions,' it does not specify the methods or types of data collected. Additionally, under the Personal Information Protection Act, users can exercise the right to refuse, request explanations, or appeal against fully automated decisions that significantly affect them. However, this provision may not apply to digital textbooks where AI decision-making is an auxiliary means. The absence and ambiguity of regulations related to data collection and use can infringe on the privacy rights of information subjects and lead to excessive data collection and privacy violations [2].

Furthermore, since different private companies provide services for each subject and transmit the data collected to a public hub for management, developers must separate the infrastructure for their services from the AI digital textbook infrastructure. The data

collected through AI digital textbooks should not be used for the developers' service purposes. However, discussions on related regulations, management, and oversight are lacking. This raises the risk of developers misusing or abusing collected learning data to launch and promote their services, potentially violating student privacy [3].

Therefore, information on the data that can be collected and used by AI digital textbooks should be defined explicitly, and the data rights of information subjects, including how to refuse or counter data collection, should be communicated in understandable language. Additionally, specific and effective regulations are needed to manage, secure, and dispose of sensitive information contained in learning outcomes to protect privacy. Furthermore, rigorous management, oversight, and regulatory mechanisms should be established for developers' data collection and use.

3.2 Technical Aspect

AI digital textbooks' personalized learning support functions include learning diagnostics and recommendations, dashboards and data visualization, AI tutors for learning guidance and support, and AI assistant teachers for lesson design and prescriptions. The AI assistant supports teachers in lesson design, feedback writing, and grading. At the same time, the AI tutor can receive and record students' questions through chat at any time, providing content that matches their learning level and even understanding psychological aspects to convey to the teacher. While this technology can enhance the efficiency of educational learning, there are security threats due to the technical limitations of generative AI technologies, such as LLM (Large Language Models), that make this possible. A prominent issue is the provision of incorrect information. Causes include algorithmic hallucinations and bias. In fact, large language models used by generative AI have limited capabilities to analyze learned data and cannot determine the authenticity of the data. As a result, due to imbalances in learning data, insufficient feedback data learning, and direct or indirect biases within the data, there is a risk of providing students with answers that are presented as facts but are not or information that may encourage social discrimination and prejudice. Providing incorrect information in textbooks for learning can cause social and ethical confusion and lead to incorrect decision-making. Additionally, stored user data may be leaked due to database hacking or membership inference attacks by developers or hackers with access to this data. Therefore, it is necessary to carefully review whether language models can provide safe, accurate, and ethical responses and research safe AI models. AI literacy education should help users understand AI's principles, functions, and limitations and use AI critically. Furthermore, an independent and professional quality evaluation system is needed to assess the processes and procedures of personalized education service developers to provide safer and more effective education [4].

Meanwhile, AI digital textbooks should be developed as cloud-based web services to implement services through flexible AI function expansion, and the Cloud Computing Act stipulates that infrastructure and software with a security certification level of ‘medium’ or higher should be used. Although cloud services are very convenient in terms of accessibility, they also carry the risk of allowing access from insecure networks. If the service is used in an unsafe environment, unauthorized third parties may access and leak important data or attempt supply chain attacks against the service

provider. Additionally, despite high-level security measures, if a disaster affects the data center running the cloud service and storing data, all critical data collected from students and teachers could be lost, causing social chaos. Therefore, disaster preparedness measures must also be established. Moreover, there is the possibility of cyber threats such as web hacking or zero-day attacks exploiting vulnerabilities in the cloud service's APIs and software. In such cases, the server may be compromised, leading to the leakage of all data in the database. To prevent this, multi-factor authentication should be utilized, accounts should be locked after multiple failed login attempts, and users should be notified. Log viewing capabilities should also be provided to identify the causes of abnormal behavior and incidents. Additionally, security technologies and procedures such as data encryption and access control for learning data security should be applied, and security education should be provided to raise students' awareness and vigilance about security [5].

4 Conclusion

The introduction of AI digital textbooks is expected to provide personalized education tailored to the individual learning levels of students. However, there are considerable concerns about the side effects of its rapid implementation. Data security is crucial given the nature of AI digital textbooks, which collect and process extensive and sensitive data. Despite this, there are insufficient policies regarding the safety and privacy of the collected data, as well as regulations and oversight mechanisms concerning the data collection and use by developers. This poses risks of data misuse and infringement on privacy rights. Therefore, clear and specific regulations are necessary.

Moreover, there are inherent threats due to the limitations of generative AI and cloud technologies, such as hallucinations, confusion caused by data bias, and data leaks or losses due to hacking or disasters. Thus, research on safe and reliable language models and the application of technologies and procedures to secure learning data are required. Additionally, user education on security and AI literacy is needed.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Secu-rity Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) pro-gram (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Ministry of Education, Korea Education and Research Information Service. (2023). AI Digital Textbook Development Guidelines.
2. The 3rd Summit for Democracy: Privacy and Surveillance Issues in AI Digital Textbook Policy by Kyoungmi Oh.

3. Jeong Soon-won. (2023). Legal Issues of AI Digital Textbooks. Journal of Edu-cation Law Studies, 35(3), 171-197.
4. National Intelligence Service, National Security Research Institute. Security Guidelines for Generative AI such as ChatGPT.
5. Kim Hyun-woo. (2023). Research on Enhancing the Security of Cloud SaaS Services: Focusing on Service Providers and Users.

