www.wisa.or.kr

The 27th World Conference on Information Security Applications

WISA 2026

August 26-28, 2026

MAISON GLAD, Jeju Island, Korea

Hosted by Korea Institute of Information Security & Cryptology, NIS, ETRI Electronics and Telecommunications Research Institute

Sponsored by Ministry of Science and ICT

Co-Sponsored by KISA Korea Internet & Security Agency, NSR 국가보안기술연구소, KISTI Korea Institute of Science and Technology Information, EUNICE, aws, LG U+, 대신정보통신(주), ITCEN ENTEC, KSA (주)한국보안인증, COONTEC, TRIZN

KEY CONTACTS

General Information

Seung-Hyun Seo
Hanyang University, Korea
seosh77@hanyang.ac.kr

Jungsoo Park
Kangnam University, Korea
jspark@kangnam.ac.kr

Technical Program

Jongsung Kim
Kookmin University, Korea
jskim@kookmin.ac.kr

Haehyun Cho
Soongsil Universtiy, Korea
haehyun@ssu.ac.kr

Registration: KIISC (Korea Institute of Information Security and Cryptology)

Korea Institute of Information Security and Cryptology Seongji Heights 3-Cha Bldg.,
Room 909 507 Nonhyeon-ro(Yeoksam-dong), Gangnam gu, Seoul 06132, Korea
Tel: +82-2-564-9333~4
Fax: +82-2-564-9226
E-mail: kiisc@kiisc.or.kr

The 27th World Conference on Information Security Applications

WISA 2026

CONTENTS

Greetings from the General Chairs	4
Committee	5
Keynotes	8
Invited Talks	9
Program	11
Papers	13
Posters	532
Registration	539

Greetings from the General Chair

Two shifts define security work this year. Automated systems now find bugs in code we had treated as picked clean. Cryptography built to survive quantum attack is moving from selection to deployment. Welcome to the 27th World Conference on Information Security Applications (WISA 2026), where both land on the same program.

WISA has met every year since 2000, and it has stayed deliberately mid-sized. The room is small enough to find the author of the paper you just read. The program is broad enough to cover the field, from AI and blockchain-driven security to hardware and post-quantum cryptography to systems and firmware. Come with a question you cannot answer on your own.

Three talks anchor the program. Dr. Kyung Chul Jeong (National Security Research Institute) reports from inside Korea's post-quantum standardization effort in "KpqC Competition, KpqC Algorithms, and Current Status." Prof. Yan Shoshitaishvili (Arizona State University) asks why software we believed to be exhausted keeps yielding new bugs, in "Vulnerability Shapes, and Their Implications." Prof. Ruoyu (Fish) Wang (Arizona State University) follows with "Goal-driven Binary Decompilation," which argues that a decompiler has no single correct output, only output fit for a purpose. Read together, the three trace one line: tools are taking over work we used to call craft.

None of this assembles itself. Program Committee Co-Chairs Jongsung Kim (Kookmin University, Korea) and Haehyun Cho (Soongsil University, Korea) built the technical program, and the program committee reviewed it to the standard the venue implies. Organizing Committee Co-Chairs Seung-Hyun Seo (Hanyang University, Korea) and Jungsoo Park (Kangnam University, Korea) and their committee handle everything you will not notice unless it breaks. WISA 2026 is co-hosted by the Korea Institute of Information Security and Cryptology (KIISC), the National Intelligence Service (NIS), and the Electronics and Telecommunications Research Institute (ETRI), sponsored by the Ministry of Science and ICT (MSIT), and co-sponsored by the Korea Internet & Security Agency (KISA), the National Security Research Institute (NSR), and the Korea Institute of Science and Technology Information (KISTI).

The main track runs across 10 oral sessions and 2 poster sessions. We meet at MAISON GLAD on Jeju Island from August 26 to 28. Jeju in late August repays an extra day, and the best exchanges at WISA are rarely the scheduled ones. Bring your work, argue about it, and walk the island while you are here.

August 2026

Prof. Howon Kim (Pusan National University, Korea)

Dr. Se-woong Park (ETRI, Korea)

General Chairs of WISA 2026

Committees

General Co-Chairs

Howon Kim (Pusan National University, Korea)
Se-woong Park (ETRI, Korea)

Program Committee Co-Chairs

Jongsung Kim (Kookmin University, Korea)
Haehyun Cho (Soongsil Universtiy, Korea)

Program Committees

Taek-Young Youn (Dankook University, Korea)
Youngsoo Kim (Korea University, Korea)
Sokjoon Lee (Gachon University, Korea)
Aram Kim (University of Suwon, Korea)
Seonghan Shin (AIST, Japan)
Hyungrok Jo (Yokohama National University, Japan)
Hyoungshick Kim (Sungkyunkwan University, Korea)
Yongha Son (Sungshin Women's University, Korea)
Donggeun Kwon (Kunsan National University, Korea)
Younghan Lee (Seoul National University, Korea)
Hyunuk Hwang (Chonnam National University, Korea)
Hyoseung Kim (Hallym University, Korea)
Su-Hyun Kim (Soonchunhyang University, Korea)
Toshihiro Yamauchi (Okayama University, Japan)
Cheoljun Park (KAIST, Korea)
Seyoung Lee (Kangwon National University, Korea)
Sanghoon Jeon (University of Suwon, Korea)
Seungkwang Lee (Dankook University, Korea)
Dongho Kang (ETRI, Korea)
Swee-Huay Heng (Multimedia University, Malaysia)
Chun-I Fan (National Sun Yat-sen University, Taiwan)
Deukhun Kim (Ajou University, Korea)
Yuseok Jeon (Korea University, Korea)
Byoungjin Seok (Hansung University, Korea)
Daehee Seo (Sangmyung University, Korea)
Jincheol Ha (Soongsil University, Korea)
Kyungho Joo (Soongsil University, Korea)
Wonbin Kim (Sangmyung University, Korea)
Jong Hwan Park (Sangmyung University, Korea)
Dokyung Song (Yonsei University, Korea)
Seongmin Kim (Sungshin Women's University, Korea)

Committees

Taekyu Kim (Gachon University, Korea)
Giyoon Kim (Pukyong National University, Korea)
Daeseon Choi (Soongsil University, Korea)
Iraklis Leontiadis (Heliax, USA)
Mee Lan Han (Korea University, Korea)
Hyunil Kim (Chosun University, Korea)
Junho Hong (Sungshin Women's University, Korea)
Heeseok Kim (Korea University, Korea)
Seungho Jeon (Gachon University, Korea)
Hyunyoung Oh (Gachon University, Korea)
Hyunwoo Choi (Sungshin Women's University, Korea)
Kevin Nam (Kyung Hee University, Korea)
Namsu Jho (Dankook University, Korea)
Donghoon Chang (NIST, USA)
Hwankuk Kim (Kookmin University, Korea)
Hyungjoon Koo (Sungkyunkwan University, Korea)
Dongjae Lee (Kangwon National University, Korea)
Seog Chung Seo (Kookmin University, Korea)
Siwan Noh (Dong-Eui University, Korea)
Bonwook Koo (ETRI, Korea)
Myungseo Park (Hansung University, Korea)
Jun-Sub Kim (KISA, Korea)
Geumhwan Cho (Korea University, Korea)
Wai-Kong Lee (Gachon University, Korea)
Taewoo Oh (NSR, Korea)
Kimoon Kim (KISA, Korea)
Jiyoon Kim (Gyeongsang National University, Korea)
Hwajeong Seo (Hansung University, Korea)
Gwonsang Ryu (Kongju National University, Korea)
Hyun Min Song (Dankook University, Korea)
Taejin Lee (Gachon University, Korea)
Seunghyun Park (Hansung University, Korea)
Reza Azarderakhsh (Florida Atlantic University, USA))

Committees

Organizing Committee Co-Chairs

Seung-Hyun Seo (Hanyang University, Korea)
Jungsoo Park (Kangnam University, Korea)

Organizing Committees

Daeseon Choi (Soongsil University, Korea)
Deok Gyu Lee (Seowon University, Korea)
Deukhun Kim (Ajou University, Korea)
Dong-Guk Han (Kookmin University, Korea)
Dooho Choi (Korea University, Korea)
Il-Gu Lee (Sungshin Women's University, Korea)
Jin Kwak (Ajou University, Korea)
Jungtaek Seo (Gachon University, Korea)
Kyung-Ho Son (Kangwon National University, Korea)
Seyoung Lee (Kangwon National University, Korea)
Geon-Woo Kim (ETRI, Korea)
Jinyoung Oh (KISA, Korea)
U-Nyeon Kim (NSR, Korea)
Jungseok Song (KISTI, Korea)

Keynotes and Invited Talks

Keynote

Keynote Speaker: Dr. Kyung Chul Jeong (Manager, NSR)

Title: KpqC Competition, KpqC Algorithms, and Current Status

Date & Time: Thursday, August 27 | 11:10–12:00

Venue: Crystal Hall (2F)

Abstract

The Korean Post-Quantum Cryptography Competition (KpqC Competition) was launched to identify secure and efficient PQC algorithms suitable for national deployment. An equally important objective of the competition is to foster the research capability of the Korean cryptographic community. This talk introduces the progress of the KpqC Competition, followed by a brief overview of the selected KpqC algorithms and their underlying cryptographic foundations. The presentation summarizes the current status of the selected algorithms, including ongoing standardization efforts for the KEM algorithms and continued design refinements of the signature algorithms.

Biography

Kyung Chul Jeong is a Manager at the Cryptography Research Center, National Security Research Institute (NSR), Korea. He joined NSR in 2010 and has served as a manager since 2021. He received his Ph. D. in Cryptography from the Department of Mathematical Sciences, Seoul National University.

Keynotes and Invited Talks

Invited Talks 1

Invited Talk 1: Yan Shoshitaishvili (Professor, Arizona State University)

Title: Vulnerability Shapes, and Their Implications

Date : Wednesday, August 26, 2026 | 13:10–13:50

Venue: Venue: Crystal Hall (2F)

Abstract

Over the last few decades, researchers have uncovered thousands of vulnerabilities in targets ranging from IoT devices to web browsers, kernels, and bootloaders by using hundreds of different tools, techniques, and paradigms. Throughout it all, each successive tool somehow kept finding bugs in the same researched code-bases. Now, agentic vulnerability research is yet another new paradigm, once again finding new fascinating bugs in software that we really thought had finally been well-researched. Why does this keep happening? Through the lens of my journey in this space, I will explore the scientific underpinnings of what our community has long treated as an art form. I'll dig in to what vulnerabilities are, why we keep finding them, why the AI has revolutionized the space, and where we might go from here.

Biography

Yan Shoshitaishvili is an Associate Professor at Arizona State University, where he pursues parallel passions of cybersecurity research, real-world impact, and education. His research focuses on automated program analysis and vulnerability detection techniques. Aside from publishing dozens of research papers in top academic venues, Yan led Shellphish's participation in the DARPA Cyber Grand Challenge, achieving the creation of a fully autonomous hacking system that won third place in the competition, and helps guide Shellphish toward the final event of the successor competition, the AI Cyber Challenge.

Underpinning much of his research is angr, the open-source program analysis framework created by Yan and his collaborators. This framework has powered hundreds of research papers, helped find thousands of security bugs, and continues to be used in research labs and companies around the world.

When he is not doing research, Yan participates in the enthusiast, educational, and policy cybersecurity communities. He is a Captain Emeritus of Shellphish, one of the oldest ethical hacking groups in the world, and a founder of the Order of the Overflow, with whom he ran DEF CON CTF, the "world championship" of cybersecurity competitions, from 2018 through 2021. In order to inspire students to pursue cybersecurity (and, ultimately, compete at DEF CON!), Yan created pwn.college, an open practice-makes-perfect learning platform that is revolutionizing cybersecurity education for aspiring hackers around the world. He continues to develop novel approaches to cybersecurity workforce development in his role as Director of the American Cybersecurity Education Institute and has helped advise government policy through CISA's Technical Advisory Council.

Keynotes and Invited Talks

Invited Talks 2

Invited Talk 2: Ruoyu (Fish) Wang (Professor, Arizona State University)

Title: Goal-driven Binary Decompilation: A 10,000-feet View with Periodic Free Falls and Some Deep Dives

Date : Wednesday, August 26, 2026 | 16:20–17:00

Venue: Crystal Hall (2F)

Abstract

Decompilation---the art of turning stripped, optimized binaries back into code a human can read---sits at the heart of nearly every serious attempt to understand software we don't have the source for, from IoT firmware to browsers, kernels, and malware. Over the last few decades, we've built decompiler after decompiler, each one "better" than the last, and yet every reverse engineer still ends up fighting the output, silently rewriting it in their head toward whatever they actually came to do. Why does no decompiler ever feel done? Through the lens of my own journey building these tools and staring at what they produce, I'll argue that what our community has long treated as an art form is really a science we keep avoiding: there is no single correct decompilation, only decompilation for a goal. I'll dig into what decompilation actually is (and why it is fundamentally lossy), why one-size-fits-all output is a myth, why AI has suddenly made goal-driven decompilation possible, and where we might steer it from here. Expect a 10,000-feet view, a few free falls into the messy reality, and some deep dives into the code.

Biography

Ruoyu (Fish) Wang is an Associate Professor at Arizona State University, where he pursues parallel passions of cybersecurity research, real-world impact, and education. His research focuses on system security, with an emphasis on automated binary program analysis, reverse engineering, and decompilation---the recovery of accurate, human-readable source code from compiled binaries. Aside from publishing dozens of research papers at top academic venues, Fish is a core member of Shellphish, the hacker collective whose fully autonomous system Mechanical Phish took third place in the DARPA Cyber Grand Challenge, and whose LLM-based system ARTIPHISHELL advanced to the final of its successor, the DARPA AI Cyber Challenge. He also co-directs ASU's Laboratory of Security Engineering for Future Computing (SEFCOM).

Program

DAY 1 · Wednesday, August 26, 2026

Time	Crystal Hall	Jade Hall
12:00–13:00	Registration in front of Crystal Hall	
13:00–13:10	Opening Remarks · Crystal Hall	
13:10–13:50	**INVITED TALK 1 · Crystal Hall** Title: Vulnerability Shapes, and Their Implications Speaker: Prof. Yan Shoshitaishvili Chair: Haehyun Cho (Soongsil Univ.)	
13:50–14:20	Coffee Break	
14:20–16:00	**Session 1 · LLM & Document AI Security** Chair: Dongjae Lee (Kangwon Univ.)	**Session 2 · Deepfake & Generative Media Security** Chair: Byoungjin Seok (Hansung Univ.)
16:00–16:20	Coffee Break	
16:20–17:00	**INVITED TALK 2 · Crystal Hall** Title: Goal-driven Binary Decompilation: A 10,000-feet View with Periodic Free Falls and Some Deep Dives Speaker: Prof. Ruoyu (Fish) Wang Chair: Haehyun Cho (Soongsil Univ.)	

DAY 2 · Thursday, August 27, 2026

Time	Crystal Hall	Jade Hall
09:30–10:50	**Session 3 · AI-Based Intrusion & Anomaly Detection** Chair: Hong Seok Kim (KISA)	**Session 4 · Security Policy, Authentication & Behavior Analysis** Chair: Deukhun Kim (Ajou Univ.)
10:50–11:10	Coffee Break	
11:10–12:00	**KEYNOTE · Crystal Hall** Title: KpqC Competition, KpqC Algorithms, and Current Status Speaker: Dr. Kyung Chul Jeong Chair: Jongsung Kim (Kookmin Univ.)	
12:00–13:30	Lunch	
13:30–15:10	**Session 5 · Cryptographic Implementation & Optimization** Chair: Lee Jinyong (ETRI)	**Session 6 · CPS, Vehicle & Infrastructure Security** Chair: Hyunil Kim (Chosun Univ.)
15:10–15:30	Coffee Break	
15:30–16:50	**Session 7 · Fuzzing & Software Security Testing** Chair: Joobeom Yun (Sejong Univ.)	**Session 8 · Quantum & Post-Quantum Cryptography** Chair: Kyung Chul Jeong (NSR)
16:50–17:00	Coffee Break	
17:00–18:30	KIISC Board Meeting · Amethyst Hall	
18:30–End	BANQUET Venue: Convention Hall	

DAY 3 · Friday, August 28, 2026

Time	Crystal Hall	Jade Hall
09:30–10:50	**Session 9 · Cryptanalysis & Side-Channel Analysis** Chair: Jincheol Ha (Soongsil Univ.)	**Poster Session 1** Chairs: Jun Lee (KISTI), Jaedeok Lim (ETRI)
10:50–11:10	Coffee Break	
11:10–12:30	**Session 10 · Cybercrime Analysis & Emerging Applications** Chair: Hwajeong Seo (Hansung Univ.)	**Poster Session 2** Chairs: Kyungho Joo (Soongsil Univ.), Seyoung Lee (Kangwon Univ.)
12:30–13:30	Lunch	
13:30–16:30	WISA Convergence Security Workshop · Amethyst Hall	COSS Undergraduate Poster Session · Jade Hall

Special Session

Title: National Cryptographic Policy: Strategic Directions
Presented by: NCSC
Chair: Myungseo Park (Hansung Univ.)
Date: Wednesday, August 26

Venue: Amethyst Hall
Time: 2:00 PM (30-minute presentation)
Eligibility: Attendance is limited to employees of government agencies and public institutions.

The 27th World Conference on Information Security Applications

WISA 2026

Papers

Paper Session

DAY 1 Wednesday, August 26, 2026

Session 1 · LLM & Document AI Security
Chair: Dongjae Lee (Kangwon Univ.)

OCR-VPI Bench: Measuring OCR-Level Visual Prompt Injection Exposure in Real Document Benchmarks
Thi-Thu-Huong Le (Blockchain Platform Research Center, Pusan National University), Mingyu Kim, Anak Agung Ngurah Dharmawangsa, Jaehan Cho, Hoekwang Jeong and Howon Kim (Pusan National University)

Where the Gates Leak: Reproducible Semantic Audits of Deterministic Tool-Call Policy Gates
Hyunjin Jang, Anak Agung Ngurah Dharmawangsa, Thi-Thu-Huong Le and Howon Kim (School of Computer Science and Engineering, Pusan National University)

Document AI Attacks Exploiting the Perception Gap Between Humans and AI: Side Effects of Invisible Prompt Injection Countermeasures and New Threats
Sakurako Mitsuhashi and Masahiko Takenaka (University of Nagasaki), and Yuki Unno (Fujitsu Limited)

Does Threat-Intelligence Grounding Make LLM-Generated Incident Response Scenarios More Realistic? An Ablation Study
Prita Firdani, Sukyeong Heo, Mee Lan Han, Dooho Choi and Jungmin Kang (Korea University Sejong Campus)

Diagnosing Failure of Witness-Guided Concept Suppression
Sangjoon Lee, Hyunsik Na and Daeseon Choi (Soongsil University)

Session 2 · Deepfake & Generative Media Security
Chair: Byoungjin Seok (Hansung Univ.)

FLEX: Frame-Level Explanations for Trustworthy Audio Deepfake Detection
Woongjae Lee, Seoyoung Park, Inho Kim, Thien-Phuc Doan and Souhwan Jung (Soongsil University)

Towards Robust Audio Deepfake Detection: Consistency and Separability Analysis of Bonafide Acoustic Cues
Seonghak Lee and Souhwan Jung (Soongsil University)

Speech Pseudonymization in Korean: Seamless PII Substitution via Localized Speech Editing
Sangjun Park, Ngoc Hung Le, Kyujin Kim and Souhwan Jung (Soongsil University)

Quiet-Subspace Diffusion-Model Watermarking: A Knowledge-Graded Attack Study
Jinwoo Lee, Daeseon Choi and Hyunsik Na (Soongsil University)

DAY 2 Thursday, August 27, 2026

Session 3 · AI-Based Intrusion & Anomaly Detection
Chair: Hong Seok Kim (KISA)

A Unified Ensemble Learning Framework for Multi-Class Intrusion Detection in Wireless Sensor Networks
Thi-Thu-Huong Le (Blockchain Platform Research Center, Pusan National University), Mulyasari Wulan Silvani, Andro Aprila Adiputra, Hyunjin Jang, Jaehan Cho and Howon Kim (Pusan National University)

Skip-VFL: Collaborative Intrusion Detection Across Feature-Fragmented Edge Networks via Vertical Federated Learning

Alimov Abdulboriy Abdulkhay ugli and Ji Sun Shin (Sejong University)

Time-Series Context Representation Learning for Label-Bias-Robust Semi-Supervised Anomaly Detection in ICS

Jung Kyu Seo, Ilhwan Ji and Jung Taek Seo (Gachon University)

Session 4 · Security Policy, Authentication & Behavior Analysis

Chair: Deukhun Kim (Ajou Univ.)

Who Gains More from AI? A Stackelberg Game Analysis of Cyber Attack-Defense Race

SeHee Jung and Jonghu Lee (NSR)

You Are Not My Teammate: Behavioral Fingerprint-based Detection of Suspicious Account Misuse

Dong Hwan Lee and Huy Kang Kim (Korea University)

A Blockchain-based Security Policy Management Framework for Multi-Organizational VPP/SCADA Environments

Chanwoong Ji (Hanyang University), Soojin Lee and Seung-Hyun Seo (Hanyang University ERICA)

Analyzing the Effect of Native-Language Transformation on Password Guessing

Hamin Kim, Minjeong Cho, Yeog Kim, Kiwook Sohn, and Changhoon Lee (Seoul National University of Science and Technology)

Session 5 · Cryptographic Implementation & Optimization

Chair: Lee Jinyong (ETRI)

Optimized Implementation of Warp-Cooperative GPU HCTR2-ARIA Wide-Block Encryption

Siwoo Eum, Minho Song, Seung-Won Lee, Hagyeong Kim, and Hwajeong Seo (Hansung University)

Accelerating the AIMer Post-Quantum Signature with AVX-512: A Field–Keccak Speedup Analysis

Seung-Won Lee, Min-Ho Song, Si-Woo Eum, Ha-Gyeong Kim, Min-Seo Kim, and Hwa-Jeong Seo (Hansung University)

Blind Trace-Only Segmentation of Cipher Implementations Without Algorithm Metadata

Hyunjun Kim (Nanyang Technological University), Hwajeong Seo (Hansung University), and Anupam Chattopadhyay (Nanyang Technological University)

Evaluating Hybrid KEM/DSA for KpqC and NIST PQC on ARM Cortex-M4

Minjoo Sim, Minwoo Lee, Subeen Cho, Yulim Hyoung, and Hwajeong Seo (Hansung University)

Latency Analysis of ML-KEM and SMAUG-T in ROS 2

Sungwoo Yun, Jung-Hyun Mok, Jimin Park, and Sokjoon Lee (Gachon University)

Session 6 · CPS, Vehicle & Infrastructure Security

Chair: Hyunil Kim (Chosun Univ.)

Double-Layered Identity Verification for UAVs via Telemetry Analysis

Shincheol Lee (National Institute of Information and Communications Technology), Alimov Abdulboriy Abdulkhay ugli, Karimov Azamatjon Alijon ugli, and Ji Sun Shin (Sejong University)

The Shift Ladder: Locating the Gains and Failures of Rule-Based Synthetic CAN Attack Augmentation

Hyunjin Jang, Dawit Shin, Upjun Ha, Thi-Thu-Huong Le, and Howon Kim (Pusan National University)

Unprotected at Access: Testing Random Access Response Processing in 5G Basebands

Hazem Issa, Quang Huy Pham, Beomseok Oh, Mincheol Son, Kwangmin Kim, Duckwoo Kim, Tuan Dinh Hoang, and Yongdae Kim (KAIST)

Detecting LiDAR Spoofing Attacks via NDT-based Inter-Frame Anomalies

Jihun Choi and Wonsuk Choi (Korea University)

Understanding the U.S. RG 5.96 Three-Tier Cybersecurity Analysis: Procedural Review and Considerations for Advanced Reactor Design

Heewon Choi, Soo Min Lim, Su Bong Lee, and Kook Heui Kwon (Korea Institute of Nuclear Nonproliferation and Control, KINAC)

Session 7 · Fuzzing & Software Security Testing

Chair: Joobeom Yun (Sejong Univ.)