Hardware-based NTT accelerator design for ML-DSA

Byeonguk Jeong[0009−0008−7365−6218], Eunse Kang[0009−0007−7398−1769], and Howon Kim[0000−0001−8475−7294]

Pusan National University
jbu0828@gmail.com, saintmansiu@gmail.com, howonkim@pusan.ac.kr

Abstract. As Advances in quantum computing have made it possible to crack traditional ciphers in polynomial time, raising concerns about the integrity of data. In response to this, the National Institute of Standards and Technology (NIST) has held a competition to standardise on quantum-resistant cryptography, and in digital signatures, it is standardising on the Module Lattice-Based Digital Signature Standard (ML-DSA). ML-DSA provides security through polynomial multiplication operations based on lattice space, but it has the disadvantage of high overhead compared to existing digital signatures. In this paper, we propose a high-speed NTT structure by optimising the hardware-based polynomial matrix-vector multiplication using the Dadda Tree Algorithm to speed up ML-DSA.

Keywords: ML-DSA · NTT · Dadda Tree Algorithm · Polynomial Multiplication

1 Introduction

Advances in quantum computing have made it possible to apply the Shor and Grover algorithms to break traditional digital signatures in polynomial time, which has led to increasing concerns about data integrity in communications and network environments. In response, the National Institute of Standards and Technology (NIST) held a competition to standardise quantum-resistant ciphers that are compatible with existing systems and quantum-resistant among various bases. In July 2023, one algorithm in the Key Encapsulation Mechanism (KEM) category and three algorithms in the digital signature category were selected and are being standardised. In the digital signature standardisation, the standardisation of lattice-based algorithms is underway based on Crystals-Dilithium among the three algorithms selected under the name ML-DSA (Module Lattice-Based Signature Standard). Lattice-based algorithms are resistant to attacks using Shore and Grover algorithms, but they have the disadvantage of high computational overhead compared to conventional RSA and ECC-based algorithms. In this paper, we propose a structure to optimise polynomial matrix-vector multiplication through hardware-based parallelism, which can speed up the Number Theoretic Transform (NTT) operation, which has the highest computational overhead.

2 Background

2.1 Crystals-Dilithium

Crystals-Dilithium is a lattice-based digital signature algorithm based on the Module Learning with Errors (MLWE) problem, which embeds small error values in the lattice space to make it difficult to find the original value, providing high security on classical and quantum computers. Crystals-Dilithium is secure against techniques such as Gaussian cancellation, and the computation is performed via polynomial operations on a space called the Polynomial Ring in $\mathrm{Z}_q[X]/(x^n+1)$.

2.2 NTT

The Number Theoretic Transform (NTT) is an algorithm that applies the Fast Fourier Transform (FFT) algorithm, which is widely used in signal processing, to the integer theoretic domain. NTT uses the concept of root of unity in a finite body to transform a polynomial into the NTT domain, and then performs multiplication of polynomials. Multiplication of conventional polynomials has a computational complexity of $O(n^2)$ because multiplication operations must be performed on all coefficients of the polynomial. However, by converting to NTT domain, the overhead of multiplication operation can be efficiently reduced because the total computational complexity of multiplication operation is $O(nlogn)$ due to the multiplication operation with the coefficients corresponding to each degree and the multiplication operation generated by converting to NTT domain.

2.3 Dadda Tree Reduction

The Dadda Tree Algorithm is one of the algorithms to increase the efficiency of multiplication operations, and it can perform multiplication operations quickly by compressing the partial products of the whole multiplication in stages to perform addition. The compression of the partial product generated at each bit position is repeated until a certain number of bits remain at each level through 3-2, 4-2 compressors, etc. In addition, the number of addition operations performed in the overall multiplication process and the carry transfer can be handled effectively to minimise the depth of the adder to speed up the computation.

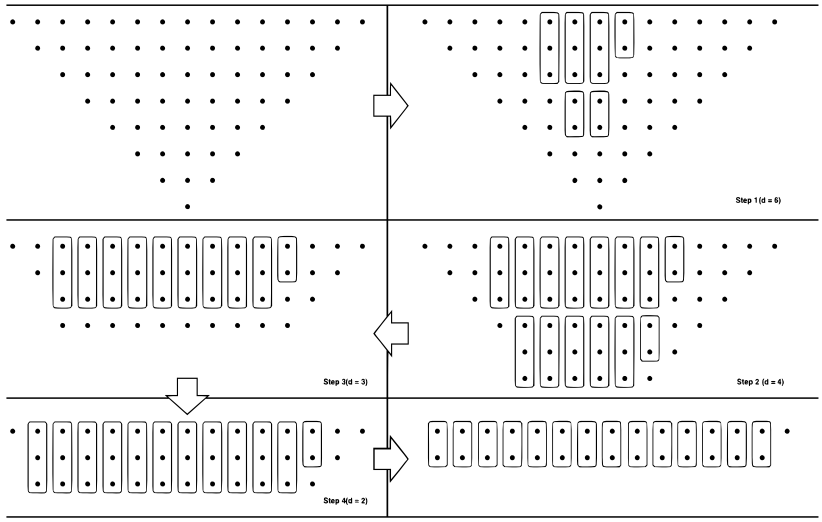


Fig. 1. Dadda 8x8 multiplier

3 Proposed Architecture

The NTT/INTT operation structure proposed in this paper is shown in Figure 2.

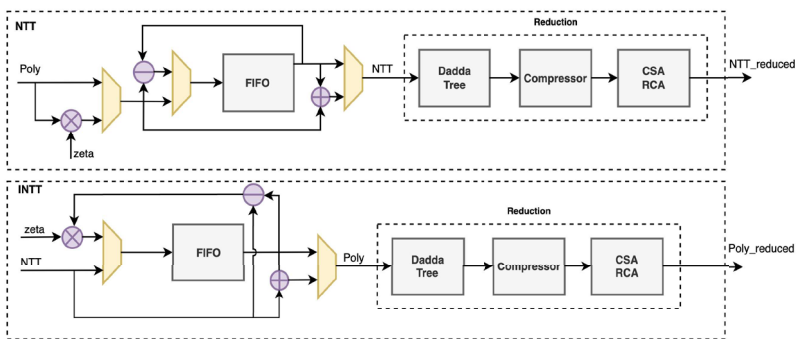


Fig. 2. FIFO-based NTT and Reduction operation schematics

According to the model structure of ML-DSA published by NIST, Montgomery Reduction is used to perform multiplication operations of polynomials

for reduction. When implemented in hardware, it is implemented through multipliers and adders, which generates overhead in the process of this operation and requires a lot of time. In this paper, the overhead is reduced by applying the Dadda Tree Algorithm. Dadda tree algorithm performs multiplication by decomposing the addition of each bit digit. The carry of each bit is generated between each operation, and additional overhead is incurred in the process of handling the carry. The carry is handled using the Carry Save Adder (CSA) and the Ripple Carry Adder (RCA). CSA can reduce the computation time by parallelising multiple bit additions, and RCA can reduce the overhead between operations of the dadda tree Algorithm because it requires fewer gates. The NTT and INTT conversions are organised in stages. Pipeline structure is applied by implementing each stage, and FIFO (First In First Out) is applied to efficiently control the input and output of data. The module receives each coefficient corresponding to the input and output sequentially, stores half of the number of inputs per stage in the FIFO, and outputs from the FIFO when the other half is input to perform the operation. The addition output is input to the next stage, and the subtraction output is stored in the FIFO again and exported after the addition output is completed. This reduces the computation time between NTT/INTT conversions.

4 Conclusion

In this paper, we proposed a hardware-based accelerator structure for NTT of quantum-resistant cryptography in order to make it quantum-resistant as quantum computers develop, and it is expected to contribute to the design of hardware-based quantum-resistant cryptography in fields such as real-time communication and IOT devices in the future by reducing the time required for computation when using this structure.

Acknowledgements This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

References

1. FIPS 204, Module-Lattice-Based Digital Signature Standard, NIST Post-Quantum Cryptography Standardization (2023)
2. Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: Algorithm specifications and supporting documentation (version 3.1). NIST Post-Quantum Cryptography Standardization Round 3 (2021)
3. Rentería-Mejía, C. P., and J. Velasco-Medina. "Hardware design of an NTT-based polynomial multiplier." 2014 IX Southern Conference on Programmable Logic (SPL). IEEE, 2014.

4. Dadda L. (1965), 'Some Schemes for Parallel Multipliers', Alta Frequenza, Vol. 34, No. 5, pp. 349-356.
5. Veeramachaneni, Sreehari, et al. "Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors." 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07). IEEE, 2007.
6. Kim, Taewhan, William Jao, and Steve Tjiang. "Circuit optimization using carry-save-adder cells." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 17.10 (1998): 974-984.
7. Sadeghi, Mohsen, Mahya Zahedi, and Maaruf Ali. "The Cascade Carry Array Multiplier–A Novel Structure of Digital Unsigned Multipliers for Low-Power Consumption and Ultra-Fast Applications." Annals of Emerging Technologies in Computing (AETiC), Print ISSN (2019): 2516-0281.

Efficient Fuzzing Technique for Lightweight Firmware-Over-The-Air

Na-Hyun Kim[1] [0009-0004-5514-237X], Jin-Min Lee[1] [0000-0002-4337-1771] and Il-Gu Lee[1] [0000-0002-5777-4029]

[1] Sungshin Women's University, Seongbuk-gu, Seoul 02844, Korea
20221079@sungshin.ac.kr, csewa56579@gmail.com, iglee@sungshin.ac.kr

Abstract. As Internet of Things (IoT) devices are widely used in the industry and daily life, over-the-air technology for IoT device maintenance and firmware update is emerging. However, the conventional firmware analysis technology is inefficient because the entire firmware file must be fuzzing including the previously verified part for vulnerability detection. Therefore, this study proposes a hierarchical partial fuzzing technique, which aims to fuzzing only the updated part except the previously verified part by making the firmware update file into a hierarchical structure. According to the experimental results, the proposed fuzzing technique found at least 4,565, up to 36,116 more crashes than conventional fuzzing technique, and reduced the time to find 10,000 crashes by 3 hours, 10 minutes, and 48.9 seconds. In addition, we found at least 1.5 more unique crashes and up to 1.75 more, and reduced memory usage by about 111KiB.

Keywords: Internet of Things, Lightweight Devices, Hierarchical Firmware, Firmware-over-the-air Update, Partial Fuzzing

1 Introduction

IoT devices used in various industries have limited memory resources and computing power, making it difficult to analyze firmware vulnerabilities. To solve this problem, research is being conducted on how to efficiently detect vulnerabilities in lightweight IoT devices. Considering the characteristics of lightweight devices, automated detection technology is being used to reduce vulnerabilities by cost-effectively reducing labor costs, saving time, and preventing losses [1]. This technology contributes to reducing security vulnerabilities by mitigating the limited parts of some IoT devices, but it still does not completely solve all the limitations. The conventional fuzzing technique is inefficient because the process of designing and verifying the firmware is not separated and thus the part that was previously analyzed must be re-verified. Therefore, in this paper, we proposed a hierarchical wireless partial firmware update for the lightweight IoT. The proposed technique can efficiently detect vulnerabilities even in IoT environments with resource constraints by detecting and fuzzing only the updated code.