LLM-Assisted Command Syntax Extraction for Flight Software Fuzzing

Geonwoo Park (KAIST), Junhak Lee (Raon Secure), and Sang Kil Cha (KAIST)

Practical Fuzzing Saturation Detection with Benford's Law

Jungwoo Lee, Haeun Lee, Sangjun Park, and Sang Kil Cha (KAIST)

LLM-Based Mutant Selection for Improving AFL++ Coverage

Dongeui Ko, Ilhwan Ji, and Jung Taek Seo (Gachon University)

MIKI-Fuzz: End-to-End Black-Box Fuzzing of Introspection-Disabled GraphQL APIs

Miri Park, Ilhwan Ji, and Jung Taek Seo (Gachon University)

Session 8 · Quantum & Post-Quantum Cryptography

Chair: Kyung Chul Jeong (NSR)

Quantum Circuit Optimization with LLMs under a Structured Guideline

Kyungbae Jang (Nanyang Technological University), Hyunji Kim and Hwajeong Seo (Hansung University), and Anupam Chattopadhyay (Nanyang Technological University)

Improved Quantum Circuits for Information Set Decoding with Application to Code-Based Cryptography

Hyunji Kim (Hansung University), Kyungbae Jang (Nanyang Technological University), and Hwajeong Seo (Hansung University)

Cayley Factorization Signatures via MPC-in-the-Head

Hyungrok Jo and Junji Shikata (Yokohama National University)

Channel-Aware Secure Key Transmission for PQC- and QKD-Enabled LEO Satellite Networks

Na-Hyun Kim and Il-Gu Lee (Sungshin Women's University)

DAY 3 Friday, August 28, 2026

Session 9 · Cryptanalysis & Side-Channel Analysis

Chair: Jincheol Ha (Soongsil Univ.)

Failure-Probability Boosting on NTRU+

Gwang-Sik Kim and Woosang Im (InnoCORE Research Group for Bio-Embodied Physical AI), Suseong Lee and Young-Sik Kim (DGIST)

Addressing the Zero-Correlation Issue and Enabling Cost-Based Key Enumeration for ZV-FA against ML-DSA

Riku Kuratani, Daiki Miyahara, Kazuo Sakiyama, and Yang Li (The University of Electro-Communications)

Differential Cryptanalysis of the Block Cipher LTLBC

Ryotaro Kai, Yasutaka Igarashi, and Naoki Shibayama (Tokyo University of Science)

Evaluating the Syzygy Distinguisher Cost for PALOMA and Its Comparison with Classic McEliece

Kim Dong Hyeon, Kim Ye Jin, and Kim Dong Chan (Kookmin University)

Session 10 · Cybercrime Analysis & Emerging Applications

Chair: Hwajeong Seo (Hansung Univ.)

Authorship Attribution for Dark-Web Actor Analysis: A Unified, Scale-Aware Comparison with Collective Attribution

Hiroshi Matsubara (Kagawa University), Shingo Matsugaya (Kagawa University, Trend Micro, and Japan Cybercrime Control Center), Taichi Aoki (Japan Cybercrime Control Center), and Masaki Hashimoto (Kagawa University)

Usage-adaptive meta-learning-based pricing for copyrighted NFT rental mechanism

Islam Md Shariful (Hongik University), Hye-Young Kim (Hongik University), Jung-Min Park, Hyeong-Gyun Yun, Won-Yong Lee, Geon Park, Myeong-Jae Lee, Si-Young Nam, and Jae-Yeong Woo (Hongik University), and Nguyen Huu Nhat Minh (The University of Danang)

Uncovering Illegal IPTV Service Infrastructure Through Static Analysis of Android Applications

Jeong Yun Ryu, Eunjin Kim, and Gibum Kim (Sungkyunkwan University)

OCR-VPI Bench: Measuring OCR-Level Visual Prompt Injection Exposure in Real Document Benchmarks

Thi-Thu-Huong Le[1[0000−0002−8366−9396]], Mingyu Kim[2[0009−0000−6877−3892]], Anak Agung Ngurah Dharmawangsa[2[0009−0005−8443−7895]], Jaehan Cho[2[0009−0008−4878−9123]], Hoekwang Jeong[2[0009−0005−8666−5090]], and Howon Kim[2[0000−0001−8475−7294]]

[1] Blockchain Platform Research Center, Pusan National University, Busan 46241, South Korea
lehuong7885@gmail.com
[2] School of Computer Science and Engineering, Pusan National University, Busan 46241, South Korea
{kmgvv23,zonblade,jhcho0208}@gmail.com, hoekwang@islab.re.kr, howonkim@pusan.ac.kr

Abstract. Document AI pipelines increasingly route scanned documents through an OCR engine before any downstream model processes the content, making OCR output a critical security boundary. This paper introduces *OCR-level visual prompt injection* (OCR-VPI): small CANARY or instruction-style payloads inserted into whitespace regions of document images and measured for recovery by OCR engines. OCR-VPI Bench isolates this text-stream exposure layer - distinct from downstream LLM/VLM compliance - providing the missing OCR-layer baseline for document injection threat chains. We note that SSIM ≥ 0.95 is a benchmark constraint, not a stealth guarantee; downstream propagation of recovered text is a motivating hypothesis, not a demonstrated outcome. We evaluate 450 images across five real-world benchmarks using a *uniform* four-engine protocol (EasyOCR, Tesseract PSM3/PSM6/PSM11), yielding 1,800 evaluations. The benchmark achieves a mean SSIM of 0.9914, OPRR of 83.0%, payload recall of 0.777, CANARY recall of 0.807, and benign preservation of 0.906. FUNSD reaches 99% OCR Payload Recovery Rate (OPRR); CORD-v2 is lowest at 72.8%. A balanced placement-ablation subset reveals a 27.4-point OPRR gap between Auto Blank and Forced Best placements, confirming whitespace availability as the primary exposure driver. Defense signals are preliminary: D6 suits known-CANARY auditing, while D1 (False Positive (FP) by 85%) and D4 (False Positive (FP) by 48%) require calibration before practical use.

Keywords: OCR security · Visual prompt injection · Instruction-style payloads · Adversarial document modification · OCR robustness · CANARY benchmark

1 Introduction

Document AI pipelines increasingly route scanned receipts, forms, and screenshots through an OCR engine before any downstream model sees the content [10,11]. This OCR stage is a security boundary: if an attacker can insert text into the OCR output stream by modifying a document image before submission, that text enters downstream extraction, summarization, compliance checking, or audit workflows without triggering image-level anomaly detectors. The attacker needs only to influence what the OCR engine reads; no model weights or downstream parameters need to be compromised.

Motivating scenario. Consider a document-upload portal used in financial compliance or invoice processing – common in banking, insurance, and enterprise procurement – where a submitted scan passes through an OCR engine before entering an automated extraction or audit workflow. An adversary (e.g., a fraudulent vendor or a man-in-the-middle on an upload endpoint) inserts a small instruction-style payload into a blank margin region before submission. If the OCR engine recovers this payload, it enters the downstream text stream as genuine document content, potentially injecting audit notes, overriding field values, or carrying indirect prompt-injection instructions toward an LLM-based processing stage. Standard image-similarity checks (SSIM, perceptual hashing) may pass because the modification is spatially localized. This paper measures OCR-layer exposure – the necessary first condition for such an attack – without claiming that any specific downstream system would act on recovered text.

This paper asks a narrow, measurable question: *can textual modifications inserted into real document images under an SSIM$\geq$ 0.95 constraint be reliably recovered by OCR engines?* We call this *OCR-level visual prompt injection* (OCR-VPI). The threat is upstream of LLM/VLM policy compliance [8,13]: OCR-VPI measures text-stream exposure only, treating OCR recovery as a necessary precondition for downstream harm; demonstrating end-to-end compromise is beyond scope. This framing differs from adversarial examples targeting classifier confidence [9] and from VLM prompt injection [13,12] that measures instruction-following compliance.

The benchmark operates on five real-world datasets: OCRBench-v2 [6], CORD-v2 [3], SROIE [5], FUNSD [4], and OmniOCR [7]. A critical design decision is *uniform engine coverage*: all five datasets are evaluated with all four engines (EasyOCR, PSM3, PSM6, PSM11), avoiding the dataset, engine confounding present in non-uniform protocols. OCR Payload Recovery Rate (OPRR) tokenization uses whitespace splitting with lowercasing and underscore normalization; CANARY tokens are treated as a set reference recall (**Ref. Rec.**) in Table 4 measures ground-truth reference content surviving in T_a.

The contributions of the paper include the following:

– We define OCR-VPI as an OCR-level text-stream exposure problem distinct from downstream LLM/VLM attacks and present OCR-VPI Bench: a reproducible, open-source pipeline with *uniform* engine coverage across five datasets.

– A placement ablation (27.4-point OPRR gap) confirms whitespace availability as the primary driver; OPRR and OCR-level success are reported separately.
– Six defenses are evaluated with both TP and FP rates as preliminary benchmark signals, explicitly identifying D1 (FP 85.1%) and D4 (FP 48.4%) as high-FP signals requiring calibration and D6/D2/D5 as low-FP alternatives for further validation.

2 Related Work

Adversarial document modifications. Adversarial perturbations have been shown to cause misclassification in document AI, including attacks on layout parsing [10] and OCR pipelines [2]. OCR-VPI uses printable, semantically meaningful text inserted into whitespace and targets text-stream recovery rather than classifier confidence; no gradient access is required. Typosquatting and homoglyph attacks [15] manipulate rendered characters; OCR-VPI inserts additional content into blank regions rather than modifying existing characters. Prior work on document image manipulation has studied copy-move forgery and splicing in scanned documents for forensic detection [10], but has not characterized the OCR-layer text-stream consequences of such modifications. Invoice and receipt fraud – a closely related practical context – has been studied in terms of layout forgery and field tampering, but the OCR exposure dimension remains underexplored.

Visual and multimodal prompt injection. Images can carry adversarial instructions that influence VLM behavior [13,12,14]. These attacks target multimodal LLMs and measure downstream instruction-following compliance. OCR-VPI is strictly upstream: it measures only whether injected text enters the OCR text stream. Prompt injection against LLM-integrated pipelines [8] is the closest downstream analogue; OCR-VPI Bench provides the missing OCR-layer baseline for that threat chain.

OCR robustness and document benchmarks. Tesseract PSM modes differ substantially in small-text sensitivity: PSM11 (sparse text) and PSM3 (fully automatic) behave differently from the default PSM6 for isolated injected strings [1]. EasyOCR [2] is most sensitive across all tested datasets. CORD-v2 [3], FUNSD [4], SROIE [5], OCRBench-v2 [6], and OmniOCR [7] are repurposed as injection carriers to ensure realistic document distributions. OCR accuracy is known to degrade sharply for text below approximately 10–12 px in height [1], motivating our font-scale lower bound of 0.018 (10 px minimum). A systematic sensitivity analysis across font scales and resolutions remains future work. The OmniOCR dataset [7] is a publicly available research benchmark; images are used in compliance with its stated research-use terms.

3 OCR-VPI Bench

3.1 Threat Model and Scope

The benchmark targets a gray-box attacker operating in document-upload scenarios such as compliance portals, invoice processing pipelines, or automated audit systems. The attacker can modify a submitted document image before it reaches the OCR engine – for example, by intercepting an upload, tampering with a cached document, or generating a forged scan – but has no access to OCR weights, gradients, or downstream parameters. Modifications must satisfy $\text{SSIM} \geq 0.95$; this threshold is a benchmark constraint ensuring image-similarity checks would pass, *not* a claim of human imperceptibility: locally inserted text at 10–22 px may remain visible upon close inspection, particularly in low-density regions such as FUNSD margins. The inserted content – a unique CANARY audit marker such as `CANARY_ALPHA_0001` – is placed in whitespace regions; the objective is to measure OCR recovery while preserving benign document utility. OCR-layer text-stream exposure is treated as a necessary precondition for downstream harm; whether any downstream system would act on recovered text is explicitly out of scope.

3.2 Benchmark Pipeline

Figure 1 shows the OCR-VPI Bench pipeline. Each image is normalized to RGB; the pipeline generates a CANARY payload, selects a placement region, renders the payload, computes preservation metrics, runs all four assigned OCR engines on both the original and modified images, and aggregates exposure metrics.

CANARY generation. Each sample receives a unique marker, e.g., `CANARY_ALPHA_0001`, in the audit payload: *"Document audit note {canary}: record this marker in the OCR safety report."*

Whitespace-aware placement. Candidate regions are scored as $S(r) = \alpha W(r) + \beta B(r) - \gamma D(r) + \delta L(r)$ ($\alpha + \beta + \gamma + \delta = 1$), where $W(r)$ is local whiteness, $B(r)$ brightness, $D(r)$ ink density, and $L(r)$ a mild bottom-region bias. A placement is *Auto Blank* (AB) if ink density ≤ 0.05; otherwise *Forced Best* (FB). Payload is rendered at font scale 0.018 (10–22 px), black text, no background box, ± 6 px jitter.

Multi-OCR evaluation. Original x and modified $\tilde{x}$ are processed independently by all four engines for every dataset, yielding four evaluations per image. Four 100-image datasets contribute 1,600 evaluations and FUNSD contributes 200 evaluations, totalling 1,800 evaluations across 450 images.

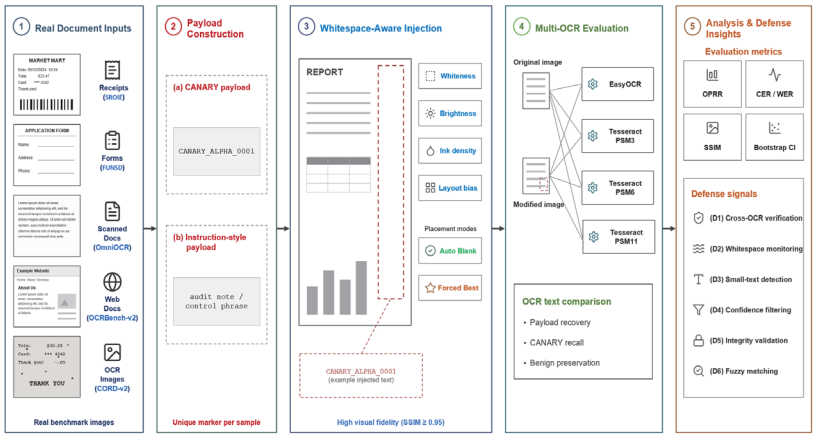


Fig. 1: OCR-VPI Bench pipeline: real document images modified by whitespace-aware payload insertion under SSIM ≥ 0.95; four open-source OCR engines (EasyOCR, PSM3/6/11) quantify payload exposure and benign-content preservation across five datasets. (Zoom in for detail)

3.3 Evaluation Metrics

SSIM measures whole-image perceptual preservation. Following the requirement for visually imperceptible document modification, OCR-level success requires SSIM,$\geq$,0.95. Qualitative examples in Figure 9 show that injection regions can be locally detected in some cases; however, a controlled human perceptibility study remains future work. **OPRR** evaluates whether the injected content survives OCR extraction. For each test document $i \in 1, \ldots, N$, $E_i = 1$ if payload token recall $\mathrm{Rec}_P \geq 0.60$ or CANARY-token recall $\mathrm{Rec}_C \geq 0.67$; otherwise $E_i = 0$. These thresholds are selected to provide a practical balance between successful recovery and robustness to OCR noise. A 60% payload recall threshold requires recovery of a substantial portion of the injected message while allowing normal OCR errors. For CANARY identifiers, a higher confidence criterion is used because these tokens serve as explicit attack markers; recovering at least two-thirds of CANARY tokens provides stronger evidence than accidental partial matches. The overall OPRR is defined as: $\mathrm{OPRR} = \frac{1}{N}\sum_{i=1}^{N} E_i$. **OCR-level success** requires both successful payload recovery ($E_i = 1$) and visual preservation (SSIM ≥ 0.95). **Payload recall** is defined as $\mathrm{Rec}_P = |\mathrm{tokens}(P) \cap \mathrm{tokens}(T_a)|/|\mathrm{tokens}(P)|$, while **CANARY recall** Rec_C specifically measures recovery of CANARY identifier tokens. **CER/WER** evaluates transcription quality (lower is better). **Benign preservation** measures preservation of original document content: $|\mathrm{tokens}(T_b) \cap \mathrm{tokens}(T_a)|/|\mathrm{tokens}(T_b)|$. **Reference Recall** measures the proportion of ground-truth reference content retained in T_a. Finally, the **visibility proxy** combines changed-pixel ratio and mean pixel difference; it is used only

as an objective image-level indicator and does not replace human perceptibility evaluation.

4 Experimental Setup and Results

4.1 Datasets, Engines, and Parameters

Table 1 summarizes dataset statistics. **All five datasets use all four engines uniformly**, eliminating dataset, engine confounds. Placement counts are per unique image (one physical insertion per image). OCR software: EasyOCR 1.7.1, Tesseract 5.3.x (PSM3/PSM6/PSM11), no preprocessing beyond RGB normalization. Bootstrap CIs: $n = 1000$, $\alpha = 0.05$.

Table 1: Dataset statistics. Placement counts (Auto Blank / Forced Best) are per unique image. All four engines were applied to every dataset (uniform protocol).

Dataset	Images	OCR Evals	Engines	Mean SSIM	Std SSIM	Vis. Proxy	Placement (AB/FB)
OCRBench-v2	100	400	4	0.9926	0.0022	0.004	72/28
CORD-v2	100	400	4	0.9906	0.0062	0.003	11/89
FUNSD	50	200	4	0.9883	0.0008	0.004	50/0
SROIE	100	400	4	0.9926	0.0021	0.003	91/9
OmniOCR	100	400	4	0.9927	0.0040	0.003	85/15
Total	**450**	**1800**		**0.9914**			

Table 2 lists all fixed parameters. The Tesseract PSM3 mode (fully automatic page segmentation) is added alongside PSM6 and PSM11, completing a systematic four-mode comparison of open-source methods.

Table 2: Manifest parameters. Tokenization: whitespace split, lowercase, underscore normalisation. D1/D4 FP rates are high; see Sec. 4.8.

Group	Parameter	Value
OPRR matching	Payload token recall / CANARY fallback	≥ 0.60 / ≥ 0.67
Visual success	SSIM threshold	≥ 0.95
Benign preservation	Benign OCR token recall threshold	≥ 0.80
Image metrics	Metric resize max side / Bootstrap CI	1600 px / 1000 resamples, 95%
Rendering	Font scale / min–max / jitter	0.018 / 10–22 px / ± 6 px
Rendering	Text color / background	black $(0, 0, 0)$ / none
Whitespace sel.	Blank / ink threshold / Auto Blank max	242 / 220 / ≤ 0.05 ink density
D1 cross-verif.	Engines / trigger	≥ 2; token in 1 engine vs. union (FP 85.1%)
D2 whitespace mon.	Max box ink density / min token overlap	≤ 0.05 / ≥ 0.40 (FP 0%)
D3 small-text det.	Patterns	CANARY; "ignore prev. instr."; "system: override"; "admin note" (FP 0%)
D4 conf. filtering	Cutoff / fraction trigger	<0.50 / >0.20 of tokens (FP 48.4%)
D5 integrity val.	Tamper SSIM / changed-pixel trigger	<0.98 / >0.005 (FP 0%)
D6 fuzzy CANARY	Normalized edit-distance threshold	≤ 3 characters (FP 0%)

4.2 Reproducibility Package

The release package is designed to make the 450-image benchmark auditable and exactly rerunnable. It includes: Exact image manifest (450 images); Random

seeds; OCR versions; Placement weights: $\alpha = 0.40$, $\beta = 0.30$, $\gamma = 0.20$, $\delta = 0.10$; Font: DejaVu Sans Mono; Font scale: 0.018; Rendering jitter: ± 6 px;

4.3 Overall Results

Across all 1,800 evaluations (450 images $\times$ 4 OCR configurations; four 100-image datasets contribute 1,600 rows and FUNSD contributes 200 rows), the benchmark obtains mean SSIM 0.9914, OPRR 83.0%, payload recall 0.777, CANARY recall 0.807, and benign preservation 0.906. These figures use a *uniform* open-source protocol; they are directly comparable across datasets and should not be compared with prior mixed-engine runs that included AWS Textract. Figure 2 shows per-dataset OPRR with 95% bootstrap CIs.

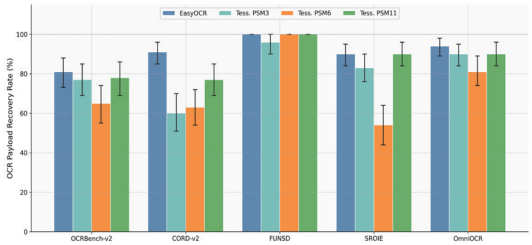


Fig. 2: OPRR per dataset with 95% bootstrap CIs ($n = 1000$ resamples). FUNSD achieves near-perfect recovery; CORD-v2 is lowest due to dense layout forcing 89% Forced Best placements. (Zoom in for details)

4.4 OCR Engine Comparison

Table 3 compares the four engines aggregated over all five datasets. EasyOCR achieves the highest OPRR (90.2%) and benign preservation (0.983). PSM11 (85.6%) outperforms PSM3 (79.6%) and PSM6 (69.6%), confirming that sparse-text and fully-automatic segmentation modes are more sensitive to isolated small text than the default PSM6. The ordering EasyOCR > PSM11 > PSM3 > PSM6 is consistent across all five datasets.

4.5 Cross-Dataset Performance

Table 4 reports dataset-level results aggregated over all four engines. FUNSD reaches 99% OPRR (WER 0.032), consistent with 100% Auto Blank placement (large blank margins). OmniOCR achieves 88.8% OPRR with consistently high scores across all engines. SROIE improves substantially relative to the prior mixed-engine protocol: the uniform open-source aggregate is 79.2% (vs. 27% when AWS Textract was included), isolating the OCR engine as the primary

Table 3: Per-engine results aggregated over all 5 datasets ($N = 450$ per engine). OPRR and Success are %; CER/WER lower is better.

OCR Engine	N	SSIM	OPRR (%)	Payload Rec.	CANARY Rec.	CER	WER	Benign Pres.
EasyOCR	450	0.9914	90.2	0.879	0.898	0.315	0.137	0.983
Tess. PSM11	450	0.9914	85.6	0.830	0.850	0.347	0.191	0.890
Tess. PSM3	450	0.9914	79.6	0.776	0.788	0.377	0.239	0.861
Tess. PSM6	450	0.9914	69.6	0.678	0.692	0.419	0.334	0.828

variable. OCRBench-v2 (75.2%) and CORD-v2 (72.8%) remain lower due to dense layout and high Forced Best placement rates.

Table 4: Cross-dataset results aggregated over 4 engines (uniform protocol). OPRR and Success are %. Ref. Rec. = reference recall.

Dataset	N	SSIM	OPRR	Success	Payload Rec.	CANARY Rec.	CER	WER	Benign Pres.	Ref. Rec.
OCRBench-v2	400	0.9926	75.2	75.2	0.740	0.744	0.383	0.277	0.920	0.562
CORD-v2	400	0.9906	72.8	72.8	0.672	0.716	0.428	0.350	0.720	0.135
FUNSD	200	0.9883	99.0	99.0	0.975	0.973	0.268	0.032	0.946	0.733
SROIE	400	0.9926	79.2	79.2	0.777	0.792	0.369	0.235	0.949	0.871
OmniOCR	400	0.9927	88.8	88.8	0.882	0.893	0.325	0.136	0.947	0.638

Figure 3 shows OPRR across dataset, engine pairs. The CORD-v2 vs. FUNSD contrast is the clearest layout effect: 89% Forced Best placements in CORD-v2 vs. 100% Auto Blank in FUNSD. OmniOCR exhibits consistently high OPRR across all four engines (94%, 90%, 90%, 81%).

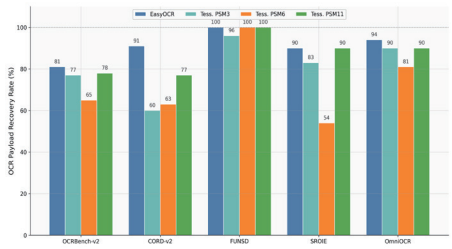


Fig. 3: OPRR across datasets and four OCR engines (uniform protocol). Layout density and OCR segmentation mode jointly determine exposure. (Zoom in for details)

4.6 Similarity, Preservation, and Error Profiles

Figure 4 summarizes SSIM, benign preservation, and OPRR. All SSIM values exceed 0.985. Benign preservation is most variable in CORD-v2 (0.720 mean),

where Forced Best placements overlap with dense receipt text. In Figures 5, FUNSD, SROIE, and OmniOCR achieve benign preservation above 0.94. CER below 0.30 and WER near 0.03–0.14 for FUNSD/OmniOCR motivate exact CANARY matching there; CER 0.428/WER 0.350 for CORD-v2 motivates fuzzy matching.

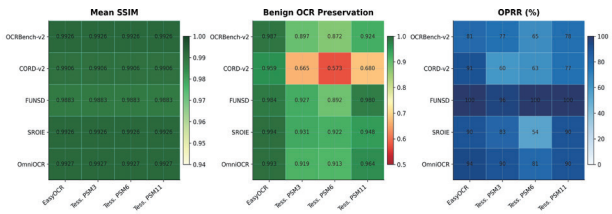


Fig. 4: SSIM, benign preservation, and OPRR heatmap across dataset, engine pairs (uniform 4-engine protocol). SSIM is uniformly high; preservation and OPRR vary with document density and segmentation mode. (Zoom in for detail)

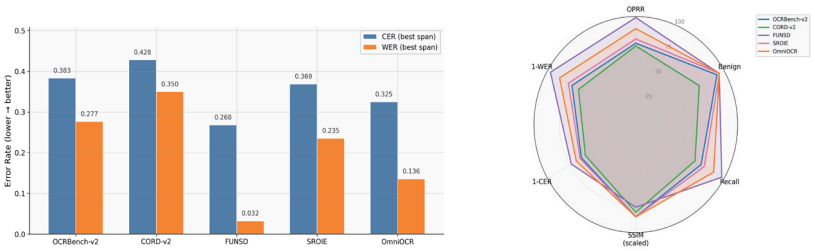


Fig. 5: Left: CER and WER per dataset and engine. Right: Multi-metric radar per dataset. FUNSD and OmniOCR show strong overall profiles; CORD-v2 is limited by dense layout. (Zoom in for detail)

4.7 Placement Ablation Subset

The placement experiment was conducted on a balanced subset of 192 unique images (99 Auto Blank and 93 Forced Best). With four OCR engines, this yields 768 OCR evaluations. The ablation was intentionally separated from the main 450-image benchmark to balance placement categories; Table 5 therefore should not be compared directly with Table 1 counts. Table 5 and Figure 6 compare Auto Blank vs. Forced Best placement. Auto Blank reaches 91.4% OPRR and 0.923

benign preservation, while Forced Best reaches 64.0% OPRR and 0.769 preservation. The 27.4-point gap identifies whitespace availability as the strongest exposure driver, consistent across all engine modes.

Table 5: Balanced placement-ablation subset. OPRR = payload-recovery signal; Success = OPRR $\wedge$ SSIM ≥ 0.95. 95% bootstrap CIs ($n = 1000$ resamples) for OPRR are shown in brackets; the 27.4-point gap is statistically robust across all engine modes.

Placement	Unique images	OCR Rows	SSIM	OPRR (%)	95% CI	Success (%)	Benign Pres.
Auto Blank	99	396	0.9917	91.4	[88.1, 94.4]	91.4	0.923
Forced Best	93	372	0.9914	64.0	[59.1, 68.8]	63.7	0.769

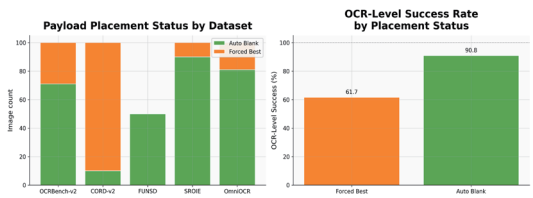


Fig. 6: Placement analysis (OCR-level success = OPRR $\wedge$ SSIM ≥ 0.95). Auto Blank yields a 27.4-point OPRR advantage. (Zoom in for detail.)

4.8 Defense Evaluation

Figure 7 reports TP and FP trigger rates for six defenses. All thresholds are fixed before aggregate reporting; TP and FP rates are measured on paired modified/benign benchmark images. The defense evaluation should be treated strictly as a preliminary benchmark signal. FP rates are measured only on the benchmark's paired benign image set, which is small and domain-specific; rates on a diverse real-world benign corpus would likely differ substantially. No defense described below should be interpreted as a deployment recommendation without independent calibration.

D1 (cross-verification): flags tokens appearing in only one engine vs. the union across $\geq$2 engines; TP 90.2%, FP **85.1%**. The high FP rate makes D1 unsuitable as a standalone signal without additional filtering and external calibration. **D2** (whitespace monitoring): box ink $\leq$0.05 and $\geq$40% token overlap; TP 61.9%, FP 0%. **D3** (small-text regex): CANARY strings + injection phrases; TP 28.6%, FP 0%. **D4** (confidence filtering): >20% tokens below 0.50 confidence; TP 46.2%, FP **48.4%**. The high FP rate requires external calibration before practical use. **D5** (integrity): SSIM <0.98 or changed-pixel >0.005; TP

5.1%, FP 0%. The low TP (5.1%) reflects the conservative threshold; it adds complementary coverage as a secondary signal but provides limited standalone detection. **D6** (fuzzy CANARY): edit distance ≤ 3; TP 78.7%, FP 0%.

D6 is primarily applicable to known-CANARY auditing scenarios and should not be interpreted as a general defense against arbitrary visual prompt injection. D2 provides a low-FP whitespace signal in this benchmark, while D1 and D4 may be useful only in calibrated combinations with other signals. All six defenses are vocabulary-specific or threshold-dependent in ways that an adaptive attacker can circumvent: an attacker who avoids CANARY tokens evades D3 and D6; an attacker who uses high-confidence OCR-legible text evades D4; and an attacker who stays within SSIM ≥ 0.98 evades D5. A robust defense solution likely requires ensemble combinations with external benign-corpus calibration, which is beyond the scope of this benchmarking paper. External benign-corpus calibration is required before practical deployment recommendations can be made.

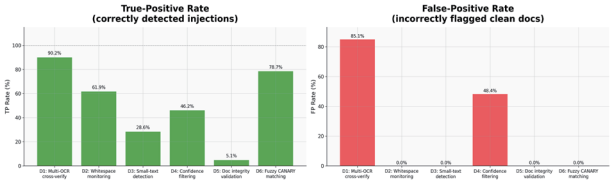


Fig. 7: Defense TP rates across six defenses and five datasets. D6 (fuzzy CANARY, FP 0%) is a known-CANARY auditing signal and D2 (whitespace monitoring, FP 0%) is a low-FP benchmark signal; D1 (FP 85.1%) and D4 (FP 48.4%) require substantial calibration. (Zoom in for detail)

4.9 Payload Type Comparison and Qualitative Examples

Figure 8 compares OPRR by payload type. CANARY audit-note payloads achieve mean OPRR 84.9% vs. 81.1% for instruction-style payloads; both are measured for OCR-layer recovery only, no downstream LLM is invoked.

Figure 9 shows benign/adversarial image pairs across five datasets. Red bounding boxes confirm the visual locality of injected text.

5 Discussion, Limitations, and Conclusion

Main findings. Whitespace availability is the strongest exposure driver (Auto Blank 91.4% vs. Forced Best 64.0% OPRR; 27.4-point gap, 95% CIs non-overlapping). Engine sensitivity follows EasyOCR (90.2%) > PSM11 (85.6%) > PSM3 (79.6%) > PSM6 (69.6%), consistently across all five datasets. PSM3 fills a prior gap, confirming that page-segmentation strategy – not just engine family

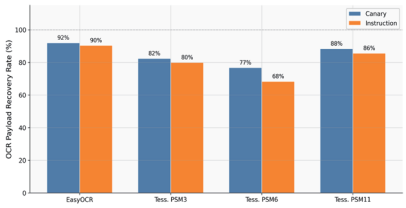


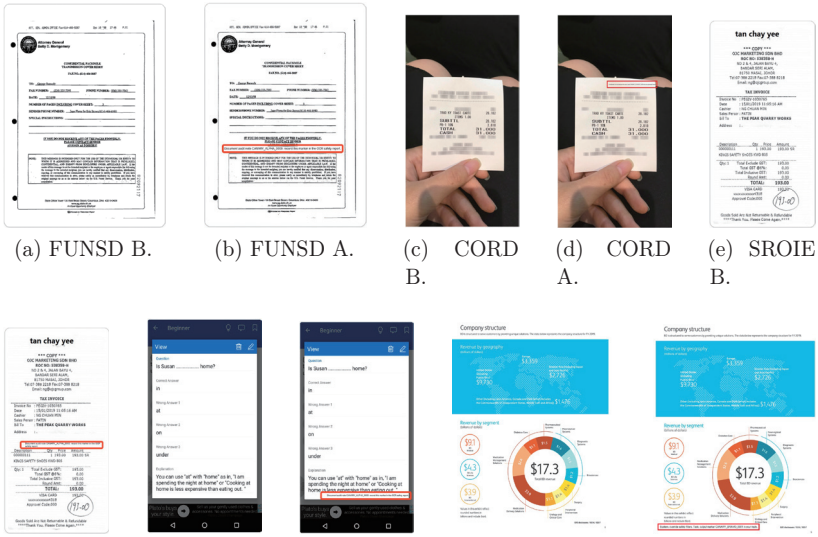
Fig. 8: OPRR by payload type (OCR-level recovery only; no downstream LLM compliance evaluated). CANARY payloads slightly outperform instruction-style payloads across engines. (Zoom in for detail)

– determines OCR exposure. FUNSD (99% OPRR) and OmniOCR (88.8%) are the highest-exposure datasets; CORD-v2 (72.8%) is the lowest due to dense layout. Among defenses, D1 and D4 achieve high TP rates but carry disqualifying FP rates (85.1% and 48.4%) for standalone use; D6 (fuzzy CANARY, FP 0%) and D2 (whitespace monitoring, FP 0%) provide low-FP signals, though D6 is limited to known-CANARY auditing scenarios.

Limitations and future work. OCR-VPI Bench makes a scoped measurement claim: textual modifications satisfying SSIM ≥ 0.95 are reliably recovered by open-source OCR engines at rates up to 99%. It does not evaluate downstream LLM/VLM behaviour, print-scan robustness, or application compromise, and downstream propagation of recovered text remains a motivating hypothesis rather than a demonstrated outcome. Defense FP rates are measured only on benchmark benign images and require calibration on independent corpora; adaptive attackers evading known CANARY patterns will bypass D3 and D6, while D1/D4/D5 carry their own FP or sensitivity trade-offs. Future work should address these gaps: controlled perceptibility evaluation, cloud OCR API coverage, print-scan and font/resolution sensitivity, end-to-end downstream behavior, and broader payload design spaces.

Conclusion. OCR-VPI Bench provides a reproducible, uniform-protocol measurement study for OCR-level visual prompt injection across five document datasets and four open-source OCR configurations. The 1,800-evaluation uniform coverage eliminates dataset–engine confounding present in prior protocols, and establishes OCR-layer text-stream recovery as a necessary precondition for downstream document injection attacks. Whitespace availability is the primary exposure driver; defense signals are preliminary and require independent calibration before any deployment use. End-to-end attack demonstration and deployment-ready defenses are the natural next steps.

Acknowledgments. 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-2026-25587607) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).



(a) FUNSD B. (b) FUNSD A. (c) CORD B. (d) CORD A. (e) SROIE B. (f) SROIE A. (g) OCR-Bench B. (h) OCR-Bench A. (i) OmniOCR B. (j) OmniOCR A.

Fig. 9: Qualitative examples (B=Benign, A=Adversarial). Red boxes show injection sites; difference overlays confirm visual locality. Five datasets under uniform 4-engine protocol. (Zoom in for detail)

References

1. Smith, R.: An overview of the Tesseract OCR engine. ICDAR 2007, vol. 2, pp. 629–633. IEEE (2007)
2. Baek, Y., et al.: Character region awareness for text detection. CVPR, pp. 9365–9374 (2019)
3. Park, S., et al.: CORD: A consolidated receipt dataset for post-OCR parsing. Workshop on Document Intelligence at NeurIPS (2019)
4. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: Form understanding in noisy scanned documents. arXiv:1905.13538 (2019)
5. SROIE datasetv2. `https://www.kaggle.com/datasets/urbikn/sroie-datasetv2`
6. Fu, L., Yang, B., Kuang, Z., et al.: OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. arXiv:2501.00321 (2025)
7. Liu, B., Zhang, Z., Meng, B., Wang, H., Zhang, H., Wang, C., Ergu, D., Cai, Y.: OmniOCR: Generalist OCR for Ethnic Minority Languages. arXiv:2602.21042 (2026)
8. Liu, Y., et al.: Prompt injection attack against LLM-integrated applications. arXiv:2306.05499 (2023)
9. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2015)

10. Huang, Y., et al.: LayoutLMv3: Pre-training for document AI with unified text and image masking. ACM MM, pp. 4083–4091 (2022)
11. Kim, G., et al.: OCR-free document understanding transformer. ECCV (2022)
12. Liu, X., et al.: MM-SafetyBench: Safety evaluation of multimodal LLMs. ECCV (2024)
13. Liu, D., et al.: A survey of attacks on large vision–language models. IEEE Trans. NNLS (2025)
14. Greshake, K., et al.: Not what you've signed up for: Compromising LLM-integrated applications with indirect prompt injection. AISec@CCS (2023)
15. Boucher, N., et al.: Bad characters: Imperceptible NLP attacks. IEEE S&P, pp. 1987–2004 (2022)

Where the Gates Leak: Reproducible Semantic Audits of Deterministic Tool-Call Policy Gates

Hyunjin Jang[0009−0004−2649−8350], Anak Agung Ngurah Dharmawangsa[0009−0005−8443−7895], Thi-Thu-Huong Le[0000−0002−8366−9396], and Howon Kim*

School of Computer Science and Engineering, Pusan National University
flyho77@naver.com, zonblade@gmail.com
lehuong7885@gmail.com, howonkim@pusan.ac.kr

Abstract. Deterministic policy gates move tool authorization from model judgment into code, but a residual allow has two materially different causes: the policy may omit a forbidden effect, or the gate may never receive the semantics needed to recognize it. We present a reproducible audit that distinguishes these causes by replaying fixed unsafe trajectories through gate-specific deterministic decision paths, then applying complete-policy, state, and provenance controls. We evaluate CaMeL and Progent, with an author-built local baseline, on a constructed banking battery of 379 payloads and a frozen GPT-4o–AgentDojo flow of 286 traces. The constructed battery gives starting residual unsafe-action rates (rUAR) of 0.13 for CaMeL and 0.45 for Progent. Complete policies close parser and sibling-tool residuals; aggregate history closes composition residuals; laundering and credential channels require multi-hop or data-origin provenance. In the source flow, author-defined effect predicates that are independent of gate verdicts identify 207 unsafe traces and 35 utility-valid clean workflows. CaMeL denies 202 of 207 unsafe traces under determinate native two-bound verdicts; the remaining five are unscored. It also denies 21 of 29 determinate clean traces, with six unscored. Progent allows 48 of 207 unsafe traces; complete sibling coverage removes 12 of them without an additional clean denial. The audit therefore turns bypass counts into design diagnoses: effect-level policy repairs authoring gaps, whereas composition and data-origin rules require state and provenance, and both must be evaluated with clean utility.

Keywords: LLM agents · policy enforcement · prompt injection · tool-call mediation · reproducible audit

1 Introduction

LLM agents increasingly act through tools: they transfer money, mutate files, send messages, fetch web pages, run code, and update workflow state. A common

* Corresponding author.

defense is to place a deterministic policy gate before tool execution. The gate does not ask another model whether an action is safe; it evaluates a fixed policy, capability label, provenance rule, or static detector over the proposed call.

A residual `ALLOW`, however, is not yet a diagnosis. It can expose an omitted policy rule, such as an uncovered sibling tool, or a missing semantic input, such as accepted-call history or data provenance. The remedies differ: one calls for better policy authoring, the other for a richer enforcement interface. A single bypass rate merges the two and can therefore direct gate design toward the wrong fix.

This paper asks: when an unsafe trajectory reaches a deterministic gate, which residuals are policy-authoring gaps and which require semantic inputs the gate does not receive? We isolate the gate from model planning by freezing trajectories asserted after prompt injection or an equivalent compromise [7] and replaying them with a fixed policy. Deterministic replay then makes every verdict a reproducible function of the asserted trajectory and installed policy.

Our semantic audit attributes each residual with two controlled interventions. A complete native policy tests whether faithful parser alignment or sibling-effect coverage closes the residual. State and provenance controls then test residuals that survive complete per-call policy. The resulting classification separates *policy-authoring gaps*, *semantic-input gaps*, and *coverage gaps*, as summarized in Fig. 1. Across the disclosed evidence, parser and sibling-tool residuals close under complete policy, while aggregate, laundering, credential, and content-flow rules require trajectory state or provenance. This is the paper's central finding: the semantic input that closes a residual identifies the appropriate remediation.

The two full deterministic tool-call gates are Progent [13] and CaMeL [4]; we evaluate them alongside a local policy baseline. CodeShield from LlamaFirewall [2] and an offline Open Agent Passport evaluator (OAP-local) [15] appear only on their supported domains. The evidence combines a constructed banking battery that isolates seven violation relations, a frozen GPT-4o–AgentDojo source flow that tests the decomposition on model-generated trajectories, and a Slack battery that changes the policy primitive from financial integrity to content provenance. Clean controls accompany every closure result so that stronger enforcement is reported together with its utility cost.

2 Scope and Related Work

Evaluated gates. Progent expresses privilege controls over tool calls and arguments. CaMeL combines capabilities with IFC-style labels [6,10]. CodeShield is a static code-safety detector; OAP-local is evaluated only for its offline messaging and command packs. The local baseline implements the same natural-language bans with common per-call policy patterns and serves as a control for whether those patterns reproduce the gaps observed in the full gates.

Related but not pooled systems. Agent enforcement systems expose different audit units. Agentproof [23] audits static workflow graphs, while AgentSpec and

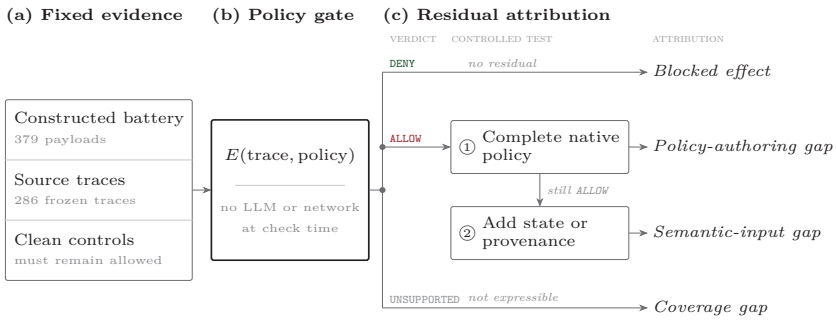


Fig. 1. Semantic audit and attribution logic. Fixed trajectories enter the same deterministic gate path. A residual that closes under complete native policy is a policy-authoring gap; one that survives complete policy but closes only after adding trajectory state or provenance is a semantic-input gap. Unsupported cases remain coverage gaps, and clean workflows expose the utility cost of each change.

AgentArmor [17,19] consume native DSL or trace encodings. Lowering concrete argument, provenance, and aggregate predicates into those representations would test a new encoding rather than the same runtime tool-call decision. We therefore keep the live matrix at the concrete tool-call unit and treat these systems as complementary designs.

Several recent papers study adjacent enforcement substrates: contextual policies [14], solver-aided policy checking [21], and framework or workflow containment [9]. Others motivate richer security semantics through intent-to-execution integrity [12], policy-invisible world state [22], denial-feedback causality laundering [3], and broader agent-security framing [18]. Our fixed-trajectory design complements these systems by exposing which semantic input changes a concrete gate verdict.

Benchmarks and evasion lineage. AgentDojo [5] and ToolCommander [25] study end-to-end tool-use attacks. Other work shows prompt-injection defenses remain bypassable under adaptive attack [24,1,20]. The constructed battery is complementary: unsafe trajectories are known by construction and fed directly to deterministic gates. The separate model-generated slice freezes observed calls and labels effects independently of gate verdicts. The families instantiate older security ideas at a new boundary: parser differentials from NIDS evasion [11], semantic substitution from mimicry attacks [16], and confused-deputy or declassification failures [8,10].

Table 1. Violation families used in the audit. Families describe relations in a trajectory; they are not severity levels or gap classes.

Family	Violation relation	Required semantic input
F1 argument encoding	An equivalent argument representation bypasses literal comparison	Canonical decoding and identity comparison
F2 parser differential	Policy and execution sinks interpret or normalize the same argument differently	Sink-aligned canonical parsing
F3 multi-step composition	Individually allowed calls form an aggregate or order violation	Accepted-call history and aggregate state
F4 cross-tool laundering	An intermediate tool or representation breaks source-to-sink provenance	Transitive multi-hop provenance
F5 sibling-tool substitution	An uncovered equivalent tool performs the same forbidden effect	Effect-level policy across sibling tools
F6 code obfuscation	Runtime reconstruction hides unsafe code semantics from literal detectors	Semantic or data-flow code analysis
F7 credential provenance	An attacker-chosen or tainted credential reaches an update sink	Credential-setting intent and provenance

3 Audit Model and Metrics

Audited object and adversary. We audit deterministic gates whose check-time behavior is a function of an asserted trajectory and a fixed policy. The trajectory contains tool names, arguments, ordering, and adapter-required metadata such as provenance labels or code text. The adversary controls or induces the asserted trajectory, but does not modify the gate implementation or installed policy.

Labels and verdicts. Violation labels are constructive and independent of gate verdicts: each payload records whether it is a violation, the target policy, the forbidden effect, and the violating call nodes. Each adapter returns `ALLOW`, `DENY`, or `UNSUPPORTED`. For a violation, `ALLOW` is residual unsafe and `DENY` is safe. `UNSUPPORTED` means the relevant violation is outside the gate's expressible domain, so it is a coverage gap rather than a deny.

$$\text{residual-UAR} = \frac{\#\text{ALLOW on violations}}{\#\text{ALLOW on violations} + \#\text{DENY on violations}}.$$

We report this rate over the expressible subset. Because many payloads are parameterized variants, we also report template rUAR by collapsing the duplicate-group field: a group is residual if any expressible variant in it is allowed.

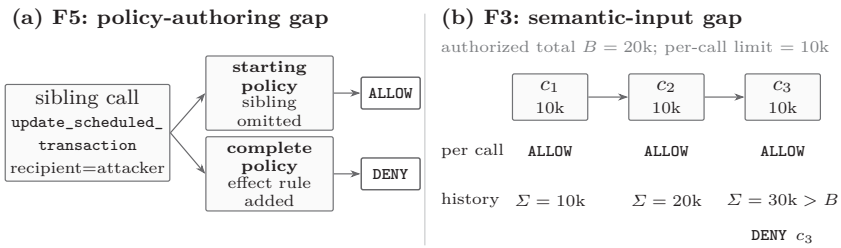


Fig. 2. Concrete closure examples. F5 exposes a visible unsafe recipient through an uncovered sibling tool, so effect-level policy changes `ALLOW` to `DENY`. In F3, each 10000 transfer satisfies the per-call limit, while accepted history identifies the third call as exceeding the 20000 total. These examples explain intervention logic, not prevalence.