2 Related Work

Research related to the detection of firmware vulnerabilities in IoT environment is underway. Novak et al.[2] introduce a callable native code fragment using a Lua scripting interface. This reduces data transmission and increases energy efficiency when updating the firmware of IoT device. However, since the Lua API is fixed, it is necessary to update the entire firmware when adding a new native code function. Sachidananda et al.[3] proposed a static analysis framework to detect code-related vulnerabilities and security vulnerabilities such as buffer overflows and injection for IoT. However, this technique is an automated adaptive static analysis technology, which is not suitable for lightweight IoT devices because of the increased time delay in the process of collecting crawling files. Hans Chandra et al.[4] proposed a wireless firmware update using a lightweight mesh-based P2PMesh (peer-to-peer) architecture in consideration of low-power IoT. However, there is a problem of increasing the power consumption of IoT due to the increase in the amount of communication with Hub Node.

3 Hierarchical Partial Fuzzing

Fig. 1 is a comparison of the conventional fuzzing technique, partial fuzzing technique, and hierarchical partial fuzzing technique. The conventional fuzzing technique is inefficient because the entire firmware file is fuzzing including the previously verified part. If only partial code is applied in conventional fuzzing technique, the complex interaction of the entire code may not be sufficiently reflected. Therefore, this study proposes a hierarchical partial fuzzing technique that applies fuzzing to multiple layers according to the structural layer of the code. The hierarchical partial fuzzing technique can test only the detailed part even within the firmware file and can increase the vulnerability detection rate or efficiency.

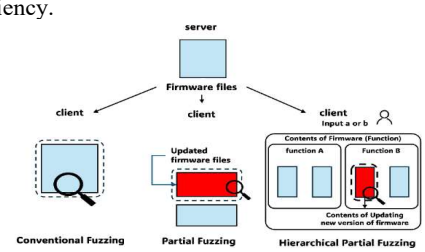


Fig. 1. Comparison among conventional fuzzing, partial fuzzing, and hierarchical partial fuzzing

By applying the server function, the contents of the conventional firmware file and the contents of the new firmware file are divided. When the server function A is applied, only the new firmware content can be fuzzing. The pseudo code of the hierarchical partial fuzzing technique to be proposed is shown in Fig. 2. Each of the firmware files is hierarchized so that partial fuzzing can be performed using the input of the server function. As this time, since the files are each hierarchized, when updating, only the newly updated part of the firmware update content can be selected and updated, excluding the conventional firmware update content. This implements a test logic that can fuzzing only the newly updated part using input.

Algorithm 1 Server Function A for Old Firmware Content

```
function SERVER_FUNCTION_A(old_firmware_content)
    Perform fuzzing on old_firmware_content
    print "Fuzzing completed for old firmware content"
end function
```

Algorithm 2 Server Function B for New Firmware Content

```
function SERVER_FUNCTION_B(new_firmware_content)
    Perform fuzzing on new_firmware_content
    print "Fuzzing completed for new firmware content"
end function
```

Algorithm 3 Hierarchical Fuzzing Function

```
function HIERARCHICAL_FUZZING(level_content, level)
    for all sub_level_content in level_content do
        if is_old_firmware(sub_level_content) then
            SERVER_FUNCTION_A(sub_level_content)
        else
            SERVER_FUNCTION_B(sub_level_content)
        end if
        if has_sub_levels(sub_level_content) then
            HIERARCHICAL_FUZZING(sub_level_content.get_sub_levels(),
level + 1)
        end if
    end for
end function
```

Fig. 2. Pseudo code of hierarchical partial fuzzing technique.

4 Performance Evaluation and Analysis

To evaluate the performance of the proposed fuzzing technique and the conventional fuzzing technique, we used AFL (American Fuzzy Loop), an automation tool that randomly inputs data into the program to find bugs and vulnerabilities. Each experiment ware repeated 20 times to obtain the average value. We compared the vulnerability detection rate by adding buffer overflow vulnerabilities to the new firmware file. The conventional method that does not partially fuzzing the entire code of 3.7109 KiB, but the proposed hierarchical partial fuzzing technique fuzzing the code of 1.6602 KiB into AFL.

4.1 Effect of Partial Fuzzing

Table 1 is a comparison table of the conventional fuzzing technique with the partial fuzzing technique. Table 1(a) shows the total number of crashes found in 10 minutes. The experimental results indicate that the proposed method found 1,044 more crashes (2 unique) on average compared to the conventional method and used 232 KiB less memory than the conventional method. Table 1(b) compares the time it took to find 11 unique crashes in the conventional and proposed methods. Compare to the conventional method, it was confirmed that the partial fuzzing shortened the time taken to fuzzing by about 3 minutes and 1 second.

Category	Number of Unique Crashes	Memory Usage
Conventional model	1,257 (8 unique)	4,412 KiB
Partial fuzzing	2,301 (10 unique)	4,180 KiB

(a)

Category	Number of times
Conventional model	8min 35sec
Partial fuzzing	5min 34sec

(b)

Table 1. Evaluation results between the conventional model and partial fuzzing model

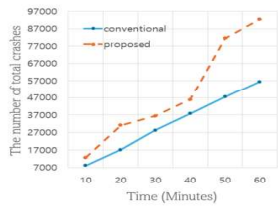
Even though the partial fuzzing technique is applied, good performance is not stratified, making it difficult to fuzz only the updated part. In addition, if only partial fuzzing is applied, the previously verified part cannot be fuzzed when an error occurs, but by making it hierarchically, the desired part can be fuzzed according to the server function.

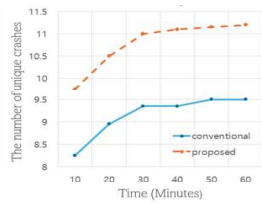
4.2 Effect of Hierarchical Partial Fuzzing

Fig. 3 compares the average number of crashes found for 10minutes in 1 hour in the proposed hierarchical partial fuzzing technique with the conventional fuzzing technique. Fig. 3(a) compares the average total number of crashes and shows that the

proposed fuzzing technique found at least 4,565 and up to 36,116 more than the conventional fuzzing technique. Fig. 3(b) compares the unique crashes, and according to the experimental results, the proposed model found at least 1.5 more and up to 1.75 more unique crashes than the conventional model.