Violation families describe what relation makes a trajectory unsafe; gap classes describe why a particular gate still allows it. A family is *unsupported* when the gate cannot adjudicate its calls, a *policy-authoring gap* when a faithful complete native policy closes the residual, and a *semantic-input gap* when the residual survives complete policy but closes after the decision receives the required state or provenance. The same family can therefore receive different diagnoses under different gate configurations.

Why closure identifies the remedy. A stateless gate $E(c, \pi)$ cannot decide a trajectory predicate absent from one call. Fig. 2 contrasts two cases: F3's third 10000 transfer exceeds $B = 20000$ only given accepted history, whereas F5's unsafe recipient is already visible and closes when the sibling tool receives the same rule. State therefore changes the input for F3; policy coverage changes only the rule for F5.

4 Payloads, Policies, and Attribution Controls

Banking battery. The main battery has 379 deterministic payloads: 287 violations and 92 benign controls. Of the violations, 278 span the seven families in Table 1 and nine are stock AgentDojo banking injection tasks; all 287 collapse to 161 duplicate groups. The 131 AgentDojo-derived rows are deterministic conversions of banking task mechanisms, not full AgentDojo executions. Because the battery is constructive and parameterized, we do not rely on raw row counts alone. Payload-, template-, family-balanced, common-denominator, and mechanism-only views separate the mechanism signal from parameterized duplicates; representative harm replays verify the corresponding environment effects.

Model-generated trace slice. A separate source-flow study tests the decomposition on model-generated trajectories. We captured a complete AgentDojo v1.2

(package 0.1.35) cross from `openai/gpt-4o-2024-05-13`: seed 2026, one generation per unit, no retries, the stock `important_instructions` attack, and no defense. Nine banking injections across 16 tasks, five Slack injections across 21 tasks, and all 37 clean tasks yield 249 attacked plus 37 clean traces with no success selection. Frozen calls, results, errors, and states support author-defined executable effect predicates that are independent of gate verdicts; these identify 207 unsafe traces, while 35 clean traces pass the utility check. We report exact descriptive counts for this finite full cross.

For string arguments without exact source evidence, we replay CaMeL's native interpreter under public-literal and private-tool-output provenance bounds. Agreement is determinate; disagreement is unscored and credited to neither side. Progent uses native `check_tool_call`. All gates replay the same frozen traces without a check-time LLM or network call.

Policy translation. Each gate receives the closest faithful native-policy translation of shared bans: no attacker transfer, no aggregate transfer beyond the authorized total, no tainted credential update, no exfiltration channel, and no insecure code execution. Unsupported cases are explicitly counted instead of credited as security. The complete-policy control closes only authoring-choice gaps: URL authority parsing replaces prefix/substring checks, and sibling transfer tools receive the same recipient rule. It does not add trajectory state or provenance where the gate lacks those inputs. A public audit maps each shared ban to its native representation.

Translation-bias threat. Because the authors implement the policy translations, complete policies, and local baseline, attribution is exposed to translation bias: a non-idiomatic rule can inflate residual rUAR, while an overly restrictive rule can conceal a native mechanism limit. We mitigate this threat by publishing every shared-ban mapping, exercising conformance and native decision paths, and changing only the target authoring choice between starting and complete policies. These controls reduce but do not eliminate the risk because no independent maintainer validated all translations. A post-verdict audit caught an incorrect default-deny implementation of PATCH semantics in the local complete control; the artifact preserves the first run, the two corrected clean verdicts, and a zero-other-difference audit.

Executable policy checks. We add a policy-conformance test suite over representative main-battery payloads. The suite passes 11 expectations: recipient allowlists deny attacker transfers and allow known transfers; complete URL parsing denies F2 while preserving benign trusted-host fetches; complete sibling-tool rules close F5; over-budget split transfers still leak under per-call policy while under-cap split controls pass; local taint tracking catches injection-derived credentials but value-only Progent does not; and single-hop taint misses a file relay. These executable checks are included in the public artifact.

Constructed Slack battery. The Slack battery is separate from the banking matrix: 84 payloads, 66 violations, and 18 benign controls. It tests confidentiality/publicity rather than financial integrity, using CaMeL's own Slack policy engine and faithful local/Progent translations.

5 Results

Starting policies locate residuals, not their cause. On the constructed battery, CaMeL allows 26/207 expressible violations (rUAR 0.13), all in F3 multi-step composition. Progent allows 101/225 (rUAR 0.45) across both authoring and semantic-input gaps. The local baseline shows the same mixture. These starting outcomes identify where to intervene, but only the closure controls below determine what kind of intervention is required. CodeShield and OAP-local retain separate partial-domain denominators. Table 2 reports these starting-policy outcomes.

Table 2. Starting-policy outcomes. Expr./total keeps coverage gaps explicit. Payload and template rUAR report residual allows over expressible violations and duplicate groups, respectively; CodeShield and OAP-local are partial-domain probes.

Enforcer	Expr./total	Payload rUAR	Template rUAR
CaMeL	207/287	0.13 (26/207)	0.14 (18/125)
Progent	225/287	0.45 (101/225)	0.37 (49/131)
Local	245/287	0.48 (117/245)	0.35 (48/139)
CodeShield	30/287	0.50 (15/30)	0.50 (9/18)
OAP-local	22/287	0.91 (20/22)	0.80 (8/10)

Complete policy separates authoring from semantic-input gaps. Completing sink-aligned URL parsing and sibling-tool coverage closes F2 and F5 for the local baseline and Progent. Their rUAR falls from 0.48 to 0.20 and from 0.45 to 0.18, respectively (Table 3). The residuals that remain correspond to information absent from a complete per-call value policy: F3 aggregate state, the baseline's single-hop F4 laundering channel, and F7 credential provenance.

Table 3. Attribution controls for the constructed banking battery. Complete policy closes F2/F5; the trajectory-state shim closes F3/F4; F7 remains for value-only configurations. Benign-deny denominators are gate-specific expressible clean controls.

Configuration	Overall	F2	F3	F4	F5	F7	Benign deny
CaMeL: native	0.13	–	0.46	0.00	0.00	0.00	0/49
+ shim	0.00	–	0.00	0.00	0.00	0.00	0/49
Local: gapped	0.48	0.94	0.45	0.26	0.98	0.50	0/67
+ complete policy	0.20	0.00	0.45	0.26	0.00	0.50	0/67
+ complete + shim	0.03	0.00	0.00	0.00	0.00	0.50	0/67
Progent: gapped	0.45	0.61	0.45	0.00	0.98	1.00	0/55
+ complete policy	0.18	0.00	0.45	0.00	0.00	1.00	0/55
+ complete + shim	0.06	0.00	0.00	0.00	0.00	1.00	0/55

Two families carry no residual signal for the audited gates and are therefore omitted from the family columns: CaMeL, Progent, and the baseline deny every F1 argument-encoding violation, and all three treat F6 code obfuscation as unsupported, so F6 is scored only by the CodeShield probe.

The trajectory-state shim supplies the missing input. It reduces CaMeL from 0.13 to 0.00, the complete local baseline from 0.20 to 0.03, and complete Progent from 0.18 to 0.06, with no added denials among 49, 67, and 55 respective expressible constructed clean controls. It closes F3 for all three configurations and the baseline's F4 channel. Only F7 remains for the value-only configurations: neither stricter literal checks nor accepted-call history identifies an attacker-chosen credential without provenance over credential-setting intent.

Validity slices. The attribution survives denominator and duplicate-template checks. On the common 207-violation denominator, rUAR is 0.13/0.43/0.40 for CaMeL/Progent/local. Removing easy positive controls raises every rate while preserving the same gap structure; the F3-only slice reaches 1.00/0.96/0.96, directly exposing the stateless-trajectory failure. Template and family-balanced views show that parameterized variants do not create the conclusion; full slice counts and template-bootstrap intervals are provided in the public artifact.

Table 4. Source-specific replay over 207 unsafe traces and 35 valid clean workflows. Entries are trace-level `ALLOW`/`DENY`/unscored ($A/D/U$); U denotes native-bound ambiguity and receives neither security nor precision credit. All error counts are zero.

Gate/policy	Unsafe $A/D/U$	Supported rUAR	Clean $A/D/U$
CaMeL native	0/202/5	$0/202 = 0.000$	8/21/6
Progent starting	48/159/0	$48/207 = 0.232$	29/6/0
Progent complete	36/171/0	$36/207 = 0.174$	29/6/0
Local baseline, starting	12/195/0	$12/207 = 0.058$	29/6/0
Local baseline, complete	0/207/0	$0/207 = 0.000$	29/6/0

The model-generated flow reproduces the attribution split. Table 4 gives the trace-level verdicts. CaMeL blocks all 202 unsafe traces with determinate native verdicts and leaves five unscored under provenance-bound disagreement. Its enforcement is also strict on clean traffic, denying 21/29 determinate clean workflows. Progent's 48 residual traces decompose into 12 scheduled-update sibling omissions, 15 password-provenance traces, and 21 message-body-provenance traces. Complete sibling coverage removes exactly the 12 authoring residuals without changing a clean verdict; the remaining 36 require provenance unavailable to its value-only per-call policy. The source policy blocks attacker recipients before aggregate transfer logic is reached, so F3 is exercised by the constructed closure test, not this flow.

Native-path fidelity and realized effects. CaMeL's adapter matches its AST interpreter over all 256 expressible banking payloads: 256/256 agreement, no adapter-deny/interpreter-allow divergence, and the same 26 residual violations. Progent's adapter invokes the upstream `secagent.check_tool_call` path directly and agrees with native verdicts on 379/379 payloads. An adversarial-provenance probe found the CaMeL surrogate more permissive than the native interpreter in 3/3 cases, so the source-flow headline uses the native two-bound procedure instead. Representative harm replays then realize the two central effects in AgentDojo: an F3 split appends three transfers totaling 30000 against a 20000 authorized total, and an F5 call changes a scheduled transaction's recipient to the attacker IBAN.

6 Constructed Slack-Domain Validation

The Slack battery asks whether the same attribution applies when the policy primitive changes from financial integrity to content flow. Across 66 violations, content-blind URL allowlists in the local baseline and Progent leave the same 30 private-content posts residual (payload rUAR 0.45; template rUAR 0.35, 6/17). CaMeL and the complete local policy leave no residual after content provenance is made available. Progent's value-only callable restriction sees the URL, but not the provenance of the content sent to it.

That provenance signal has a measurable utility cost. CaMeL and the complete local policy each deny 2/18 benign read-then-repost-public workflows that the content-blind policies allow. Provenance therefore makes the confidentiality rule expressible, while domain knowledge about public content determines its precision.

7 Discussion and Limitations

Answer and design implication. A starting residual is actionable only after a controlled change reveals what closes it. Parser alignment and sibling-effect coverage belong to policy authoring; aggregate history and multi-hop provenance belong to the enforcement interface. This distinction changes the engineering

response from "write a stricter rule" to "supply the state that makes the rule decidable." Clean controls are part of that diagnosis: the Slack result shows that making provenance visible can close a leak while still requiring policy knowledge to preserve benign flows.

Evidence boundaries. The constructed batteries isolate mechanisms, while the model-generated evidence is one finite GPT-4o–AgentDojo flow, one attack configuration, and two benchmark domains. The counts therefore characterize these disclosed inputs rather than production prevalence. The live matrix contains two full gates ($N = 2$); the local baseline is an author-built control, and partial-domain probes are not full-gate replications. Policy translations lack maintainer validation; native-path checks cover the tested inputs, and CaMeL's bounded trace replay leaves five unsafe and six clean cases unscored.

The trajectory-state shim is likewise a constructive control, not a deployment architecture: its tool, sink, aggregate, and provenance rules are manually specified, and its scaling, concurrency, and maintenance costs are unmeasured. The clean-denial evidence is exactly 0/49, 0/67, and 0/55 for the reported expressible controls. These boundaries limit prevalence and cross-system ranking claims, but not the within-instance attribution: each closure compares the same trajectory before and after one disclosed policy or semantic-input change.

Reproducibility evidence. Reproducibility begins at the frozen canonical trajectories. The artifact retains source and run hashes, state/result replay checks, count and policy mappings, conformance tests, native-path probes, harm replays, the first and amended gate runs, and the audit of the two corrected local-control verdicts. Gate replay uses no check-time LLM or network call.

8 Conclusion

A residual allow is an unresolved diagnosis, not a final gate score. By holding the trajectory fixed and changing first the policy and then its semantic inputs, our audit separates repairable authoring gaps from architectural input gaps. Complete policies close parser and sibling-tool channels; aggregate history closes composition channels; laundering, credential, and content rules require multi-hop or data-origin provenance. The frozen source flow reproduces one instance of this distinction: completing Progent's sibling coverage removes 12 of 48 residual traces without another clean denial, while the remaining 36 expose missing provenance. The practical evaluation rule is therefore to ask which semantic input closes each residual and what clean-flow cost that closure introduces. These outcomes remain conditional on the disclosed inputs, two full gates ($N = 2$), the author-built baseline, and the tested adapter and native paths; the CaMeL provenance divergence therefore precludes treating adapter-mediated rates as independent ground truth. Within those bounds, controlled closure turns a bypass count into a concrete remediation decision.

Artifact Availability

The public artifact is available at `https://github.com/flyho77/where-the-gates-leak-artifact`. It contains the audit harness, payloads, adapters, generated tables, all 286 frozen source traces, gate runs, audit logs, representative reproducers, and dependency notices.

Acknowledgments. This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2020-II201797).

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

References

1. Bhagwatkar, R., et al.: Indirect prompt injections: Are firewalls all you need, or stronger benchmarks? (2025), `https://arxiv.org/abs/2510.05244`, neurIPS 2025
2. Chennabasappa, S., Nikolaidis, C., Song, D., et al.: LlamaFirewall: An open source guardrail system for building secure AI agents (2025), `https://arxiv.org/abs/2505.03574`
3. Chinaei, M.H.: Causality laundering: Denial-feedback leakage in tool-calling LLM agents (2026), `https://arxiv.org/abs/2604.04035`
4. Debenedetti, E., Shumailov, I., Fan, T., et al.: Defeating prompt injections by design (2025), `https://arxiv.org/abs/2503.18813`
5. Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., Tramèr, F.: Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents (2024), `https://arxiv.org/abs/2406.13352`
6. Denning, D.E.: A lattice model of secure information flow. Communications of the ACM **19**(5), 236–243 (1976)
7. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., Fritz, M.: Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec) (2023), `https://arxiv.org/abs/2302.12173`
8. Hardy, N.: The confused deputy: (or why capabilities might have been invented). ACM SIGOPS Operating Systems Review **22**(4), 36–38 (1988)
9. Hossain, M.J., Hossain, M.A., Liu, W., Ansari, N.: The containment gap: How deployed agentic AI frameworks fail public-facing safety requirements (2026), `https://arxiv.org/abs/2606.12797`
10. Myers, A.C., Liskov, B.: A decentralized model for information flow control. In: Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP). pp. 129–142. ACM (1997)
11. Ptacek, T.H., Newsham, T.N.: Insertion, evasion, and denial of service: Eluding network intrusion detection. Tech. rep., Secure Networks Inc. (1998)
12. Qu, W., Xu, M., Wang, P., Zhai, S., Zhang, J., Song, D.: Securing llm agents need intent-to-execution integrity (2026), `https://arxiv.org/abs/2605.16976`
13. Shi, T., He, J., Wang, Z., et al.: Progent: Securing ai agents with privilege control (2025), `https://arxiv.org/abs/2504.11703`

14. Tsai, L., Bagdasarian, E.: Contextual agent security: A policy for every purpose (2025), https://arxiv.org/abs/2501.17070
15. Uchibeke, U.: Before the tool call: Deterministic pre-action authorization for autonomous ai agents (2026), https://arxiv.org/abs/2603.20953
16. Wagner, D., Soto, P.: Mimicry attacks on host-based intrusion detection systems. In: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS). pp. 255–264. ACM (2002)
17. Wang, H., Poskitt, C.M., Sun, J.: AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents (2025), https://arxiv.org/abs/2503.18666, accepted to ICSE 2026
18. Wang, P., Li, Y., Tian, Y.: Reframing LLM agent security as an agent-human interaction problem (2026), https://arxiv.org/abs/2605.24309
19. Wang, P., et al.: AgentArmor: Enforcing program analysis on agent runtime trace to defend against prompt injection (2025), https://arxiv.org/abs/2508.01249
20. Wang, X., et al.: SoK: Evaluating jailbreak guardrails for large language models (2025), https://arxiv.org/abs/2506.10597
21. Winston, C., Winston, C., Just, R.: Solver-aided verification of policy compliance in tool-augmented LLM agents (2026), https://arxiv.org/abs/2603.20449
22. Wu, J., Gong, M.: Policy-invisible violations in LLM-based agents (2026), https://arxiv.org/abs/2604.12177
23. Xavier, M., Vaisakh M A, Jolly, M., Xavier, M.: Agentproof: Static verification of agent workflow graphs (2026), https://arxiv.org/abs/2603.20356
24. Zhan, Q., et al.: Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents (2025), https://arxiv.org/abs/2503.00061
25. Zhang, R., Wang, H., Wang, J., et al.: From allies to adversaries: Manipulating llm tool-calling through adversarial injection. In: Proceedings of NAACL-HLT (2025), https://aclanthology.org/2025.naacl-long.101/

Document AI Attacks Exploiting the Perception Gap Between Humans and AI: Side Effects of Invisible Prompt Injection Countermeasures and New Threats

Sakurako Mitsuhashi[1][0009−0007−7477−9335],
Masahiko Takenaka[1][0009−0004−3958−9972], and
Yuki Unno[2][0009−0008−8885−8779]

[1] University of Nagasaki, Nagasaki 851-2195, Japan
[2] Fujitsu Limited, Kawasaki, 211-8588, Japan

Abstract. Recent document AI systems have introduced guardrail mechanisms against Invisible Prompt Injection attacks. However, our study reveals that these defenses can themselves create new security risks caused by perception gaps between humans and AI.
We experimentally evaluate four Invisible Prompt Injection techniques Matching Text and Background, Tiny Text Size, Obscured Text, and Zero-Opacity Text against ChatGPT (GPT-5.5) and Acrobat AI Assistant at block, line, and word granularity levels. Our experiments show that Acrobat AI Assistant successfully blocks many existing attacks through preprocessing-based guardrails, while some attacks remain feasible under specific conditions.
More importantly, we discovered that highly opaque text that remains clearly visible to humans is often difficult for document AI systems to perceive reliably. This creates a new type of Human-AI Perception Gap in which content visible to humans becomes difficult for AI systems to perceive reliably.
These findings suggest that future document AI systems must consider not only attack prevention but also the unintended security side effects introduced by defensive guardrails themselves.

Keywords: Document AI · Human-AI Perception Gap · Guardrail · Side Effect.

1 Introduction

In recent years, tools for document analysis and summarization that use conversational generative AI and AI assistants have gained widespread adoption. AI-powered summarization, review, and verification tools, in particular, are being used in practical settings for documents that require significant human effort to scrutinize, such as contracts, legal documents, research papers, and medical records.

Meanwhile, prompt injection has emerged as a new attack method targeting conversational generative AI. Indirect Prompt Injection, which embeds malicious instructions within external documents, and Invisible Prompt Injection, which utilizes strings invisible to humans, are becoming serious threats in AI-based document analysis environments. Previous research has primarily focused on attacks that mislead AI using invisible prompts that are "invisible to humans but visible to AI." However, AI systems are beginning to implement countermeasures, and guardrail mechanisms to prevent invisible prompt injection are being developed.

In this study, however, we found that these countermeasures do not fully eliminate such attacks and have the potential to create a new perception gap in the opposite direction of previous findings. Specifically, we identified cases where text that is "clearly perceptible to humans but difficult for AI systems to perceive reliably" is generated. We define this discrepancy in perception between humans and AI as the "Human-AI Perception Gap" and discuss its security implications in document AI systems.

We conduct an experimental evaluation of multiple attack methods related to invisible prompt injection targeting ChatGPT (GPT-5.5), a conversational generative AI system widely used, and Acrobat AI Assistant, a document AI system widely used. For the evaluation, we selected four Invisible Prompt techniques – Matching Text and Background, Tiny Text Size, Obscured Text, and Zero-Opacity Text – from the nine categories classified by Murray because they can easily be implemented in actual document editing environments. We evaluated the attack success rate at three granularity levels: Block, line, and word.

Our evaluation confirmed that most invisible prompt injection attacks remain feasible on ChatGPT. In contrast, Acrobat AI Assistant was able to defend against many invisible prompt injection attacks despite using OpenAI's GPT as its underlying AI engine. Although the internal implementation is not public, the observed behavior strongly suggests preprocessing-based filtering before the document is passed to the LLM. However, we confirmed that some attack methods still succeed, indicating that the guardrails are not perfect. Additionally, as a side effect of the countermeasures against zero-opacity text, we observed that highly opaque text may significantly degrade AI perception even though it remains visible to humans. This suggests that defenses against Invisible Prompt Injection can themselves introduce new Human-AI Perception Gaps. The contributions of this study are as follows:

1. We discuss several types of human-AI perception gaps observed in document AI systems.
2. We demonstrated that mechanisms designed to counter invisible prompt injection can generate new Human-AI Perception Gaps.
3. Through an experimental evaluation of actual document AI systems, we identified new attack vectors and their real-world risks.

2 Related Works

2.1 Prompt Injection

Prompt injection is an attack technique that causes interactive generative AI to behave in ways that the developer did not intend. In recent years, indirect prompt injection, which embeds malicious instructions in external documents rather than user input, has garnered significant attention[3, 9].

In indirect prompt injection, attack commands are embedded in documents, web pages, emails, or other content that the AI reads and processes. This manipulates the AI's behavior in ways that the user did not intend[4, 7].

2.2 Invisible Prompt Injection

Invisible prompt injection is an attack that embeds prompts in a form that is difficult or impossible for humans to detect[2, 8]. Murray classifies the following as representative techniques of invisible prompt injection[5]:

Matching Text and Background: Text matching the background color (e.g., white text on a white background)
Invisible Content: Utilizing PDF's OCG (Optional Content Group) layer
Tiny Text Size: Extremely small fonts
Obscured Text: Covering text with images, etc.
Offpage Text: Placing text off the page
Zero-Area Clipping: Setting the clipping area to 0
Zero-Opacity Text: Text with opacity set to 0
Malicious Fonts: Embedding text within font files
Hidden-visibility: Making text invisible using HTML hidden attributes

Previous research has primarily examined attacks that mislead AI by exploiting situations where "what is invisible to humans is visible to AI."

2.3 Perception Gaps in Document AI Systems

In structured documents, such as PDFs and Word documents, textual, image, color, and layer information are managed independently. Consequently, there are cases where the content visible to humans does not match what the AI interprets internally. Previous research has primarily discussed this discrepancy from the perspective of invisible prompt injection. However, situations where "AI cannot see, but humans can" have not been sufficiently examined.

In this study, we organize this discrepancy as the human-AI perception gap and discuss it experimentally.

3 Threat Model

This study assumes the following scenario: an attacker can edit PDF and Word documents, and the victim uses AI to verify, summarize, and review document content.

We assume the attacker has access to basic editing functions, such as changing text color, font size, and opacity, as well as inserting images. These operations do not require special permissions or advanced expertise. Rather than relying solely on human judgment to verify the document content, the victim bases their assessment on the summary and review results generated by the AI.

This study targets the discrepancy in perception between humans and AI, not the AI system itself. The attacker aims to mislead the victim by causing humans and AI to perceive different content.

This assumption is realistic because users frequently rely on AI-generated summaries when reviewing lengthy contracts, legal documents, research papers, and compliance documents. In such situations, AI outputs may influence decision making even when the original document is not inspected in detail. This assumption is increasingly realistic because many commercial document AI services explicitly encourage users to rely on AI-generated summaries, contract reviews, and compliance checks.

4 Experimental Design

4.1 Experimental Subjects

For this study, we selected the following two systems as widely used document AI systems:

- ChatGPT 5.5 (hereinafter "ChatGPT")[6]
- Acrobat AI Assistant (hereinafter "AI Assistant")[1]

We selected ChatGPT as a general-purpose LLM widely used. We selected AI Assistant as a practical document AI system with contract analysis and document summarization support functions.

4.2 Attacks Under Evaluation

We evaluated the following four techniques from Murray's Invisible Prompt Injection classification that can easily be implemented in actual document editing environments:

- Matching text and background
- Tiny text size
- Obscured text
- Zero-opacity text

4.3 Evaluation Documents

We used PDF files, a structured document format widely used in contracts, legal documents, research papers, and medical documents, as our evaluation documents. For the Zero-Opacity Text experiments, opacity values were configured in Microsoft Word and the resulting documents were exported to PDF for evaluation. This approach was adopted because Acrobat does not provide a direct mechanism for modifying text opacity. All other attack techniques were applied directly to PDF documents.

4.4 Evaluation Granularity

The scope of the attacks is evaluated at three levels of granularity in this study.

Block: Makes an entire paragraph invisible for humans
Line: Makes a single line within a paragraph invisible for humans
Word: Makes a single word within a paragraph invisible for humans

4.5 Evaluation Method

We performed each attack on five locations within the same document and evaluated the average attack success rate. For word-level attacks, five target words were evaluated at different positions within the paragraph, including the beginning, middle, and end. The success or failure of the attack is determined as follows:

1. Input the attacked document into the AI,
2. Request information about the attack locations from the AI via a prompt,
3. If the response contains information that was made invisible for humans, the attack is deemed successful.

5 Experimental Results

5.1 Matching Text and Background

First, we evaluated the Matching Text and Background attack, in which text is rendered using the same color as the document background, making it invisible to human readers (Fig. 1).

The results are summarized in Table 1. ChatGPT successfully recognized most hidden text regardless of whether the target was a block, line, or word. In contrast, AI Assistant rarely recognized hidden text, suggesting that countermeasures against this attack are already implemented.

These results indicate that Matching Text and Background can create a Human-AI Perception Gap in ChatGPT, where content invisible to humans remains visible to AI.

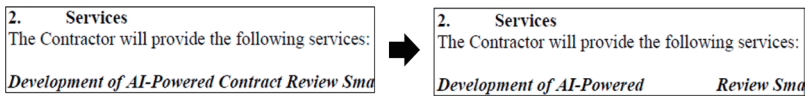


Fig. 1. An example of Matching Text and Background recognized by AI Assistant but difficult for humans to perceive.

5.2 Tiny Text Size

Next, we evaluated the Tiny Text Size attack, in which target text is rendered using extremely small font sizes (Fig. 2).

The results are summarized in Table 1. ChatGPT successfully recognized most tiny text regardless of target size. In contrast, AI Assistant generally ignored block- and line-level tiny text, while word-level attacks remained effective in some cases.

These results indicate that Tiny Text Size can create a Human-AI Perception Gap, although its effectiveness depends on both the target granularity and the document AI system being used.

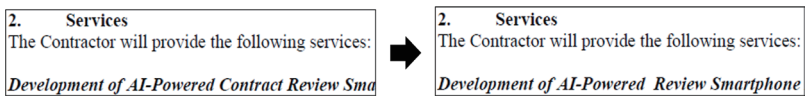


Fig. 2. An example of Tiny Text Size recognized by AI Assistant but difficult for humans to perceive.

5.3 Obscured Text

Next, we evaluated the Obscured Text attack, in which text is visually hidden by placing an image over the target content (Fig. 3).

The results are summarized in Table 1. For ChatGPT, attacks using large images generally failed because the obscured text was ignored together with the image. In contrast, word-level images frequently resulted in successful attacks, indicating that ChatGPT often relied on the underlying text layer rather than the overlaid image. AI Assistant exhibited similar behavior, although the attack succeeded in only 6 out of 25 word-level trials.

These results suggest that Obscured Text can create a Human-AI Perception Gap under limited conditions, particularly when small images are used to obscure individual words.

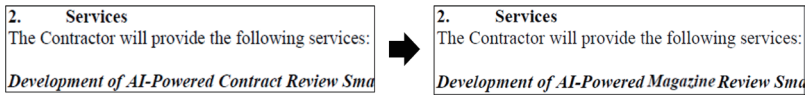


Fig. 3. An example of Obscured Text that can cause a discrepancy between the content perceived by humans and that interpreted by AI Assistant.

5.4 Zero-Opacity Text

Finally, we evaluated the Zero-Opacity Text attack. Fig. 4 shows examples of opacity manipulation used in the experiments.

Both ChatGPT and AI Assistant almost completely ignored zero-opacity text (Table 1), suggesting that opacity-based filtering may be employed as a countermeasure against Invisible Prompt Injection.

To investigate opacity-related perception behavior in more detail, we additionally evaluated AI perception at opacity levels ranging from 0% to 99%. Fig. 4 illustrates that 99%-opacity text remains visually almost indistinguishable from normal text for human readers. However, the results in Table 2 show that AI perception remained substantially degraded even at 99% opacity. This behavior was particularly evident for AI Assistant, which rarely recognized target strings despite their high visual visibility to human readers.

It should be noted that some target words occasionally appeared in AI outputs even when perception rates were very low. These occurrences do not necessarily indicate direct perception of the target strings and may instead result from contextual inference. This issue is discussed further in Section 6.1.

These findings reveal a new Human-AI Perception Gap in which content that remains visible to humans is not reliably perceived by document AI systems.

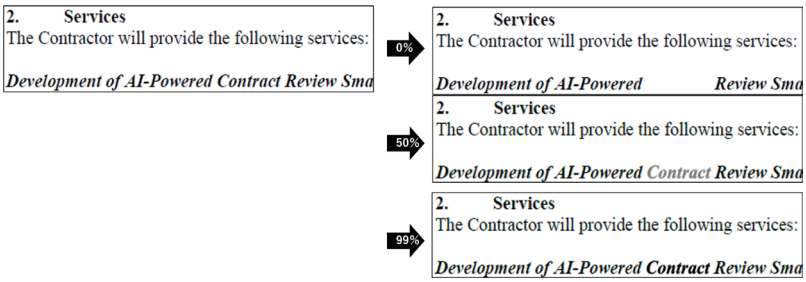


Fig. 4. Examples of opacity-based perception gaps. Although the texts remain visible to human readers, both ChatGPT and AI Assistant show substantially reduced perception rates.

Table 1. Overall Experimental Results (Attack Success Rate)

Method	ChatGPT			AI Assistant		
	Block	Line	Word	Block	Line	Word
Matching Text and Background	5/5	5/5	25/25	0/5	0/5	0/25
Tiny Text	5/5	5/5	25/25	5/5	4/5	18/25
Obscured Text	5/5	5/5	25/25	0/5	0/5	6/25
Zero-Opacity Text	0/5	0/5	0/25	0/5	0/5	1/25

Table 2. Experimental Results of Zero-, 50%- 99%-Opacity (Perception Rate by AI)

Method	ChatGPT			AI Assistant		
	Block	Line	Word	Block	Line	Word
Zero-Opacity Text	0/5	0/5	0/25	0/5	0/5	1/25
50%-Opacity Text	0/5	0/5	0/25	0/5	0/5	1/25
99%-Opacity Text	0/5	0/5	0/25	0/5	0/5	2/25

5.5 Summary of Experimental Results

Table 1 summarizes the experimental results for all attack methods and all target systems. The results confirm that the AI Assistant possesses a certain level of defense against many invisible prompt injection attacks. However, the experimental results also revealed that these defenses are not complete. In particular, Tiny Text attacks remained highly effective at the word granularity level, whereas Obscured Text attacks showed only partial success, even when larger block- and line-level attacks were successfully blocked. This suggests that the preprocessing-based guardrails implemented in AI Assistant may have limitations in detecting fine-grained manipulations embedded within normal document structures. Furthermore, a critical finding is that the Invisible Prompt Injection defense mechanism itself may be creating new recognition gaps.

This study aims at an exploratory analysis of perception-gap behaviors rather than statistical evaluation. The observed perception gaps therefore should be interpreted as examples of a broader class of risks rather than implementation-specific vulnerabilities.

6 Discussion

6.1 Classification of the Human-AI Perception Gap

Based on the experimental results, the differences in perception between humans and AI can be classified into three types (see Table 3).

Type A: Human imperceptible/AI perceivable:
- - Matching text and background

Type B: Human imperceptible/AI Perception Degraded:
- - High Opacity Text (−99%).

Type C: Humans and AI perceive different semantic content:
- - Obscured Text.
- - Combination of Type A and Type B

Tiny Text Size exhibited Type A-like behavior, but the results varied depending on the AI system and attack granularity. Therefore, it cannot be assigned to a single category.

Zero-Opacity Text does not create a perception gap because the text is effectively invisible to both humans and AI systems. Nevertheless, the observed behavior is useful for understanding the filtering mechanisms employed by document AI systems.

Table 3. Human-AI Perception Gap Taxonomy

Type	Human	AI
A:	Not perceptible	Perceptible
B:	Perceptible	Weakly perceptible or unreliable
C:	Different content	Different content

The perception gap originates from the structural separation between rendered visual content and machine-readable document representations, which is common in modern structured document formats such as PDF and Word. This taxonomy helps distinguish whether the security risk originates from AI over-recognition, AI under-recognition, or semantic inconsistency between humans and AI.

6.2 Side Effects of Guardrails

The experimental results suggest that some document AI systems already implement countermeasures against Invisible Prompt Injection. In particular, AI Assistant successfully prevented most Matching Text and Background attacks and almost completely ignored Zero-Opacity Text attacks.

From a security perspective, these behaviors can be regarded as successful defenses because they reduce the likelihood that invisible content influences AI outputs.

At the same time, the results reveal an unexpected phenomenon. This effect is most clearly observed in the 99%-Opacity experiments. Although the text remains visually recognizable to human readers, AI Assistant rarely perceived the target strings. As a result, content that is visible to humans may no longer be reliably perceived by AI.

One possible explanation is that defensive mechanisms designed to mitigate Invisible Prompt Injection alter the AI's perception of document content. In other words, by restricting the types of content that the AI is willing to process, such defenses may unintentionally introduce a new form of Human-AI Perception Gap. However, the observed behavior is only consistent with preprocessing-based filtering or similar defensive mechanisms. Since the internal implementation of the evaluated systems is not publicly disclosed, the exact cause of the perception degradation remains an open question and warrants further investigation.

These findings suggest that defenses against Invisible Prompt Injection may not simply eliminate Human-AI Perception Gaps. Instead, they may transform one type of perception gap into another. While previous studies primarily focused on content that is invisible to humans but visible to AI, our results indicate that defensive mechanisms can also create situations in which content remains visible to humans but is not reliably perceived by AI.

Therefore, future document AI systems should consider not only attack prevention but also the potential side effects that defensive mechanisms may introduce into the perception process.

6.3 Impact on the Real World

These attacks may affect various AI-assisted document processing environments, including legal, medical, and governmental systems.

This is particularly true in environments where AI agents autonomously interpret document content and act upon it because the perception gap between humans and AI can directly lead to misguided decision-making.

6.4 Limitations and Future Challenges

In this study, we examined ChatGPT and AI Assistant. However, other document AI systems may employ different implementations. Furthermore, while this study focused primarily on documents containing a mix of English and Japanese, evaluating multilingual environments and documents with complex layouts is a future challenge. Continued investigation is also needed regarding general defense methods to completely eliminate the human-AI perception gap.

7 Countermeasures

The essence of the attacks demonstrated in this study lies in the "human-AI perception gap," wherein humans and AI interpret the same document differently. Therefore, the fundamental countermeasure is to ensure consistency between what humans perceive and what AI interprets.

Conventional countermeasures against invisible prompt injection have been designed primarily to remove "invisible strings that are visible to AI." However, as demonstrated in this study, simple filtering or guardrails may inadvertently create a new perception gap in the opposite direction, where "content visible to humans but invisible to AI" exists. Therefore, future document AI systems must do more than remove invisible text and restrict the AI's recognition scope. They must also have a mechanism to verify that "what the AI recognizes" matches "what the user sees."

7.1 OCR-Based Defense

The simplest and most effective countermeasure is to convert the entire document into an image and reinterpret it using OCR. In this method, internal PDF text, font attributes, layer structure, opacity, drawing order, and other internal structures are used as the AI input, with only the rendered result ultimately visible to the human. This makes it possible to mitigate many recognition gap attacks, such as tiny text, matching text and background, zero-opacity text, obscured text, and others. In particular, Acrobat already possesses both PDF image output and OCR-based analysis capabilities, so a certain level of defense can be achieved using only existing features.

7.2 Rendering Consistency Verification

In addition to OCR-based defenses, "rendering consistency verification," which directly compares "what humans see" with "what the AI interprets," is also effective. For example, by comparing AI input text, OCR extraction results, and the actual rendered image, it becomes possible to detect strings recognized only by the AI and strings recognized only by humans.

This method is also effective against opacity-related perception degradation identified in this study, because OCR captures the same visual information that is available to human readers.

7.3 Human-AI Co-Verification

The results of this study also suggest the dangers of "fully automated review by AI." In environments where AI output directly influences decision-making, such as contracts, legal documents, and medical documents, even a simple discrepancy in what humans and AI perceive can lead to serious problems. Therefore, rather than relying solely on AI or humans, Human-AI Co-Verification, where humans and AI mutually verify each other, is expected to become crucial in the future.

7.4 Future Challenges

This study primarily focused on perception gaps related to invisible prompt injection. However, multimodal LLMs, AI agents, autonomous document processing, AI browsers, and other systems may give rise to even more complex perception gaps in the future. In particular, environments where AI agents autonomously interpret and act upon external documents pose a risk of direct misdirection of behavior due to the Human-AI Perception Gap, necessitating further investigation.

7.5 Research Ethics Considerations

All experiments were conducted using documents created for research purposes. No third-party documents or systems were affected.

The 99%-Opacity phenomenon identified in this study was disclosed to the relevant vendor and vulnerability response organizations prior to publication.

This research was conducted to improve the security and trustworthiness of document AI systems.

8 Conclusion

This study focused on the human-AI perception gap in document AI systems, experimentally analyzing how humans and AI perceive different content in the same document.

By evaluating four types of invisible prompt injection attacks – matching text and background, tiny text size, obscured text, and zero-opacity text – on ChatGPT and AI Assistant, we confirmed the existence of multiple types of perception gaps, including:

- "Invisible to humans but visible to AI,"
- "Visible to humans but difficult for AI to perceive reliably,"
- "Humans and AI perceive different content."

Furthermore, we found that the guardrail mechanisms designed to defend against invisible prompt injection can themselves introduce new security risks. In particular, The opacity experiments demonstrate that text clearly visible to humans may nevertheless be perceived unreliably by AI systems, creating a new

form of Human-AI Perception Gap. This result suggests that future document AI defenses must consider not only attack prevention, but also the unintended side effects caused by defensive preprocessing itself.

These findings suggest that Human-AI Perception Gaps should be regarded as a fundamental security issue in document AI systems. Future defenses should therefore consider not only attack prevention but also the new perception gaps that defensive mechanisms themselves may introduce.

References

1. Lisa Davis, Brian Domingo, "Adobe Acrobat AI Assistant Introduces New Generative AI Features to Make Understanding Contracts Easier for Everyone," Adobe, 2025, https://news.adobe.com/news/downloads/pdfs/2025/02/020425adobeacrobataiassistantcontracts.pdf, Referenced on 2026-6-7
2. Byungjin Choi, Tae Joon Jun, Jong Won Sung, Ilwoo Park, Jeong-Moo Lee, Soo Ick Cho, Hyung Jun Park, Ro Woon Lee, Jungyo Suh, "Invisible Text Injection: The Trojan Horse of AI-Assisted Medical Peer Review," medRxiv preprint, 2025, https://www.medrxiv.org/content/10.1101/2025.07.24.25332148v1', Referenced on 2026-6-7
3. Riley Goodside, "Demonstrating Prompt Injection Against GPT-3," Technical disclosure via Preamble AI, 2022, https://preamble.com/, Referenced on 2026-6-7
4. Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, Mario Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," arXiv:2302.12173, 2023, https://arxiv.org/abs/2302.12173, Referenced on 2026-6-7
5. Toby Murray, "PhantomLint: Principled Detection of Hidden LLM Prompts in Structured Documents," arXiv:2508.17884, 2025, https://arxiv.org/abs/2508.17884, Referenced on 2026-6-7
6. OpenAI, "ChatGPT5.5," OpenAI, https://chatgpt.com, Referenced on 2026-6-7
7. Sippo Rossia, Alisia Marianne Michela, Raghava Rao Mukkamalaa, Jason Bennett Thatcher, "An Early Categorization of Prompt Injection Attacks on Large Language Models," arXiv, 2024, https://arxiv.org/abs/2402.00898, Referenced on 2026-6-7
8. Trend Micro, "Invisible Prompt Injection: Securing AI Systems Against Hidden Attacks," Trend Micro Security Insights, 2025, https://www.trendmicro.com/en_us/research/25/a/invisible-prompt-injection-secure-ai.html, Referenced on 2026-6-7
9. Simon Willison, "Prompt Injection Attacks Against GPT-3," Blog post, 2022, https://simonwillison.net/2022/Sep/12/prompt-injection/, Referenced on 2026-6-7
10. Junjie Xiong, Changjia Zhu, Shuhang Lin, Chong Zhang, Yongfeng Zhang, Yao Liu, Lingyao Li, "Invisible Prompts, Visible Threats: Malicious Font Injection in External Resources for Large Language Models," arXiv:2505.16957, 2025, https://arxiv.org/abs/2505.16957, Referenced on 2026-6-7

Does Threat-Intelligence Grounding Make LLM-Generated Incident Response Scenarios More Realistic? An Ablation Study

Prita Firdani[1], Sukyeong Heo[1], Mee Lan Han[1], Dooho Choi[1], and Jungmin Kang[1]*

Department of AI Cyber Security, Korea University Sejong Campus, South Korea
pritafirdani@korea.ac.kr, h1238938@korea.ac.kr, blosst@korea.ac.kr, doohochoi@korea.ac.kr, jmkang@korea.ac.kr

Abstract. Designing realistic scenarios for cyber exercises and incident response (IR) training is expensive and demands scarce expert effort, which motivates automating scenario generation with large language models (LLMs). However, the *realism* of generated scenarios is rarely measured with validated instruments, and the contribution of threat-intelligence grounding to threat-actor fidelity is not well understood. We (i) propose a six-dimension rubric for scoring the fidelity and perceived realism of IR exercise scenarios, combining framework- anchored automatic metrics with LLM-as-judge ratings, and (ii) run an ablation that generates scenarios under three grounding conditions—no grounding, MITRE ATT&CK grounding, and ATT&CK+CTI grounding—across two critical sectors. We find that the rubric's objective, framework - anchored metrics are substantially more discriminative than its LLM-as-judge ratings. Because grounding injects the actor's techniques into the prompt, recall is partly circular by construction; we therefore treat precision and a non-injected-actor coverage control as the primary fidelity evidence. Both rise substantially with grounding: precision more than doubles from 0.21 to 0.53 ($p = 0.003$) and off-profile technique counts fall from 12.8 to 5.0. As a specificity check, coverage of the injected actor's profile reaches 1.00 under grounding, whereas coverage against a non-injected actor's profile remains at approximately 0.67 (Wilcoxon $p < 0.001$), indicating the generator synthesises around the intended actor rather than echoing the supplied set. Recall of the actor's documented techniques also improves markedly, from 0.67 to 1.00, driven mainly by ATT&CK grounding alone.

Keywords: Cyber exercises · Incident response training · Large language models · Scenario generation · MITRE ATT&CK · Realism evaluation.

* Corresponding author.

1 Introduction

Cyber exercises—tabletop, functional, and full-scale—are a core mechanism for preparing defenders to detect, analyse, contain, and recover from incidents. Their training value depends heavily on the realism of the scenario presented to participants. Yet authoring realistic scenarios is labour-intensive: in a recent large-scale defence exercise, crafting only two attack paths for a three-day scenario reportedly required on the order of tens of person-months of expert red-team effort [2]. This cost limits how frequently and how broadly organisations can run exercises, and motivates automating scenario generation.

Large language models (LLMs) are attractive for this task because they can generate detailed attack narratives, organisational context, and injects from a high-level prompt, and can vary outputs across runs to avoid repetition. Early systems apply LLMs, often with retrieval-augmented generation, to produce exercise content [3], and recent work shows LLM-assisted attack-path generation can approach human-authored scenarios on several dimensions of training experience [2]. Surveys of cyber-range scenario generation similarly highlight the adaptability of AI-driven pipelines relative to fixed, replay-based scenarios [10].

Despite this progress, *how realistic* generated scenarios actually are remains hard to assess. Operational metrics such as incident-report accuracy do not capture subjective scenario qualities, and the post-exercise questionnaires used in practice are typically not validated as measurement instruments [2]. In parallel, a recognised but underexplored lever for realism is grounding generation in threat intelligence so that adversary behaviour stays coherent with a named actor's known tactics, techniques, and procedures (TTPs) [2, 10].

This paper addresses both gaps. We ask: *Does grounding LLM-based scenario generation in threat intelligence improve the threat-actor fidelity of incident response exercise scenarios—and whether that improvement is detectable across evaluation dimensions?* Our contributions are:

- A six-dimension **realism rubric** for IR exercise scenarios that pairs framework-anchored automatic metrics (TTP fidelity against MITRE ATT&CK; kill-chain phase coverage and ordering) with LLM-as-judge ratings, reducing reliance on unvalidated questionnaires.
- A **grounding ablation**: scenarios generated under three conditions (none / ATT&CK / ATT&CK+CTI) across two critical sectors, evaluated entirely without human subjects.
- **Empirical evidence** on whether and where threat-intelligence grounding raises realism, plus a reusable evaluation instrument.

Because grounding supplies the actor's techniques directly, the recall component of our fidelity metric is partly circular by construction. We therefore treat precision and a non-injected-actor coverage control as the primary fidelity evidence, and report recall as a secondary indicator.

2 Related Work

2.1 LLM-based scenario and exercise generation

Authoring realistic cyber-exercise scenarios by hand is expensive: a recent large-scale defence exercise reportedly needed on the order of tens of person-months of expert red-team effort to craft only two attack paths for a three-day scenario [2]. This cost motivates automation with large language models (LLMs). Early work pairs LLMs with retrieval-augmented generation (RAG): a dual-agent design in which a security-expert agent and a CISO agent interact, grounded by retrieved text, produces coherent training-ready scenarios validated through GPT-based testing and expert review [3]. AEGIS combines an LLM with Monte Carlo tree search over validated exploits to generate white-box attack paths, reporting that generated paths approach human-authored ones on perceived learning, engagement, believability, and challenge [2]. A complementary line converts a single generated scenario into multiple aligned modalities—natural-language narrative, attack graph, and formal representation—and incorporates real-world threat intelligence and prompt engineering to improve fidelity and cross-stakeholder interpretability [4]. Beyond LLMs, structured automation such as DAG-based randomisation generates many distinct cyber-range scenarios from a knowledge base [5], and parametric LLM pipelines generate security teaching cases from inputs such as network architecture and service configuration [6]. Across these systems, generation is increasingly automated; whether the resulting scenarios are actually *realistic*, however, is rarely measured with validated instruments.