(a)





(b)

Fig. 3. The number of total crashes and unique crashes to time

Table 2 is a comparison of the conventional fuzzing technique with the partial fuzzing technique. Table 2(a) is comparing the memory usage used by the conventional fuzzing technique and the proposed technique for 1 hour. The proposed fuzzing technique used less memory usage of about 111KiB than the conventional model. Table 2(b) is comparing the time it took to find a total of 10,000 crashes with the conventional and proposed fuzzing techniques. The proposed fuzzing technique shortened the fuzzing time by 45minutes and 48.67 seconds compared to the conventional fuzzing technique.

Category	Conventional	Proposed
Memory Usage (KiB)	4605.19167	4494.06167

(a)

Category	Conventional	Proposed
Number of Times	3hours 21min 55sec	11min 6.1sec

(b)

Table 2. Evaluation result between the conventional model and the proposed model

Overall, the proposed fuzzing technique was able to reduce memory usage while detecting many vulnerabilities in a short period of time compared to the conventional technique. It was able to reduce the lightweight the firmware because it was not analyzed again the part that was previously verified.

5 Conclusion

Lightweight firmware updates are essential to perform vulnerability analysis inside IoT devices with limited resources. This study proposes a hierarchical partial fuzzing technique for lightweight wireless firmware updates that can make firmware update files into a hierarchical structure and fuzzing only the update parts except the previously verified parts. The proposed technique found a total of 4,565 more crashes and a maximum of 36,116 compared to the conventional technique, and reduced the time taken to find 10,000 crashes by 3 hours, 10 minutes, and 48.9 seconds. In addition, we found a minimum of 1.5 more unique crashes and a maximum of 1.75 more, and reduced memory usage by about 111KiB. In future research, we play to apply the proposed model to Raspberry Pi, a real IoT device, to conduct a performance comparison experiment with the conventional method.

Acknowledgements. This work was supported by Korea Foundation for Women In Science, Engineering and Technology (WISET) grant funded by the Ministry of Science and ICT(MSIT) under the team research program for female engineering students (No. 2024-138), the MOTIE under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Eceize, M., Flores L., Iturbe M.,: Fuzzing the Internet of Things: A Review on the Techniques and Challenges for Efficient Vulnerability Discovery in Embedded Systems. IEEE Access **8**(13), 10390-10411 (2021)
2. Novak, M., Skryja, P.: Efficient Partial Firmware Update for IoT Devices with Lua Scripting Interface. In: RADIOELEKTRONIKA (29th International Conference Radioelektronika), Pardubice, Czech Republic (2019)
3. Sachidananda V., Bhairav S., Yuval Elovici.:OVER: Overhauling Vulnerability Detection for IoT through an Adaptable and Automated Static Analysis Framework. In: SAC'20 (The 35th ACM/SIGAPP Symposium on Applied Computing), Brno, Czech Republic (2020)
4. Chandra, H., Anggadjaja, E., Setya Wijaya, P., Gunawan, E.: Internet of Things: Over-the-Air (OTA) firmware update in Lightweight mech network protocol for smart urban development. In: In: APCC (22nd Asia-Pacific Conference on Communications), Yogyakarta, Indonesia (2016)

Enhancing Operational Technology Cybersecurity using Security Levels, Differential Privacy, and Post-Quantum Cryptography

Junghyun Moon[1][0009-0005-9148-2168], Eunji Chang[1][0000-0002-8364-9213] and Kookheui Kwon[1][0000-0001-6104-6000]

[1] Korea Institute of Nuclear Nonproliferation and Control, 1418 Yuseong-daero, Korea
moon_aver@naver.com

Abstract. International and domestic bodies have established standards and regulatory guidelines aimed at enhancing Operational Technology (OT) cybersecurity and implementing measures to ensure a safe operational environment. However, OT systems prioritize availability and integrity for the safe OT environment. Therefore, there is a need for comprehensive security enhancement strategies to efficiently protect sensitive information. This paper proposes a security solution based on Security Levels (SL) utilizing Differential Privacy (DP) and Post-Quantum Cryptography (PQC). Through experiments, it is demonstrated that differential application of DP based on SL can effectively protect information within the OT industry.

Keywords: Security Level (SL), Operational Technology (OT), Differential Privacy (DP), Post Quantum Cryptography (PQC)

1 Introduction

The integration of ICT technologies such as IoT and AI has converged OT and IT, enhancing production efficiency but also exposing these systems to security threats like hacking and malware. Advancements in communication technologies have enabled new initiatives to safely utilize wireless connections in previously restricted environments like nuclear facilities. For instance, the Nuclear Energy Institute (NEI) in the US seeks the Nuclear Regulatory Commission (NRC) endorsement of Revision 7 of its cybersecurity plan (NEI 08-09) for nuclear facilities. Thus, enhanced security measures are crucial to address new technological threats, including Year-2-Quantum (Y2Q).

This paper proposes differential security measures based on Security Levels (SL) for Critical Digital Assets (CDAs) performing Safety, Security, and Emergency Preparedness (SSEP) functions to enhance OT security, including nuclear facilities. The contributions are:

- Proposal and Evaluation of a Protection Method for CDAs Using Differential Privacy (DP): This method secures essential digital assets through DP technology.

- Ensuring Secure Wireless Connectivity in Nuclear Facilities through Post-Quantum Cryptography (PQC): This approach secures wireless connectivity in nuclear facilities, even in a quantum computing environment.