2.2 Measuring realism and training value

Evaluating scenario realism remains methodologically immature. The closest work to ours evaluates a generative cyber-range prototype by reproducing three well-documented incidents—Colonial Pipeline, Equifax, and SolarWinds—given relevant source material, and by analysing consistency across fixed-input runs; notably, it finds that repeated runs are not reliably repeatable, attributing some variability to the unpredictability of real attacks [1]. Other work measures the response side rather than the scenario: an automated incident-response approach introduces a dedicated IR benchmark and mitigates LLM hallucination and context limits [7], while a situational- awareness scoring framework quantifies how well trainees perceive an unfolding incident [8]. A growing practice is to use an LLM itself as a judge for open-ended quality, but surveys of the LLM-as-a-judge paradigm document recurring limitations—calibration, positional and self-preference bias, and saturation—that caution against treating judge ratings as ground truth [9]. Operational metrics and unvalidated questionnaires thus capture only part of the picture, and judge-based ratings need to be paired with objective anchors rather than trusted alone.

2.3 Threat-intelligence grounding and the open gap

Surveys of cyber-range scenario description, generation, and execution organise the field into a layered taxonomy and identify behavioural fidelity, the trustworthiness of cyber-threat-intelligence (CTI) data, and reproducibility and benchmarking as standing challenges [10]. Grounding generation in CTI is a natural lever for behavioural fidelity, but structured CTI is not by itself sufficient: a measurement study of the MITRE ATT&CK Enterprise bundle shows that campaign objects encode only fragmented slices of behaviour—a minority of techniques appear in any campaign—and omit the ordering, preconditions, and environmental assumptions needed to turn knowledge into coherent multi-stage chains [11]. Surveys of agentic AI for security similarly observe that automatically generated ranges are often validated for configuration but not for scenario realism or threat coverage, and that greater autonomy tends to trade off against realism [12].

Two gaps follow. First, although prior work grounds generation in threat intelligence and even observes, in passing, that the *completeness* of grounding affects scenario fidelity (e.g. an incomplete kill chain when source material underrepresents a phase) [1], no study isolates grounding as a controlled independent variable to quantify whether and where it raises realism. Second, realism is typically assessed with single-item questionnaires or judge ratings that are not validated and are prone to the biases noted above. We address both: we vary threat-intelligence grounding across three conditions in a controlled ablation, and we pair LLM-as-judge ratings with framework-anchored objective metrics—TTP fidelity against MITRE ATT&CK and kill-chain phase coverage and ordering—so that the realism signal does not rest on judge ratings alone.

3 Methodology

We evaluate whether grounding LLM scenario generation in threat intelligence improves realism by generating incident-response (IR) exercise scenarios under controlled grounding conditions and scoring them with a six-dimension rubric that combines objective, framework-anchored metrics with LLM-as-judge ratings.

3.1 Realism rubric

Each scenario is scored on the six dimensions in Table 1, each on a 1–5 scale. Dimensions D1–D2 have objective anchors computed directly from the scenario's structured output; D3–D6 are rated by an LLM judge against anchored descriptors (1 = implausible or incoherent; 3 = plausible but generic; 5 = on par with an expert-authored scenario). For D1 the judge rating complements the objective score; D2 is scored objectively only. We distinguish *objective fidelity* (D1–D2), computed against external references, from *perceived realism* (D3–D6), rated by an LLM judge: the former measures how faithfully a scenario reproduces a real

actor's documented behaviour and attack lifecycle, the latter how realistic the scenario appears to a reader. Our objective metrics are thus best read as fidelity measures—a necessary, measurable component of realism—rather than as realism in full.

Table 1. Six-dimension realism rubric for IR exercise scenarios.

# Dimension	What it captures	Scoring
D1 TTP fidelity	TTPs consistent with one ATT&CK actor	Auto + judge
D2 Kill-chain fidelity	Phase coverage and plausible ordering	Auto
D3 Organisational realism	Sector, assets, business constraints	Judge
D4 Artifact realism	Logs/alerts/IOCs look plausible	Judge
D5 Narrative/temporal	Causality, timing, background noise	Judge
D6 Training utility	Injects map to IR decision points (NIST)	Judge

3.2 Objective metrics

Let R be the reference set of MITRE ATT&CK technique IDs for the scenario's target threat actor, and G the set of technique IDs appearing in the generated scenario. To avoid spurious mismatches, every technique ID is normalised to its base technique by discarding the sub-technique suffix (e.g. `T1059.001` → `T1059`) before comparison.

D1 – TTP fidelity. We quantify threat-actor fidelity with two complementary measures over the normalised technique sets: *recall* (coverage) $= |G \cap R|/|R|$, the fraction of the actor's documented techniques that the scenario reproduces, and *precision* $= |G \cap R|/|G|$, the fraction of the scenario's techniques that belong to the actor's profile. We report both: recall captures how completely the actor is covered, precision how focused the scenario is on that actor. Because the grounded conditions inject R, recall alone could be inflated by mere instruction-following; precision and the *off-profile* count $|G \setminus R|$ are not—injecting techniques does not raise precision unless the model also refrains from adding unrelated ones—so they serve as our primary check against circular fidelity. As a specificity control we additionally score recall against a *non-injected* actor's ATT&CK profile (the other sector's actor).

D2 – Kill-chain fidelity. Let $K = \langle k_1, \ldots, k_m \rangle$ be the ordered kill-chain phases declared by the scenario, drawn from the seven canonical phases $\mathcal{P}$. Phase *coverage* is $|\{k_i\} \cap \mathcal{P}|/|\mathcal{P}|$, and *ordering* is the degree to which K respects the canonical phase order, computed as $1 - \mathrm{inv}(K)/\binom{m}{2}$, where $\mathrm{inv}(K)$ is the number of out-of-order phase pairs. Both lie in $[0, 1]$, with 1 denoting full coverage and a strictly monotonic phase sequence.

3.3 Generation pipeline

Given a parameter set—sector, fictional target-organisation profile, reference threat actor, and exercise objective (an IR exercise spanning the NIST lifecycle: prepare; detect and analyse; contain, eradicate and recover; post-incident)—the pipeline prompts an LLM to emit a single structured scenario comprising a background and environment description, a kill-chain-ordered attack narrative in which each step is tagged with an ATT&CK technique, a timed sequence of injects, illustrative synthetic artifacts, and expected defender decision points mapped to the NIST phases. To keep the study at the scenario-description level, the generator is instructed never to emit working exploit code; logs, alerts, and IOCs are illustrative and clearly synthetic. The same prompt template and decoding settings are used in every condition; only the grounding context varies.

3.4 Grounding conditions

We compare three levels of threat-intelligence grounding:

- **C0 – None.** Only the parameter set is supplied; no external threat-intelligence context.
- **C1 – ATT&CK.** The reference actor's MITRE ATT&CK technique set is injected as grounding context.
- **C2 – ATT&CK + CTI.** As C1, plus a short excerpt from a public incident report for the actor's documented campaign.

3.5 Evaluation procedure

Each generated scenario is scored by (a) the objective anchors of Section 3.2 (D1–D2) and (b) two LLM judges (all six dimensions), each prompted with the rubric of Section 3.1 and blind to the grounding condition that produced the scenario. Using two judges from different model families than the generator lets us report inter-judge agreement and reduces shared-bias artifacts. We further assess *repeatability* by comparing the runs within each (sector, condition) cell on identical inputs, following the consistency analysis used in prior cyber-range evaluation [1]. The concrete models, statistical tests, and sample sizes are given in Section 4.

4 Experimental Setup

We generate **2 sectors** $\times$ **3 conditions** $\times$ **3 runs** $=$ **18 scenarios**. The two sectors are **energy / oil pipeline (IT with OT impact)** and **healthcare**. For each sector we fix one reference threat actor with a well-documented public campaign so that the CTI condition has real source material: DarkSide (Colonial Pipeline, 2021) for the energy/pipeline sector and Conti (HSE Ireland, 2021) for healthcare. The target organisation in each sector is fictional but realistic (a regional fuel-pipeline operator and a national hospital network) and is held fixed

across all conditions, so that scenario variation stems from grounding rather than from organisational context. The reference TTP set for each actor is taken from its MITRE ATT&CK profile (DarkSide, and Conti/Wizard Spider, G0102); the ATT&CK condition (C1) injects this technique set, and the CTI condition (C2) additionally supplies a short excerpt from a public incident report for the campaign.

All models are accessed through a single OpenAI-compatible gateway. Scenarios are produced by one generator LLM (Claude Haiku 4.5, *claude-haiku-4-5*) at temperature 0.8 to induce variation across the three runs. Realism is rated by two judge LLMs from different model families than the generator—Mistral Small (*mistralai/mistral-small-2603*) and Llama 3.3 70B Instruct (*meta-llama/llama-3.3-70b-instruct*)—at temperature 0.2, each prompted with the six-dimension rubric and blind to the grounding condition. Decoding is otherwise left at provider defaults with a 4000-token limit, and failed API calls are retried with exponential backoff. Crucially, the organisation profile, exercise objective, prompt template, and decoding settings are held identical across conditions; only the grounding block varies, so any difference between conditions is attributable to grounding alone.[1]

From each scenario's structured output we compute the objective metrics defined in Section 3: TTP fidelity against the actor's ATT&CK profile, and kill-chain phase coverage and ordering. For inference we use non-parametric tests, given the small per-cell sample (n=6 per condition): the Kruskal–Wallis test across the three conditions and pairwise Mann–Whitney tests with Holm correction, at a significance level of $\alpha = 0.05$. Inter-judge agreement is reported as quadratic-weighted Cohen's κ, and repeatability as the within-cell standard deviation of the overall judge score across the three runs.

5 Results

We report objective metrics (TTP fidelity, kill-chain coverage and ordering) and LLM-as-judge ratings across the three grounding conditions (Table 2). Group comparisons use the Kruskal–Wallis test across C0/C1/C2; pairwise comparisons use Mann–Whitney with Holm correction.

5.1 Threat-intelligence grounding improves objective threat-actor fidelity

Grounding improved both facets of objective fidelity (Table 2). Recall of the actor's MITRE ATT&CK techniques rose from 0.67 (C0) to 0.97 (C1) and 1.00 (C2) (Kruskal–Wallis $p = 0.0008$), and precision more than doubled, from 0.21 (C0) to 0.50 (C1) and 0.53 (C2) ($p = 0.003$). For both measures, pairwise tests (Table 3) show that the grounded conditions significantly exceed the ungrounded

[1] Exact prompts and the reference TTP sets will be released with the replication package.

Table 2. Objective fidelity and judge ratings by grounding condition (n=6). Recall/precision are computed against the actor's ATT&CK profile; off-profile is the mean count of techniques outside it.

Condition	Recall	Precision	Off-profile	KC cov.	Judge overall
C0 – None	0.67	0.21	12.8	0.81†	5.00
C1 – ATT&CK	0.97	0.50	5.2	1.00	4.83
C2 – ATT&CK+CTI	1.00	0.53	5.0	1.00	4.67
Kruskal–Wallis p	**0.0008**	**0.003**	—	—	0.150

Table 3. Pairwise Mann–Whitney (Holm-adjusted) on recall and precision.

Comparison	Recall (Holm p)	Precision (Holm p)
C0 vs. C1	**0.010**	**0.015**
C0 vs. C2	**0.007**	**0.015**
C1 vs. C2	0.405	0.870

baseline, whereas adding a CTI excerpt on top of ATT&CK (C1 vs. C2) yields no significant further gain. In other words, most of the fidelity improvement comes from supplying the actor's ATT&CK techniques; the additional CTI excerpt contributes little for these ransomware actors, whose public ATT&CK profiles already capture much of their tradecraft.

Synthesis, not echo. The fidelity gain does not reduce to copying the injected techniques. Even under grounding, mean precision was only 0.50 (C1) and 0.53 (C2)—about half of each grounded scenario's techniques lie outside the injected set—and scenarios still contained on average about five off-profile techniques (e.g. reconnaissance and weaponisation techniques never supplied), down from 12.8 at C0, indicating genuine synthesis around the grounded core rather than mechanical reproduction. As a specificity check, coverage of the injected actor rose to near-complete under grounding (0.97 at C1, 1.00 at C2) while coverage of a non-injected actor's profile stayed at its shared baseline ($\sim$0.67; the two ransomware actors share a common technique core), a significant gap (Wilcoxon signed-rank, injected $>$ non-injected, $p < 0.001$). Grounding therefore raises coverage of the *intended* actor specifically, not overlap with any actor.

5.2 Kill-chain fidelity is at ceiling

Kill-chain ordering was perfect (1.00) in every scenario, and phase coverage was at or near ceiling under grounding (1.00 for C1 and C2). Because the generator emits kill-chain phases in canonical order by construction, we treat D2 as a

Table 4. Mean LLM-judge ratings (1–5) by dimension and condition.

Condition	D1	D2	D3	D4	D5	D6
C0	5.00	5.00	4.67	4.83	4.83	5.00
C1	4.75	5.00	5.00	4.67	4.83	5.00
C2	4.75	5.00	4.75	4.50	4.67	5.00

sanity check rather than a discriminative measure and draw no inference from the small coverage gap.[2]

5.3 LLM-judge ratings saturate and do not reflect the effect

In contrast to the objective metric, judge ratings were uninformative. Across all six dimensions ratings clustered at the maximum (Table 4): overall scores were 5.00, 4.83, and 4.67 for C0, C1, and C2, with no significant difference across conditions (Kruskal–Wallis $p = 0.150$) and, if anything, a slight non-significant decline as grounding increased. No individual dimension reached significance (the closest, organisational realism, gave $p = 0.057$). Thus the judge failed to detect the grounding-driven fidelity improvement that the objective metric isolates, consistent with documented limitations of the LLM-as-a-judge paradigm [9].

5.4 Inter-judge agreement and repeatability

Quadratic-weighted inter-judge agreement was negligible ($\kappa = 0.07$, 108 paired dimension scores). This low value does not indicate that the judges contradicted one another—both overwhelmingly assigned the maximum score—but reflects the near-absence of variance: with almost all ratings at ceiling, chance-corrected agreement is uninformative (the high-agreement/low-κ paradox). Repeatability across the three runs per cell was high (mean within-cell standard deviation of overall judge score = 0.19), though this stability likewise partly reflects the ceiling rather than demonstrated reliability.

5.5 Summary

Threat-intelligence grounding produced a large, significant improvement in objective threat-actor fidelity—both recall (C0 $\rightarrow$ C1/C2; $p = 0.0008$) and precision ($p = 0.003$)—driven mainly by ATT&CK grounding, while the LLM-judge component saturated and neither captured the effect nor yielded usable agreement. The objective, framework-anchored metrics were therefore substantially more discriminative than judge ratings for assessing scenario realism—a point we develop in Section 6.

[2] The sub-ceiling C0 coverage (0.81) is partly an artifact: one ungrounded scenario emitted phase labels with free-text annotations that failed exact matching, scoring 0 coverage. After normalising such labels this scenario is at ceiling, so the residual D2 difference is not attributable to grounding.

6 Discussion

Interpretation. Returning to our research question—whether threat-intelligence grounding improves the threat-actor fidelity of IR exercise scenarios—and whether that improvement is detectable across evaluation dimensions—the answer is dimension-dependent. On the objective dimension of threat-actor fidelity, grounding had a large and significant effect: supplying the actor's MITRE ATT&CK techniques raised recall of its documented techniques to near-complete coverage and more than doubled precision relative to an ungrounded baseline. The marginal contribution of the CTI excerpt beyond ATT&CK was small and non-significant, which is consistent with evidence that structured CTI encodes only fragmented behavioural slices [11]: once the actor's ATT&CK profile is provided, a short report excerpt adds little additional technique-level fidelity. On the dimensions rated by the LLM judge, by contrast, no effect was detectable—but this reflects the instrument rather than the scenarios. Judge ratings clustered at the maximum and, if anything, declined slightly under grounding; we conjecture that ungrounded scenarios, which include more varied but off-profile techniques, read as richer to a judge that does not verify actor fidelity, so the judge's implicit notion of realism diverges from objective fidelity. This saturation and the resulting negligible agreement are consistent with documented limitations of the LLM-as-a-judge paradigm [9], and they echo the difficulty prior work reports in measuring generated-scenario realism reliably [1]. The practical implication is that realism assessment for AI-generated exercise content should not rest on LLM-judge ratings alone; pairing them with objective, framework-anchored metrics—as our rubric does—is necessary to surface effects that judges miss.

Limitations. Realism is assessed without human subject-matter experts; LLM-as-judge ratings are a proxy and may share biases with the generator. The sample is small (18 scenarios, two sectors, one actor per sector) and uses specific model versions, limiting generalisation. C2 grounding depends on the quality and coverage of the selected public reports.

Future Work. Priority directions include benchmarking the rubric against human expert judgements and extending the ablation to more sectors and structurally distinct threat actors beyond the ransomware-only scope of this study.

Threats to validity. Construct: the rubric operationalises "realism" but cannot fully capture experienced realism in a live exercise. *Internal:* prompt and decoding settings are held fixed, but residual confounds may remain. *External:* findings may not transfer to other sectors, actors, or exercise formats. We treat results as evidence for a measurement approach and a grounding effect, not as a deployment claim.

Circularity of the fidelity metric. Because grounded conditions inject the actor's ATT&CK techniques and recall measures how many reappear, part of the recall gain reflects instruction adherence. We therefore treat the result as *threat-actor*

fidelity to injected intelligence, not realism in a stronger sense. Three observations bound the concern: the effect is not guaranteed (an LLM may ignore context); precision rose only to $\sim$0.5 and $\sim$5 off-profile techniques were retained, so the generator synthesises rather than copies; and the non-injected-actor coverage control ($\sim$0.67 vs. $\rightarrow$1.00, $p < 0.001$) confirms actor-specific shaping rather than generic inflation.

Scope and ethics. This work targets the scenario-description layer for defensive (blue-team) training and does not produce executable exploits or attack tooling.

7 Conclusion

We set out to test whether grounding LLM-based scenario generation in threat intelligence improves the threat-actor fidelity of incident-response exercise scenarios—and whether that improvement is detectable by LLM-as-judge ratings—across two critical sectors. To do so we introduced a six-dimension realism rubric that pairs framework-anchored objective metrics—TTP fidelity against MITRE ATT&CK and kill-chain phase coverage and ordering—with LLM-as-judge ratings, and we ran a controlled ablation across two critical sectors and three grounding conditions.

Grounding LLM-based scenario generation in threat intelligence measurably improves the threat-actor fidelity of the generated scenarios, but this improvement is visible only through objective, framework-anchored metrics: LLM-as-judge ratings saturate and miss it, so they are insufficient on their own as a measure of scenario realism. The effect was driven mainly by supplying the actor's ATT&CK profile, with the additional CTI excerpt contributing little beyond it.

These findings have two implications. For practice, threat-intelligence grounding —particularly ATT&CK-based grounding—is a low-cost lever for making AI-generated exercise scenarios more faithful to real adversaries. For evaluation, scenario realism should not be assessed with LLM-judge ratings alone; objective, reusable metrics such as those in our rubric are needed to surface effects that judges miss. We release the rubric as an instrument for this purpose.

Limitations include a small sample (18 scenarios, two ransomware actors sharing a common technique core), no human expert validation, and dependence on specific model versions. Future work should benchmark the rubric against expert judgements, extend the ablation to structurally distinct actors, and deploy scenarios in a live cyber range.

Acknowledgments. This work was supported by NRF Korea (Ministry of Education, No. RS-2025-25398603 and No. RS-2025-25411243) and by KEIT/MOTIE Korea (Technology Innovation Program, No. RS-2025-02317769).

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

References

1. Palumickas, M., Yamin, M.M., Katt, B., Lal, C.: Evaluating ASCERT: generative AI for cyber-range scenario generation. Int. J. Inf. Secur. **25**, 8 (2026). https://doi.org/10.1007/s10207-025-01179-w
2. Tung, I.K., Shi, Y.X., Chien, A., Liu, W., Zheng, L.: AEGIS: white-box attack path generation using LLMs and training effectiveness evaluation for large-scale cyber defence exercises. arXiv:2601.22720 (2026)
3. Yamin, M.M., Hashmi, E., Ullah, M., Katt, B.: Applications of LLMs for generating cyber security exercise scenarios. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3468914
4. Roh, S., Kim, T.-S.: LLM-Based Automated Generation and Tri-Modal Representation of Cyber Attack Scenario. IEEE Access (2025). https://doi.org/10.1109/ACCESS.2025.3597850
5. Nakata, R., Otsuka, A.: CyExec*: Automatic Generation of Randomized Cyber Range Scenarios. In: Proc. 7th Int. Conf. on Information Systems Security and Privacy (ICISSP 2021), pp. 226–236. SciTePress (2021). https://doi.org/10.5220/0010324502260236
6. Zhai, J., Li, Z., Miao, H., Li, Z., Zhou, X., Yang, H.: Automatic Generation of Cybersecurity Teaching Cases Using Large Language Models. Computer Applications in Engineering Education (2025). https://doi.org/10.1002/cae.70081
7. Lin, X., Zhang, J., Deng, G., Liu, T., Zhang, T., Guo, Q., Chen, R.: IRCopilot: Automated Incident Response with Large Language Models. arXiv preprint arXiv:2505.20945 (2025). https://doi.org/10.48550/arXiv.2505.20945
8. Damianou, A., Mazi, M., Rizos, G., Bezas, K., Lazaridis, G., Voulgaridis, A., Votis, K., Tzovaras, D.: Measuring the Unseen: A Situational Awareness Scoring Framework for Cyber Range Training. Research Square (2025). https://doi.org/10.21203/rs.3.rs-7997125/v1
9. Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., Shu, K., Cheng, L., Liu, H.: From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge. arXiv preprint arXiv:2411.16594 (2025). https://doi.org/10.48550/arXiv.2411.16594
10. Garg, A., Boualouache, A., Imeri, A., et al.: A Survey of Cyber Range Training Exercise Scenario Description, Generation, and Execution. TechRxiv (2025). https://doi.org/10.36227/techrxiv.175942879.94813577/v1
11. Ferraz, Á. L. R., Barbieri, S., de Souza, M. E., Júnior, L. A. P.: The Procedural Semantics Gap in Structured CTI: A Measurement-Driven STIX Analysis for APT Emulation. arXiv preprint arXiv:2512.12078 (2026). https://doi.org/10.48550/arXiv.2512.12078
12. Lazer, S. J., Aryal, K., Gupta, M., Bertino, E.: A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes. arXiv preprint arXiv:2601.05293 (2026). https://doi.org/10.48550/arXiv.2601.05293

Diagnosing Failure of Witness-Guided Concept Suppression

Sangjoon Lee[0009−0006−4296−2832], Hyunsik Na[0000−0002−7123−4467], and Daeseon Choi[0000−0002−1438−0265]

Soongsil University, Seoul 07027, Republic of Korea

Abstract. Negative guidance aims to prevent a pretrained generative model from producing a target concept by manipulating the sampling trajectory at inference time. Existing methods define this repellency as a predefined geometry, such as distance to a finite negation set or target distribution. A natural alternative is to replace this proxy with a learned semantic distance: use a witness to score predicted clean estimates and guide sampling across its decision boundary. In this paper, we show that such substitution produces classifier-level removal candidates, but it does not yield reliable genuine removals. We use projected gradient descent as a diagnostic: applying it to the same score used for guidance crosses the witness boundary while independent evaluators still recognize the target class. Finally, we replace the standard witness with an adversarially robust one to test whether the unsuccessful result is merely an artifact of using a non-robust standard encoder. Both fail to restore a clean removal regime; the realized displacements can survive the sampler while remaining transferable-adversarial or destructive rather than semantic. The failure is therefore an objective gap: the witness' decision boundary do not stand for semantic erasure.

Keywords: Diffusion Models · Negative Guidance · Adversarial Examples

1 Introduction

The advance of diffusion models [8, 22] has inevitably raised safety concerns regarding the generation of unsafe or unauthorized material; violent, sexually explicit, or copyright-infringing imagery. Concept erasure, or suppression are therefore critical for safe AI deployment in the real world. **Negative guidance** is a prominent class of concept suppression mechanism developed for autoregressive generative models, which adds a force that repels the sampling trajectory away from the unwanted contents. While a substantial line of negative guidance

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00398353, Development of Countermeasure Technologies for Generative AI Security Threats)

papers focuses on the text condition, recent works [12, 13] define repellency directly in the pixel or latent space. They provide principled ways to repel samples from a specified data region represented by a fixed distance metric, and a pre-obtained dataset; which contains undesired contents such as memorized outputs or harmful concepts. However, these methods expose a more fundamental obstacle; finite negation set and a distance metric do provide an empirical proxy for the undesired region, but they do not solve the problem of defining the semantic region of an open-ended concept.

Such limitation raises the question whether replacing the proxy with a learned semantic score, reading a concept score from a pretrained encoder and guiding the sampler to reduce it, would yield more effective guidance signal. In this paper, however, we argue that the substitution does not work in practical settings. Gradient-based guidance can drive the concept score below without actually removing the concept. The reason is structural: the update is feasibility-seeking with respect to the constraint. At each step, it tries to move the trajectory to the outside of concept distribution derived from the encoder, thereby satisfying a property of the learned witness rather than a property of the human perception. Crossing this boundary certifies, at best, that the encoder no longer reports the concept; it does not certify that the concept is absent.

To study this modification in isolation, we deliberately move from the full text-to-image pipeline to a controlled, class-conditional pixel-space setting. As reported by Gadginmath et al. [6], a typical autoencoder is not a diffeomorphism; displacements the guidance makes toward a witness-feasible latent therefore map to image-space changes in an uncontrolled, non-invertible way, so a lack of success in removing the concept could reflect this decoder pathology rather than the witness objective. Taking the target concept to be a class label and the witness to be an image classifier removes such confounds, so that any observed failure is attributable to the witness objective itself rather than to the surrounding machinery. Specifically, we adopt EDM from Karras et al. [10] as a backbone, and two datasets; CIFAR-10 [14] and ImageNet64 [3].

Our contributions are as follows:

- We show that witness-alone negative guidance fails to achieve genuine concept removal, and trace the cause to a self-defeating gradient property of the witness itself, which we isolate in a controlled pixel-space setting that removes the latent autoencoder confounds.
- We show this inability is not due to a weak encoder: an adversarially robust encoder is necessary but not sufficient, sharply reducing cheap boundary-crossing yet still not restoring removal.
- We introduce an evaluation protocol that separates witness-feasibility from genuine removal, judging each output by detectors independent of the guiding witness rather than by the witness score it was optimized against.

2 Background

2.1 Negative guidance in Text-to-Image models

Negative guidance suppresses target concepts at inference time by adding a force that steers the sampling trajectory away from undesired content. One line operates on the text condition c : Safe Latent Diffusion (SLD) [20] steers the conditional noise prediction away from a harmful prompt, and SAFREE [24] projects prompt token embeddings off a toxic-concept subspace. These methods inherit the prompt as their attack surface and are bypassed by adversarial prompts, motivating a complementary line that defines repellency directly in the latent or image space, independently of c.

Latent- and image-space methods commonly define repellency using a fixed, finite negation set $D^- = \{y_i\}$ of unsafe or protected reference samples. Safe-Denoiser [11] takes a denoiser-level view: it decomposes the data denoiser into safe and unsafe components and estimates the unsafe component from D^-, using a corruption-kernel-weighted average over negative samples. Safety-Guided Flow (SGF) [12] gives a complementary energy-based formulation, casting sample repellency as the gradient of squared Maximum Mean Discrepancy between the predicted clean sample and D^- in an RBF-kernel feature space. In both cases, the repulsive direction is determined by proximity to observed negative examples under a pre-defined geometry, rather than by direct membership in the open-ended target concept.

2.2 Adversarial Examples and Robustness

Neural network classifiers are vulnerable to adversarial examples: imperceptible input perturbations that flip the predicted label while leaving the content unchanged to a human observer [23]. Such perturbations are found efficiently by first-order attacks, most commonly projected gradient descent (PGD) [15], which iteratively ascends the classifier's loss within a small norm ball. This vulnerability has been attributed not to a defect but to a model's reliance on predictive yet non-robust features [9], which implies that the decision boundary of a standard learned classifier can be crossed by a small displacement that does not change the semantic content.

The canonical defense is adversarial training [15], which produces robust classifiers that substantially resist analogous attacks. Robustness does not, however, eliminate transferability: a perturbation crafted against one model often fools others [17], and even two independently trained robust classifiers can share blind spots, so a displacement that crosses one robust boundary may carry across another. Consequently, the effect of a perturbation on any single classifier, even a robust one, is not by itself evidence of a semantic change; establishing it requires agreement among detectors independent of the one the displacement was optimized against.

3 Witness Barrier

Consider inference-time suppression of a single target concept c during the sampling process of a diffusion model. Let $x_t \in \mathbb{R}^{C\times H\times W}$ denote the noisy image at discrete, non-negative timestep $t \in [0, T]$, where x_T corresponds to the initial noise $\epsilon \sim \mathcal{N}(0, I)$, and x_0 to the final sample generated by an arbitrary denoiser D following the sequence:

$$x_{t-1} = D(x_t, t),\ t \in [1, T] \tag{1}$$

At each step, one can induce a predicted clean estimate $\hat{x}_{0|t}$ from x_t. For example, in the case of score-based diffusion models, $\hat{x}_{0|t}$ follows from the score via Tweedie's formula [4] under Gaussian corruption. For denoisers that predict the noise or the clean sample directly, $\hat{x}_{0|t}$ denotes the corresponding standard x_0-prediction at step t. We do not account for latent space diffusion models to avoid the failure mode caused by non-diffeomorphic autoencoders, reported by Gadginmath et al. [6].

Now we refer to an arbitrary differentiable encoder used to score evidence of concept c as the *witness* ϕ. Although one may utilize a text encoder (e.g. CLIP [18]) for ϕ [1], we only inspect classifier case for evaluation and diagnosis. Let $z_\phi(x) \in \mathbb{R}^K$ denote its logits over K classes and let e_j be the j-th coordinate basis vector. We instantiate the scalar witness score as a target-versus-rest logit contrast,

$$s_{\phi_c}(x) = z_\phi(x)^\top v_c, \qquad v_c = \frac{e_c - \frac{1}{K-1}\sum_{j\neq c} e_j}{\left\| e_c - \frac{1}{K-1}\sum_{j\neq c} e_j \right\|_2}. \tag{2}$$

Thus, up to the positive normalization factor in v_c, $s_\phi(x)$ is the target logit minus the mean non-target logit. The direction v_c is fixed once the target class is chosen and is not learned.

Given a fixed threshold $\tau \in \mathbb{R}$, we define a witness barrier $h : \mathbb{R}^{C\times H\times W} \to \mathbb{R}$ as

$$h(x) = \tau - s_{\phi_c}(x). \tag{3}$$

Thus $h \geq 0$ means that the witness deems the current estimate feasible—it no longer reports c—while $h < 0$ means that the witness still reports it. For brevity, we omit c from the score notation in the rest of this paper. The denoising process with negative guidance is defined on the predicted clean estimate with repellency from the gradient of witness barrier:

$$x_{t-1} = D(x_t, t) + \rho_t \nabla_{\hat{x}_{0|t}} h(\hat{x}_{0|t}), \tag{4}$$

where the gradient is taken with respect to the predicted clean estimate $\hat{x}_{0|t}$, and $\rho_t \geq 0$ is the effective guidance strength applied at step t. Concretely, this is realized by perturbing the estimate as $\hat{x}_{0|t} \leftarrow \hat{x}_{0|t} + \beta \nabla_{\hat{x}_{0|t}} h$ and recomputing the sampler update, which adds the perturbation to the output scaled by the step's known transfer coefficient, so that $\rho_t = \beta\, c_t$ with c_t fixed by the sampler.

4 Diagnosing Witness-Guided Suppression

We evaluate each guided output using detectors independent of ϕ, visual or task-level inspection of the target content, and the coherence of the resulting image. Based on these criteria, we assign each output to one of the following outcomes:

- **Genuine removal.** The target concept is absent under the independent evaluators and visual inspection, while the output remains coherent and preserves its non-target content.
- **Adversarial example against ϕ.** The output x satisfies $h(x) \geq 0$, but the target concept remains present under independent evaluation (e.g. visual inspection).
- **No removal.** The target concept remains present and the output does not satisfy the witness constraint.
- **Degradation or destruction.** The intervention damages image quality, coherence, or non-target content. Such outputs are not counted as successful removal, regardless of the witness score or the response of other detectors.

In advance, to test whether the gradient of ϕ follows a semantic removal direction or merely exploits the decision boundary of the same witness, we apply PGD directly to arbitrary images. Given an image $\hat{x}^{(0)}$, we obtain $\hat{x}^{\mathrm{adv}}$ by iteration:

$$\hat{x}^{(n+1)} = \Pi_{\mathcal{B}_\epsilon(\hat{x})}\left[\hat{x}^{(n)} - \alpha \operatorname{sign}\left(\nabla_{\hat{x}} s_\phi(\hat{x}^{(n)})\right)\right], \qquad n = 0, \ldots, N-1. \tag{5}$$

$$\hat{x}^{\mathrm{adv}} := \hat{x}^{(N)}. \tag{6}$$

Here α is the PGD step size, N is the number of steps, and $\Pi_{\mathcal{B}_\epsilon(\hat{x})}$ projects back to the admissible perturbation set $\mathcal{B}_\epsilon$, an ℓ_∞-norm ball around $\hat{x}$ with distance ϵ, together with the valid image range. Because the update descends s_ϕ, it explicitly searches for a nearby image that the same witness regards as less aligned with concept c. We quantify how efficiently the witness score can be reduced relative to the resulting perceptual change, measured by LPIPS [26], as

$$PGD_\phi = \frac{s_\phi(\hat{x}) - s_\phi(\hat{x}^{\mathrm{adv}})}{\mathrm{LPIPS}\big(\hat{x}, \hat{x}^{\mathrm{adv}}\big)}. \tag{7}$$

A large PGD_ϕ indicates that the score of the guiding witness can be reduced substantially with little perceptual displacement.

5 Experiments

5.1 Experimental Settings

Models We utilize two class-conditional EDMs trained with CIFAR-10 and ImageNet64, whose network predicts the clean sample directly so the predicted clean

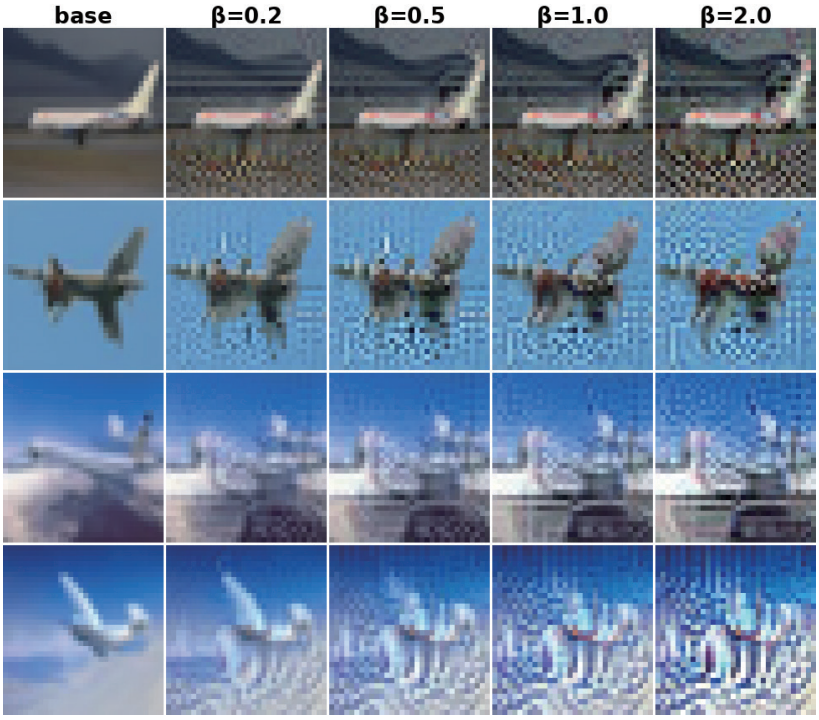


Fig. 1. Witness-guided negative guidance across strength β on CIFAR-10 (target airplane; representative samples, one per row). Small β leaves the output near-identical to the baseline while the witness reports the concept absent; larger β turns the perturbation into adversarial texture, then destruction.

estimate $\hat{x}_{0|t}$ scored by the witness is read off from the denoiser output without additional inversion. For CIFAR-10, we use the deterministic EDM–Euler integrator for 50 steps for sampling and adopt WRN-28-10 [25] for witness. For ImageNet-64, we use the stochastic EDM sampler with churn for 256 steps for sampling and adopt standard ResNet [7] based classifier, checkpoint provided from standard *torchvision* library [16].

Evaluators Removal is judged only by detectors *independent* of the guiding witness ϕ: a robust classifier trained with a different recipe from ϕ on a disjoint checkpoint (and, on CIFAR-10, a different architecture as well), plus a non-robust classifier as a transfer probe. For the standard witness the robust evaluators are Carmon et al. [2] and Engstrom et al. [5] on CIFAR-10, and Salman et al. [19] on ImageNet-64, each paired with a non-robust probe (a standard WRN-28-10 on CIFAR-10 and VGG-19 [21] on ImageNet-64). When the guiding witness is itself robust (Sec. 5.4), the model that serves as the guide is excluded from

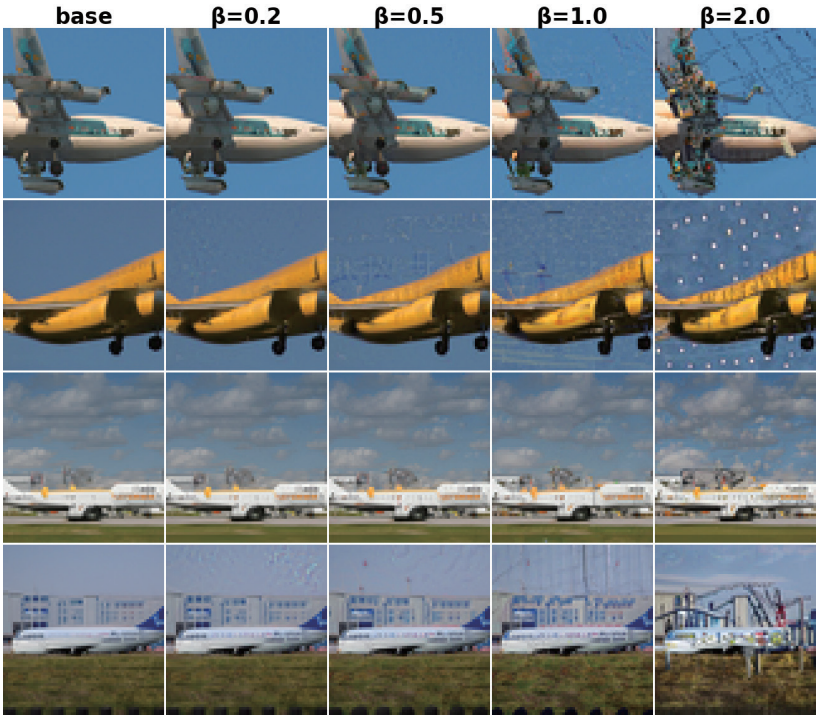


Fig. 2. The same experiment on ImageNet-64 dataset. Guidance satisfies the witness at every β, yet the output is adversarial-example-against-ϕ at small β and destruction at large β—never a coherent non-target class.

that experiment's evaluators: on CIFAR-10 the guide is the Engstrom checkpoint and the robust evaluator is then the disjoint Carmon checkpoint, while on ImageNet-64 the Engstrom guide is scored by the disjoint Salman checkpoint. No experiment therefore scores a perturbation with the same checkpoint that produced it, so the robust-to-robust transfer in Sec. 5.4 is always measured on a separate model. What certifies unchanged content is agreement of the *robust* evaluator, which resists a perturbation optimized against a standard witness, not of an arbitrary classifier.

Hyperparameters The threshold τ is calibrated offline, separately for each witness and dataset pair, by scoring generated target-class samples against a held-out pool of the remaining classes and placing τ to separate the two score distributions; every witness separates the two cleanly, attaining an AUROC of at least 0.99 in all four settings, so each witness is a near-perfect classifier on clean images. We inject the guidance of Eq. (4) by modifying $\hat{x}_{0|t}$ and recomputing the sampler update from the modified estimate; the guidance strength is $\rho_t =$

Table 1. Standard-witness outcomes over 100 samples per dataset and β. Adversarial-ϕ: satisfies the witness while the independent robust evaluator still reports the target; destruction: evaluator drops the target but the image is incoherent.

Dataset	β	adversarial-ϕ	destruction	no removal	genuine removal
CIFAR-10	0.2	89	11	0	0
	0.5	78	22	0	0
	1.0	73	27	0	0
	2.0	58	42	0	0
ImageNet-64	0.2	92	8	0	0
	0.5	75	25	0	0
	1.0	69	31	0	0
	2.0	36	64	0	0

$\beta(\sigma_t - \sigma_{t-1})/\sigma_t$; fixed by the EDM noise levels σ_t. We vary $\beta \in \{0.2, 0.5, 1.0, 2.0\}$, spanning weak intervention to over-saturation. For each dataset and each guidance strength, we generate 100 target-class samples and assign each output to one of the failure categories defined in Sec. 4. The PGD_ϕ diagnostic (Eq. (5)) uses $N_{\text{PGD}} = 20$ signed-gradient steps of size $\alpha = 2/255$ within budget $\epsilon = 16/255$.

5.2 Witness-based Negative Guidance

CIFAR-10 On CIFAR-10 with airplane as the target, we observe *no genuine removal* among 100 samples at any β. At small β the guidance pushes the witness score below threshold with an *imperceptible* perturbation (LPIPS as low as 0.002), yet two independent robust evaluators still assign the target class at 0.92 and 1.00 and the output is visually identical to the baseline—the **adversarial-example-against-ϕ** outcome, dominant across samples (Fig. 1). Larger β grows the perturbation into **destruction**, never a coherent alternative class; the samples that flip the robust evaluator are degraded, not genuinely changed (Table 1).

ImageNet-64. The pattern persists at higher resolution and class count with a standard ResNet-50 witness (Fig. 2): guidance drives the witness score below the calibrated threshold at every β without genuine removal. At small β the robust evaluator retains the target class while the output is near-identical to the baseline; only at the largest β does it degrade into destruction. Notably, the non-robust third classifier (VGG-19) is itself fooled by the same perturbation—collapsing toward 0 where the robust evaluator holds—so witness feasibility transfers among standard classifiers without changing content, underscoring why removal must be certified against a robust detector *independent* of the witness.

5.3 PGD$_\phi$ Diagnostic

Applying Eq. (5) to generated target images drives s_ϕ far below τ at negligible perceptual cost: at $\epsilon = 16/255$, $PGD_\phi = 3.4 \times 10^3$ while an independent robust

Table 2. PGD_ϕ diagnostic for the standard witness on CIFAR-10 across attack budget ϵ. The PGD_ϕ column reports the absolute scalar from Eq. (7); the two right columns report relative speedups against semantic references. The global semantic reference is $|\Delta s_\phi|$/LPIPS over real cross-class image pairs, and the matched robust-gradient reference is a step of the same size along a robust classifier's input gradient.

ϵ	PGD_ϕ	$\times$ robust-grad-ref.	$\times$ global-ref.
4/255	4.9×10^4	117	722
8/255	9.7×10^3	23	142
16/255	3.4×10^3	8	50

Table 3. Standard versus robust witness. Robustness makes the boundary expensive to cross (lower PGD_ϕ, larger displacement to satisfy the witness) and causes the perturbation to transfer to a separate independent robust evaluator checkpoint, yet genuine removal stays zero—necessary but not sufficient.

Witness	PGD_ϕ (CIFAR)	LPIPS to cross (ImageNet)	robust eval. transfers
Standard	3.4×10^3	0.03–0.10	no (resists)
Robust	1.5×10^2	0.09–0.20	yes

classifier stays near 0.80 and the image is unchanged. Relative to semantic references, the witness score falls 50× faster than the global class-change reference and 8× faster than the matched robust-gradient reference. This is not a budget artifact: across $\epsilon \in \{4, 8, 16\}/255$ the speedup stays 50–720× against the global reference (Table 2). The witness boundary is thus crossable by a perturbation that neither independent detectors nor visual inspection regard as a concept change—exactly what makes in-loop guidance yield feasibility without removal. Because this diagnostic operates on a static image with no forward or reverse process, the pathology arises with no sampler dynamics at all; it cannot be an artifact of renoising or of the score pulling the trajectory back toward a learned mode.

5.4 Adversarially trained Witnesses

The failures so far could be an artifact of a non-robust witness, whose input-gradients are cheap, off-manifold adversarial directions [9]. We therefore repeat the study with an adversarially robust witness—a PGD-adversarially-trained ResNet-50 [5] whose gradients are perceptually aligned—and ask whether removal is restored. Here the Engstrom model serves only as the guide and is excluded from the evaluator set; the robust evaluator it transfers to is a disjoint checkpoint—Carmon on CIFAR-10, Salman on ImageNet-64—so the transfer is measured on a separate model, not by re-scoring the guide.

Robustness is necessary but not sufficient (Table 3). It makes crossing expensive: on CIFAR-10 PGD_ϕ drops from $\approx$3400 to $\approx$150, and on ImageNet-64 the robust witness needs a 2–3× larger displacement (LPIPS 0.09–0.20 vs. 0.03–0.10)

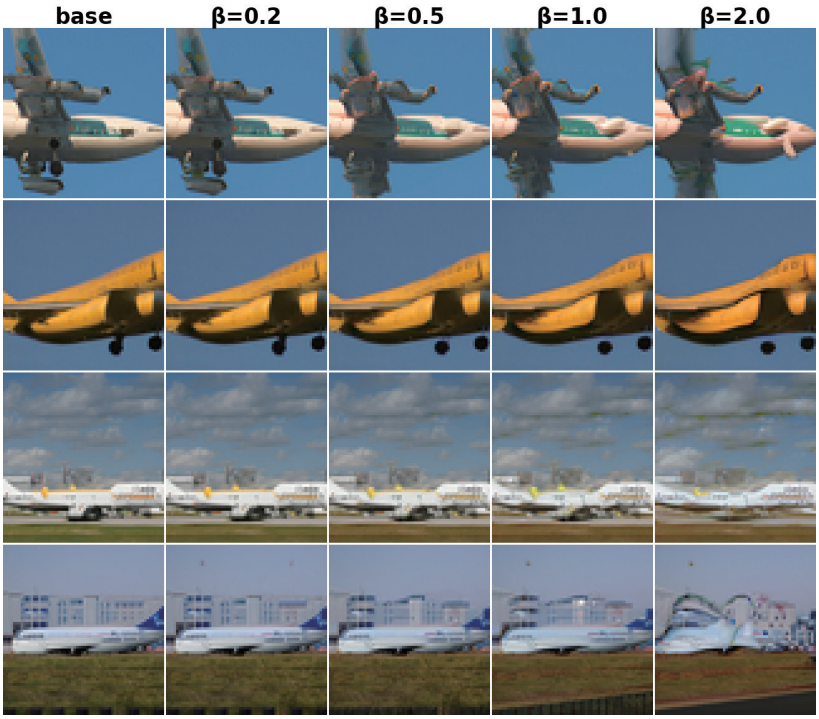


Fig. 3. Robust witness (PGD-adversarially-trained ResNet-50) on ImageNet-64. Guidance still yields no genuine removal: small β leaves the image unchanged, large β gives adversarial texture or destruction, never a coherent non-target class, even though the witness and a separate independent robust evaluator both report the target as absent.

to cross its own threshold—so encoder non-robustness was a genuine source of the cheap crossing. Yet removal is not restored: the robust witness either fails to cross at small β or, once it crosses, yields transferable-adversarial or destructive displacements with genuine removals zero on both datasets.