2 Domestic and international cybersecurity requirements

In industrial environments, various levels of security strength may be required to achieve the CIA triad (Confidentiality, Integrity, and Availability) based on the sensitivity of the information and regulatory requirements. International organizations such as the International Electrotechnical Commission (IEC) and the National Institute of Standards and Technology (NIST) provide standards, as shown in Table 1, using SL to establish a framework for addressing cyberattack issues according to different security levels[1].

Due to the critical importance of safety and security, nuclear facilities maintain a regulatory framework distinct from other industrial facilities, adhering to domestic and international cybersecurity laws and regulatory guidelines specific to nuclear facilities. Regulatory guides such as R.G.5.71 and KINAC/RS-015, illustrated in Figure 1, outline the cybersecurity defensive architecture to counter electronic intrusions[2, 3]. These frameworks classify SL into multiple level such as five levels, placing CDAs that perform safety and security functions at the highest level, like Level 4, for protection.

However, while CDAs prioritize SSEP functions, the OT industry prioritizes availability, placing relatively lower importance on confidentiality-related security measures, including data privacy. This demonstrates requires additional measures or considerations to prevent data leakage of sensitive information within these systems.

Table 1. IEC 62443-3-3 defined security levels

Security level	Define
SL 0	No specific requirements or security protection necessary
SL 1	Protection against casual or coincidental violation
SL 2	Protection against intentional violation using simple means with low resources, generic skills and low motivation
SL 3	Protection against intentional violation using sophisticated means with moderate resources, IACS specific skills and moderate motivation
SL 4	Protection against intentional violation using sophisticated means with extended resources, IACS specific skills and high motivation



Fig. 1. Cybersecurity defensive architecture

3 Proposal methods

In this paper, security measures outlined in Figure 2 are utilized to differentially apply DP to CDAs which contain highly sensitive information. For CDAs using wireless connections, PQC is employed to ensure data privacy. DP protects data privacy by applying differential privacy techniques to the algorithms that use the data, typically implemented using the Laplacian Mechanism, a randomized function K gives ε-differential privacy if for all data sets D1 and D2 differing on at most one element, and all S ⊆ Range(K)[4].

$$Pr[K(D1) \in S] \leq exp(\varepsilon) \times Pr[K(D2) \in S] \quad (1)$$

By adding noise to the data, DP maintains privacy while remaining applicable to various data types and analysis techniques. While this paper uses the SL of nuclear facilities as a case study, the proposed method is also applicable to other OT environments, including those that adhere to the IEC 62443 model.

Quantum computers pose substantial security threats. It is projected that within 10-15 years, RSA-2048 encryption could be decrypted within 24 hours by quantum computers, significantly endangering current cryptographic systems[5]. To address this, NIST selected PQC standards in 2022, including CRYSTALS-KYBER for public key cryptography and CRYSTALS-Dilithium, FALCON, and SPHINCS+ for digital signatures [6]. PQC designs public key cryptography using mathematical problems that are difficult for quantum computers to solve.

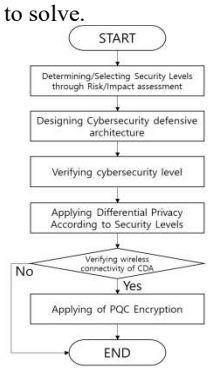


Fig. 2. Proposed method flowchart

4 Experiment environment and performance analysis

This paper assumes a data transmission environment for the application of DP. To evaluate the performance of the proposed model, SL are divided into five stages (level 0~4), and a Convolutional Neural Network (CNN) model is trained to assess accuracy and loss. The dataset used is the iris dataset, consisting of 150 records. 30% of the data is used as test data, and the remaining is used as the training dataset to measure

performance. The accuracy and loss results of the proposed method are shown in Fig. 3. By differentially applying DP depending on the security level, it is observed that as the SL increases, the strength of data protection also increases. However, this leads to a decrease in performance in terms of accuracy and loss. For level 0, where no noise is applied, the accuracy converges to nearly 100%.

Additionally, to assess the secure data transmission performance of PQC, a protocol involving mutual authentication between the server and client through a handshake process was employed. Open Secure Socket Layer (SSL) and Transport Layer Security (TLS) can perform public key/private key operations, certificate generation, and message encryption/decryption. Table 2 shows the results of measuring the key size and the signature generation and verification times for the existing RSA algorithm and the PQC algorithms, confirming the feasibility of applying the algorithm change[7].

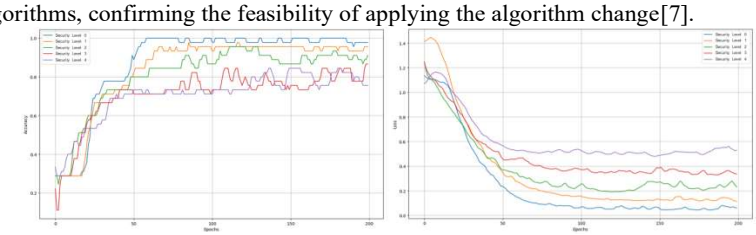


Fig. 3. Accuracy and loss per epoch

Table 2. Comparison of RSA, Dilithium and Falcon in Openssl key/signature generation

Algorithm	Signature time (ms)	Verification time (ms)	CertificateVerify Size (KB)
RSA3072	0.023	0.001	0.38
Dilithium-2	0.021	0.005	0.69
Falcon-512	0.054	0.004	3.37

5 Conclusion

This paper proposes a method to enhance OT security using SL and demonstrates through experiments that information can be efficiently protected by differentially applying DP based on SL. Given the importance of availability and safety in OT, which demands detailed performance requirements to ensure determinism, we conducted experiments to validate the applicability of the proposed solution. As shown in this paper, the study on security controls for enhancing cybersecurity between different SL can be considered when reviewing of the latest OT security technology. Future research will explore the applicability of OT communication protocols considering the functionality and parameter sizes of Post-Quantum Cryptography (PQC) based on standards such as IEEE 802.11 and IEC 62591. Furthermore, for the proposed method to be implemented, issues related to the high cost of QKD transceiver units and distance limitations due to deformation and loss during transmission must be addressed.

References

1. IEC.: System security requirements and security levels, IEC 62443-3-3 (2013)
2. U.S.NRC, Cyber Security Programs for Nuclear Facilities, Regulatory Guide(R.G) 5.71 (2010)
3. KINAC.: Security of computers and information system in nuclear facilities, Regulatory Standard(RS)-015 (2023)
4. Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds) Automata, Languages and Programming. ICALP 2006. Lecture Notes in Computer Science, vol 4052. Springer, Berlin, Heidelberg. (2006)
5. Dae-seong Kwon.: Quantum-resistant cryptography and quantum cryptography for countering quantum computer threats, TTA ICT Standard Weekly 1137 (2023)
6. NIST.: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, last accessed 2024/06/25
7. Sikeridis, Dimitrios, Panos Kampanakis, and Michael Devetsikiotis.: Post-quantum authentication in TLS 1.3: a performance study, Cryptology ePrint Archive (2020).

Evaluation and Analysis of Adversarial Attacks to Digital Watermarks

A-Young Jeon[1] [0009-0002-0162-5865], Yeon-Ji Lee[1][0000-0002-0482-2381] and Il-Gu Lee[1][0000-0002-5777-4029]

[1] Sungshin Women's University, Seoul, 02844, Korea
{qiena11jeon, cselab.lyj, iglee19}@gmail.com

Abstract. Recent research has focused on developing invisible digital watermarks to protect digital content copyrights. Traditionally, the Peak Signal-to-Noise Ratio (PSNR) has been used as an evaluation metric to assess the robustness of digital watermarks. However, this metric has limitations in evaluating resistance to distortion attacks. This study experimentally demonstrates that PSNR is inadequate for assessing the impact of adversarial attacks on digital watermarks. Furthermore, it analyzes the requirements for metrics that can effectively evaluate the robustness of digital watermarks.

Keywords: digital watermark, robustness, adversarial attacks.

1 Introduction

Recently, digital watermarking has been re-evaluated as a technology to address the issues of fake online content created using generative AI and deepfake technologies. Digital watermarking is a key technology for preventing copyright infringement and is widely used in fields such as copyright protection, integrity verification, and prevention of illegal duplication. In particular, it supports the verification of authenticity and management of ownership by embedding copyright holder information into various digital content, including images, videos, and audio.

There is active research on invisible watermarking technology, which embeds watermarks while maintaining the Quality of Service (QoS) of digital content. According to previous studies, traditional digital watermarks are vulnerable to transformation attacks such as cropping and lossy compression, as well as attacks that insert specific noise into images, known as adversarial attacks. Adversarial attacks involve adding subtle noise to data to induce misbehavior in AI models, posing a significant threat across various AI application domains. If digital watermarks are susceptible to such attacks, it could lead to serious issues regarding copyright protection, integrity verification, and legal validity.

This paper defines all image processing that may lead to watermark misdetection and communication damage as attacks. Conventional watermarking methods have used metrics like PSNR (Peak Signal-to-Noise Ratio) to evaluate the robustness of watermarks against these attacks. However, PSNR, as a measure of image quality, is unsuitable for assessing robustness. Therefore, this paper demonstrates the need for an evaluation metric that assesses the robustness of watermarks against adversarial attacks and image quality degradation.

The main contributions of this study are as follows:

- Proves the limitations of PSNR as an evaluation metric for conventional watermark robustness.
- Conducts watermark attack experiments on domestic invisible watermarks and evaluates their robustness against practical attacks.
- Analyzes the requirements for evaluation metrics for the robustness of digital watermarks.

2 Related work

Recently, research has been actively conducted on digital watermarking technology that guarantees invisibility and is robust against tampering attacks.

Matthew Tancik et al.[7] proposed StegaStamp, an invisible watermark using deep learning-based steganographic technology. StegaStamp works by embedding an invisible hyperlink bitstring within an image and extracting the information via an Internet-connected imaging system. Additionally, to ensure the robustness of the watermark, it learns by taking into account various distortions that may occur during photo-taking (perspective distortion, photo blur, defocus, color manipulation, noise, and lossy compression). Although this method can detect watermarks even when the image quality is degraded, it does not consider practical attacks targeting watermarks.

Pierre Fernandez et al.[8] proposed a Stable Signature, an invisible watermark, to solve ethical problems in distributing content generated by generative AI. A stable Signature guarantees invisibility by embedding a binary signature during the process of AI-generating images. However, the robustness of the image was evaluated based on PSNR, and robustness through realistic attacks was not considered.

3 Adversarial Watermark Attack Techniques

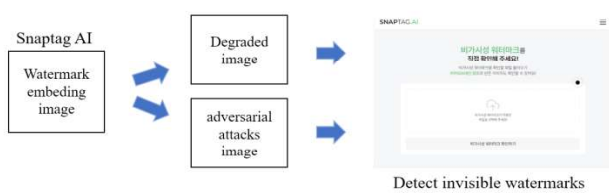


Fig. 1**.** Watermark attack scenario**.**

In this study, to evaluate the robustness of the watermark, Fig. 1. We performed an adversarial attack and image quality deterioration attack on the invisible watermark, as shown.

The watermark used in this study is an invisible watermark developed by SnapTag[9], and watermarks can be created and extracted on the SnapTag AI website. This watermark can be used to check whether it is applied, its source, and detailed information through the unique code, which is the unique number of the invisible watermark. This paper evaluated the practical robustness of the watermark by conducting two attacks against the SnapTag watermark: an adversarial watermark attack and image quality degradation.

Using an adversarial attack technique based on FGSM (Fast Gradient Sign Method), the noise intensity (epsilons, ε) value was increased by 0.01 increments until the watermark was no longer detected. For the image quality degradation attack, we gradually reduced the image quality from 100% to 10% of the original image's capacity in 10% increments. We measured the PSNR value resulting from the image degradation and evaluated whether the invisible watermark functioned normally to assess its robustness.

4 Experimental Results and Analysis

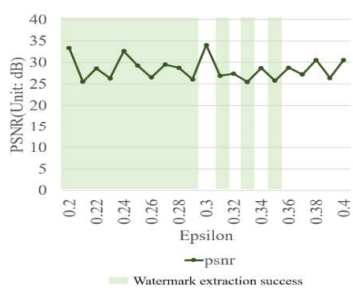


Fig. 2. PSNR and Detection Status According to Changes in Adversarial Attack Noise.

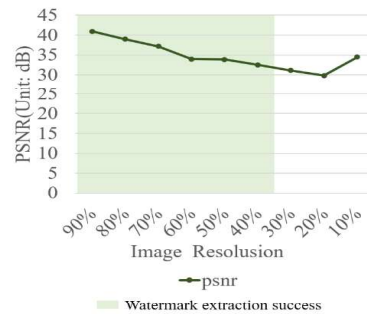


Fig.3. PSNR and Watermark Detection Status According to Changes in Image Quality.

Fig. 2. shows the watermark detection status and PSNR results as the ε value, representing noise information, increases by 0.01 increments. In the figure, the green boxes indicate successful watermark detection. According to the experimental results, the watermark was not properly detected when the ε value was 0.3 or higher. However, even though the watermark was not detected at ε values of 0.29, 0.31, 0.33, and 0.35, the PSNR values remained high. This demonstrates the limitations of PSNR as a metric for evaluating robustness against actual attacks.

Fig. 3 illustrates the PSNR values and watermark detection status according to the intensity of image quality degradation attacks. The x-axis represents the percentage reduction in image quality from the original image. As the image quality degrades, the PSNR value tends to gradually decrease. However, even when the PSNR value drops and the watermark is undetected, it remains above 30 dB. The watermark cannot be extracted when the image quality is reduced by 30% or more.

The experimental results indicate that adversarial attacks can obstruct watermark extraction without significantly impacting image quality, which cannot be effectively detected by the PSNR metric. In the case of image quality degradation, the minimum PSNR value at which the watermark cannot be extracted is maintained above the 30 dB threshold mentioned in previous studies. However, when the image quality degrades by 30% or more, watermark detection becomes impossible, indicating that even with high PSNR values, watermark integrity can be compromised as image quality deteriorates.

Additional evaluation criteria are necessary to improve the reliability of robustness evaluation for digital watermarks. This paper evaluates the extraction of watermarks according to the degree of image manipulation attacks and calls for an analysis of watermark extraction under actual attack conditions. Moreover, for studies capable of extracting bit accuracy like the stable signature, it is essential to extract bit accuracy under various image manipulation attacks to assess watermark robustness.

5 Conclusion

This paper evaluates the robustness of watermarks against adversarial attacks and image quality degradation attacks, focusing on the conventional watermark evaluation metric PSNR and watermark extraction. The experimental results indicate that PSNR remains high even when the watermark is not extracted and the attacks proceed. Therefore, new evaluation metrics are needed to assess the robustness of watermarks based on the degree of the attack.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry (RS-2024-00415520) supervised by the Korea Institute for Advancement of Technology (KIAT), and the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

References

1. Wadhera, S., Kamra, D., Rajpal, A., Jain, A., Jain, V.: A comprehensive review on digital image watermarking, arXiv preprint arXiv:2207.06909 (2022)
2. Hwang, J.Y., Oh, S.: A Brief Survey of Watermarks in Generative AI. In: 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), pp. 1157-1160. Jeju Island, Korea, IEEE (2023)
3. Lee, J.E., Seo, Y.H., Kim, D.W.: Deep Learning Framework for Watermark-Adaptive and Resolution-Adaptive Image Watermarking. Journal of Broadcast Engineering, vol. 25, no. 2, pp. 116-179. (2020)
4. Evsutin, O., Dzhanashia, K.: Watermarking schemes for digital images: Robustness overview. Signal Processing: Image Communication 100, vol. 100 (2022)
5. Bang An et al.: Benchmarking the Robustness of Image Watermarks. arXiv preprint arXiv:2401.08573 (2024)
6. Apostolidis, K.D., Papakostas, G.A.: Digital watermarking as an adversarial attack on medical image analysis with deep learning. Journal of Imaging, vol. 8, no. 155 (2022)
7. Tancik, M., Mildenhall, B., Ng, R.: StegaStamp: Invisible Hyperlinks in Physical Photographs. In : 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
8. Pierre, F. et al.: The stable signature: Rooting watermarks in latent diffusion models. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 22409-22420, Paris, France (2023)
9. Snaptag AI, https://snaptag.ai/, last accessed 2024/06/12

정보보호산업 분쟁이 발생했다면?

누구나 무료로 편리하게 이용할 수 있는

정보보호산업분쟁조정 위원회의

분쟁조정제도를 이용해 보세요!



과학기술정보통신부
Ministry of Science and ICT

KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

정보보호산업 분쟁조정위원회
Information Security Industry Dispute Mediation Committee