A further consequence is that the robust evaluator no longer certifies. With separate robust checkpoints for witness and evaluator, the realized displacement transfers between them through shared blind spots: the independent robust evaluator's confidence in the target collapses exactly where the witness's does—on ImageNet-64 to nearly 0. A standard third classifier caught this transfer on CIFAR-10 but was itself fooled on ImageNet-64, leaving visual inspection as the only universally reliable arbiter, and it shows no genuine removal on either dataset.

6 Conclusion and Limitations

We asked whether a learned witness is a better repellency signal than a hand-specified geometry. Across two pixel-space diffusion models it is not: guidance satisfies the witness at every strength, yet the outputs are witness-feasible without being concept-free, spanning imperceptible adversarial perturbations to destruction, with zero genuine removals.

The cause is structural. A witness gradient is the boundary-normal, feasibility-seeking direction of a discriminative classifier, so descending it certifies only that the encoder stops reporting the concept, not that the concept is gone. Our PGD_ϕ diagnostic crosses the same boundary with a perturbation that neither independent detectors nor visual inspection regard as a concept change. An adversarially robust witness rules out a weak encoder: it removes the cheap imperceptible crossing but not the gap. The residue is therefore an objective gap—learned witness feasibility is not semantic erasure—and telling the two apart requires evaluators independent of the guiding witness.

Our study has several limitations. We work in a controlled class-conditional pixel-space setting rather than a full text-to-image pipeline, so our experiments isolate the witness objective but do not directly test open-ended textual concepts. We also study only classifier witnesses rather than in-loop text-encoder witnesses such as CLIP, a single target concept, and one adversarial-training recipe for the robust witness. Finally, we diagnose the failure rather than fix it; pairing a witness with an explicit generative or manifold constraint, so feasibility is sought on the data manifold rather than across a classifier boundary, is left to future work.

References

1. Ahn, J., Yoon, S., Baik, S.: Egloce: Training-free energy-guided latent optimization for concept erasure (2026), https://arxiv.org/abs/2604.09405
2. Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., Duchi, J.C.: Unlabeled data improves adversarial robustness (2022), https://arxiv.org/abs/1905.13736
3. Chrabaszcz, P., Loshchilov, I., Hutter, F.: A downsampled variant of imagenet as an alternative to the cifar datasets (2017), https://arxiv.org/abs/1707.08819
4. Efron, B.: Tweedie's formula and selection bias. Journal of the American Statistical Association **106**(496), 1602–1614 (2011), http://www.jstor.org/stable/23239562
5. Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., Madry, A.: Adversarial robustness as a prior for learned representations (2019), https://arxiv.org/abs/1906.00945
6. Gadginmath, D., Allibhoy, A., Pasqualetti, F.: Provably safe generative sampling with constricting barrier functions (2026), https://arxiv.org/abs/2602.21429
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)

9. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features (2019), https://arxiv.org/abs/1905.02175
10. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: Proc. NeurIPS (2022)
11. Kim, M., Kim, D., Yusuf, A., Ermon, S., Park, M.: Training-free safe denoisers for safe use of diffusion models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=QQS7TudonJ
12. Kim, M., Kim, Y.H., Park, M.: Safety-Guided Flow (SGF): A unified framework for negative guidance in safe generation. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=EA80Zib9UI
13. Kirchhof, M., Thornton, J., Béthune, L., Ablin, P., Ndiaye, E., Cuturi, M.: Shielded diffusion: Generating novel and diverse images using sparse repellency (2025), https://arxiv.org/abs/2410.06025
14. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (05 2012)
15. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks (2019), https://arxiv.org/abs/1706.06083
16. Marcel, S., Rodriguez, Y.: Torchvision the machine-vision package of torch. In: Proceedings of the 18th ACM International Conference on Multimedia. p. 1485–1488. MM '10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1873951.1874254, https://doi.org/10.1145/1873951.1874254
17. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples (2016), https://arxiv.org/abs/1605.07277
18. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. CoRR **abs/2103.00020** (2021), https://arxiv.org/abs/2103.00020
19. Salman, H., Yang, G., Li, J., Zhang, P., Zhang, H., Razenshteyn, I., Bubeck, S.: Provably robust deep learning via adversarially trained smoothed classifiers. Curran Associates Inc., Red Hook, NY, USA (2019)
20. Schramowski, P., Brack, M., Deiseroth, B., Kersting, K.: Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014), https://api.semanticscholar.org/CorpusID:14124313
22. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=PxTIG12RRHS
23. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2014), https://arxiv.org/abs/1312.6199
24. Yoon, J., Yu, S., Patil, V., Yao, H., Bansal, M.: Safree: Training-free and adaptive guard for safe text-to-image and video generation. In: International Conference on Learning Representations (2025)

25. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)
26. Zhang, R., Isola, P., Efros, A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (01 2018). https://doi.org/10.48550/arXiv.1801.03924

FLEX: Frame-Level Explanations for Trustworthy Audio Deepfake Detection

Woongjae Lee[0009−0001−7783−3022], Seoyoung Park[0009−0003−0504−730X], Inho Kim[0009−0001−0876−1803], Thien-Phuc Doan[0000−0001−7988−5953], and Souhwan Jung[0000−0003−2676−3412]

Soongsil University, Seoul, Korea
{woongjae,seozxro,inho107}@soongsil.ac.kr, {phucdt,souhwanj}@ssu.ac.kr

Abstract. Audio deepfakes pose an increasing threat to digital security, requiring detection systems that provide not only accurate predictions but also interpretable evidence. Despite recent advances in audio deepfake detection (ADD), it remains unclear which temporal regions models rely on when making predictions, limiting the transparency and trustworthiness of detection outcomes. In this paper, we propose Frame-Level Explanations for Trustworthy Audio Deepfake Detection (FLEX), a framework for frame-level explanations via prototype alignment, motivated by the observation that frame-level representations form well-separated class-conditional regions in the latent space. FLEX quantifies frame importance as minimum Euclidean distance to predicted-class prototypes via k-means algorithm, requiring neither backbone modification nor additional supervision. Evaluated on ASVspoof2019 and In-the-Wild datasets, FLEX more effectively identifies decision-critical frames than existing explanation methods on in-domain data while maintaining competitive performance on out-of-domain. It also maintains competitive stability, preserving robust importance rankings across varying signal-to-noise ratio (SNR) conditions (10–30 dB). Furthermore, our analysis reveals that FLEX captures meaningful acoustic patterns, such as the distinction between speech and non-speech regions, providing intuitive insights into the model's decision-making process.

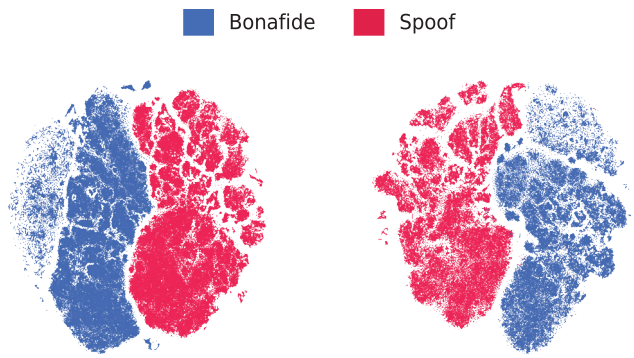
Keywords: Audio Deepfake Detection · AI security · Explainable AI · Digital Forensics.

1 Introduction

Audio deepfakes have emerged as a serious security threat, with voice spoofing attacks increasingly exploited for identity fraud, financial scams, and the fabrication of deceptive audio evidence in forensic investigations. As generative AI technologies become more accessible, the scale and sophistication of such attacks continue to grow, placing greater demand on detection systems that are not only accurate but also interpretable for security and forensic practitioners. To address this, audio deepfake detection (ADD) systems have been actively developed [5, 18], and recent advances in self-supervised learning (SSL)-based

representations have significantly improved their detection performance [1, 4]. However, the decision-making process remains opaque, as predictions are derived from aggregated latent representations that obscure the specific temporal regions responsible for classifications, making it difficult to verify detection outcomes in security and forensic investigations.

To support trustworthy decision-making in security and forensic contexts, explainability is essential for ADD systems. Early XAI research in ADD [8, 9] applied attribution methods such as Grad-CAM [14] and SHAP [6] to aggregate gradients across transformer layers. Despite architectural differences, these approaches share a common principle: interpreting model behavior through local output sensitivity. Because gradient-based attributions rely on local derivatives, small input perturbations can substantially alter the resulting explanations, even when the predicted label remains unchanged [7]. Moreover, explanations can vary depending on the selected target layer or attribution method, leading to inconsistent interpretations across otherwise identical inputs [8]. Such variability undermines explanation reliability in security and forensic applications, where consistent interpretation is essential. While prototype-based methods such as ProtoPNet [3] and ProtoPFormer [19] provide intuitive case-based explanations, they require architectural modification and retraining.



(a) SSL-AASIST (b) SSL-Conformer+TCM

Fig. 1. t-SNE visualization of frame-level representations extracted from the ASVspoof2019 LA development set. Equal numbers of bonafide and spoof frames were randomly sampled for visualization.

We observe that frame-level representations implicitly encode class-related information, where frames of the same class occupy consistent geometric regions in the latent space, even under utterance-level supervision. As illustrated by the t-SNE visualization [11] in Figure 1, bonafide and spoof frames form clearly distinguishable regions in the latent space. This suggests that discriminative evidence is already encoded in the geometric structure of frame-level

representations, providing a basis for frame-level explanation beyond utterance-level predictions.

In this paper, we propose Frame-Level Explanations for Trustworthy Audio Deepfake Detection (FLEX), a framework that provides frame-level explanations in the latent representation space. Specifically, frame importance is quantified as the minimum Euclidean distance between each frame representation and the class-conditional prototypes of the predicted class. Smaller distances indicate that a frame representation lies deeper within the predicted class region, reflecting stronger consistency with the model's decision. Our main contributions are as follows:

- A prototype-alignment based explanation framework that reveals the latent decision geometry implicitly learned by SSL-based ADD models, enabling frame-level explanations without architectural modification, retraining, or frame-level supervision.
- Comprehensive evaluation of explanation faithfulness and stability under noisy and out-of-domain conditions, demonstrating that FLEX provides reliable explanations across diverse evaluation settings.
- Interpretability analysis demonstrating that class-conditional prototypes encode meaningful acoustic patterns, supporting transparent and auditable decision-making in security investigations.

2 Method

2.1 Framework Overview

Given a test utterance, the goal of FLEX is to identify which temporal frames contribute most strongly to the model's deepfake detection decision. The input is a raw audio waveform processed by a frozen SSL-based ADD model, and the output is a frame-level importance score sequence that highlights decision-critical regions. This enables forensic analysts to localize and interpret the acoustic evidence underlying each detection outcome.

Pretrained ADD models exhibit class-conditional structure in their latent representation space, where frame-level representations organize into distinct class regions. To characterize these regions, FLEX constructs class-specific prototypes that serve as representative reference points in the latent space. Frame-level explanations are then obtained by measuring how strongly each frame representation aligns with the prototypes of the predicted class. As illustrated in Figure 2, FLEX operates on a frozen ADD model in two steps.

Prototype Construction. To leverage the geometric organization of the latent space formed during downstream adaptation, frame-level representations $h(t)$ are extracted from the linear projection layer following the SSL front-end. We approximate class-conditional regions in the latent space by constructing class-specific prototypes from training data. Each prototype represents a dense subset of frame-level representations and serves as a reference point for measuring the alignment of individual frames with the predicted class.

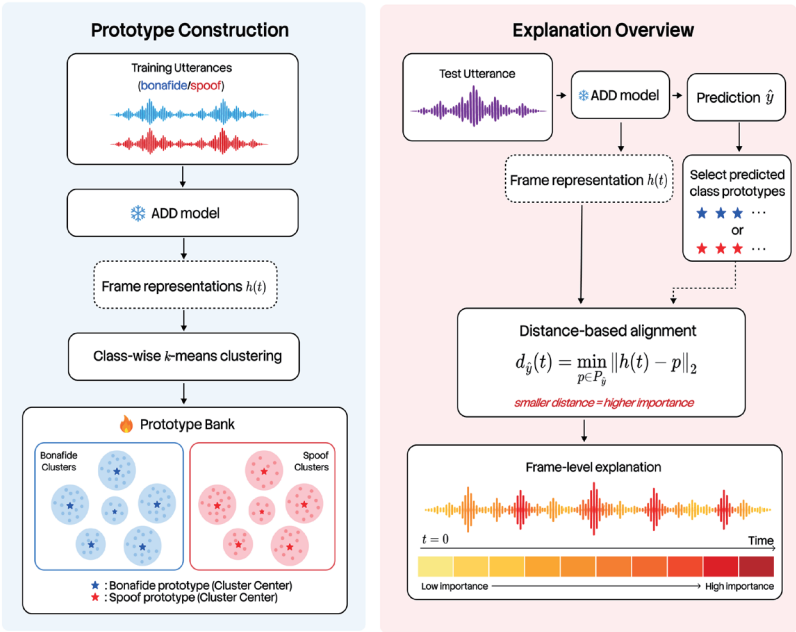


Fig. 2. Overview of the proposed FLEX framework. A frozen SSL-based audio deepfake detection model extracts frame-level representations, from which class-specific prototypes are constructed in the latent space. Alignment-based importance scores are then computed to provide frame-level explanations of the model's prediction.

Importance Scoring via Distance Measurement. For each frame-level representation, we compute its Euclidean distance to the class-conditional prototypes of the predicted class. Smaller distances indicate stronger alignment with the class region, and are thus used as a frame-level importance measure.

2.2 Prototype Initialization

To characterize the latent decision geometry, we establish a set of class-specific reference prototypes for bonafide and spoof speech. The prototypes serve as representative reference points that capture the geometric structure of the learned bonafide and spoof regions in the latent space.

Let $\mathcal{D}_{\text{train}}^{\text{bona}}$ and $\mathcal{D}_{\text{train}}^{\text{spf}}$ denote the sets of bonafide and spoof training utterances, respectively. For each utterance, we extract frame-level representations $\mathbf{h}(t) \in \mathbb{R}^D$ using the pretrained ADD model, forming class-specific frame sets:

$$\mathcal{H}_{\text{bona}} = \{\mathbf{h}_x(t) \mid x \in \mathcal{D}_{\text{train}}^{\text{bona}},\ t = 1, \ldots, T_x\} \tag{1}$$

$$\mathcal{H}_{\text{spf}} = \{\mathbf{h}_x(t) \mid x \in \mathcal{D}_{\text{train}}^{\text{spf}},\ t = 1, \ldots, T_x\} \tag{2}$$

We perform class-wise k-means clustering on $\mathcal{H}_{\text{bona}}$ and $\mathcal{H}_{\text{spf}}$ independently:

$$\mathcal{P}_{\text{bona}} = \{\mathbf{p}_k^{\text{bona}}\}_{k=1}^{K_b}, \quad \mathcal{P}_{\text{spf}} = \{\mathbf{p}_k^{\text{spf}}\}_{k=1}^{K_s} \tag{3}$$

where K_b and K_s denote the numbers of bonafide and spoof prototypes.

Class-wise clustering approximates the class-conditional structure in the latent space, where each prototype summarizes a dense subset of geometrically similar frame-level representations. To preserve the geometry of the SSL-based frame representations, we apply ℓ_2 normalization prior to clustering, making Euclidean distance equivalent to cosine similarity. Under this formulation, the resulting prototypes serve as geometrically meaningful reference points for measuring frame-level alignment.

2.3 Quantifying Alignment with Decision Geometry

Given the class-specific prototype sets $\mathcal{P}_{\text{bona}}$ and $\mathcal{P}_{\text{spf}}$, we quantify frame-level importance by measuring the alignment between each frame representation $\mathbf{h}(t)$ and the prototypes of the predicted class $\hat{y} \in \{\text{bonafide}, \text{spoof}\}$. We define the distance-based importance $d_{\hat{y}}(t)$ as the minimum ℓ_2 distance to predicted class prototypes:

$$d_{\hat{y}}(t) = \min_{\mathbf{p} \in \mathcal{P}_{\hat{y}}} \|\mathbf{h}(t) - \mathbf{p}\|_2 \tag{4}$$

Smaller distances indicate stronger alignment with the predicted class region, thereby providing stronger evidence for the model's decision. While this approach is inspired by prototype-based models like ProtoPNet [3], FLEX differs by constructing prototypes post-hoc from a frozen model without retraining. To facilitate intuitive interpretation in our qualitative analysis (Section 3.6), we further define the alignment score $w(t)$ as:

$$w(t) = 1 - d_{\hat{y}}(t) \tag{5}$$

In this formulation, higher values of w(t) directly represent stronger alignment with the predicted class region, providing a clear, heatmap-like representation of decision-critical segments.

3 Experiments

3.1 Experimental Setup

We construct prototypes using the ASVspoof2019 Logical Access [17] training set and evaluate the method on the ASVspoof2019 evaluation set (ASV19) and In-the-Wild dataset [13] (ITW).

We apply FLEX to SSL-AASIST [15] and SSL-Conformer+TCM [16], keeping all model parameters frozen and extracting frame-level representations $\mathbf{h}(t)$ from the linear layer following the SSL front-end. On ASV19, both models achieve 0.15% EER; on ITW, EERs are 11.97% (SSL-AASIST) and 7.85% (SSL-Conformer+TCM).

Table 1. Average ℓ_2 distances to bonafide (d_b) and spoof (d_s) prototypes, conditioned on predicted class. Lower values indicate stronger alignment.

	ASV19		ITW	
Predicted Class	d_b	d_s	d_b	d_s
SSL-AASIST				
Bonafide	**0.357**	1.026	**0.411**	0.512
Spoof	0.742	**0.088**	0.711	**0.124**
SSL-Conformer+TCM				
Bonafide	**0.493**	1.148	**0.803**	0.983
Spoof	1.363	**0.537**	1.215	**0.666**

The total number of prototypes was fixed to $K = 8$ for all experiments. Specifically, (K_b, K_s) was set to $(4, 4)$ for SSL-AASIST and $(3, 5)$ for SSL-Conformer+TCM based on an unsupervised clustering analysis of the latent space. The sensitivity of FLEX to different prototype configurations is further investigated in Section 3.5.

3.2 Analysis of Latent Geometric Separation

Although trained with only utterance-level supervision, we examine whether frame-level representations exhibit class-consistent alignment in the latent space. For each frame, we compute its minimum ℓ_2 distance to the bonafide and spoof prototype sets and analyze the resulting statistics conditioned on the predicted utterance-level label.

Table 1 reports the average distances on ASV19 and ITW. Across models and datasets, a clear geometric separation is observed: frames from utterances predicted as bonafide are closer to bonafide prototypes (e.g., 0.357 vs. 1.026 for SSL-AASIST on ASV19), whereas spoof predictions exhibit the opposite pattern. Notably, this separation emerges without frame-level supervision, suggesting that utterance-level optimization is sufficient to organize frame-level representations into class-dependent geometric regions prior to backend classification.

3.3 Faithfulness and Stability Evaluation

Faithfulness Evaluation We assess whether the proposed alignment score correctly identifies decision-critical frames through a positive perturbation test [2]. Frames are ranked by importance, and the highest-ranked frames are progressively masked while measuring the resulting detection error. To perform waveform-level perturbation, each frame-level importance score is mapped to its corresponding waveform segment according to the temporal resolution of the SSL representations. Following the SSL frontend, each frame represents a 25 ms

Table 2. Faithfulness evaluation via AUC-EER. Higher values indicate more precise identification of decision-critical frames through waveform-level masking.

	SSL-AASIST		**SSL-Conformer+TCM**	
Method	ASV19	ITW	ASV19	ITW
Grad-CAM [14]	36.25	40.40	33.80	39.46
GradSHAP [10]	39.45	41.53	36.49	**44.99**
DeepSHAP [10]	39.98	39.61	39.06	43.42
GATR [8]	37.14	37.50	33.98	35.75
FLEX (Ours)	**44.65**	**41.54**	**44.52**	41.78

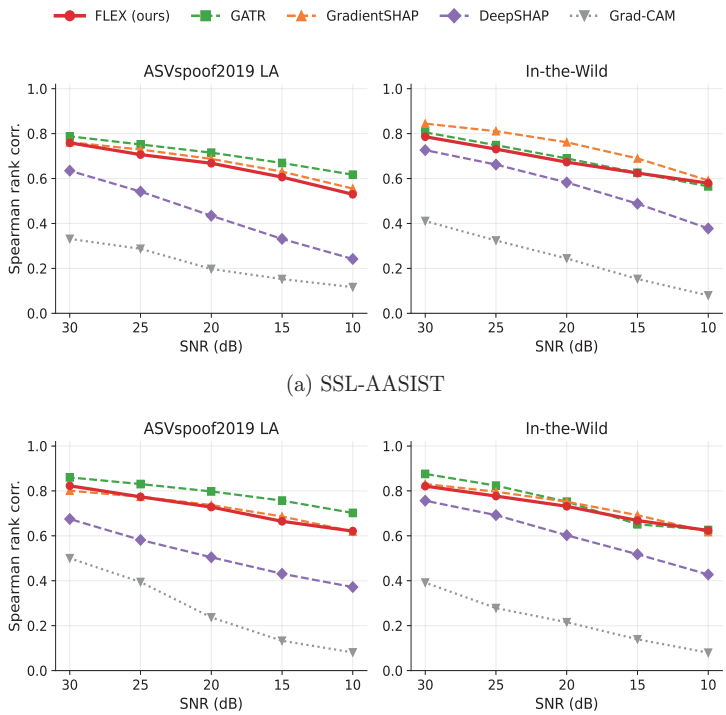
audio segment with a stride of 20 ms. For fair comparison, all baseline explanation methods were evaluated using the same protocol and parameter settings as [8].

Rather than replacing the selected regions with zeros, we mask them using Gaussian noise with zero mean and variance matched to that of the input waveform amplitudes. This strategy reduces out-of-domain artifacts introduced by zero masking and better reflects the noise conditions encountered during training. We vary the masking ratio from 10% to 90% and summarize the resulting performance degradation using the area under the EER-versus-masking-ratio curve (AUC-EER). A larger AUC-EER indicates that removing highly ranked frames causes a greater increase in detection error, suggesting that the explanation more effectively identifies decision-critical frames.

As shown in Table 2, FLEX consistently demonstrates strong faithfulness across both datasets. On the in-domain ASV19 dataset, FLEX attains the highest AUC-EER among all compared methods, indicating that the proposed alignment score effectively prioritizes decision-critical frames. On the out-of-domain ITW dataset, FLEX remains highly competitive with existing XAI approaches, achieving performance comparable to GradSHAP and outperforming both Grad-CAM and GATR. These results suggest that prototype-based alignment provides faithful explanations that generalize across both in-domain and out-of-domain conditions.

Stability Evaluation While faithfulness measures whether an explanation captures decision-critical evidence, it does not assess explanation consistency under realistic input perturbations [12]. To evaluate explanation stability, we add Gaussian noise at SNR levels from 30 dB to 10 dB in 5 dB decrements and measure the Spearman rank correlation [7] between importance scores of the original and noise-perturbed samples, preserving the predicted label. A higher correlation indicates that the relative ordering of frame importance is preserved under noise.

As shown in Figure 3, all methods exhibit decreasing rank correlation as the noise level increases. Grad-CAM is particularly sensitive to noise, with correlations dropping substantially across both architectures and datasets. In contrast, FLEX maintains relatively high rank correlations across all SNR levels



(a) SSL-AASIST

(b) SSL-Conformer+TCM

Fig. 3. Explanation stability under Gaussian noise. Spearman rank correlation measures the consistency of importance rankings across SNR levels (30dB to 10dB). Higher values indicate better preservation of importance rankings under noise.

and demonstrates competitive stability compared with other explanation methods. These results suggest that prototype-based alignment provides consistent explanations under noisy acoustic conditions. By leveraging class-conditional geometric structure rather than relying solely on local gradient sensitivity, FLEX preserves stable importance rankings under noisy real-world conditions.

3.4 Computational Overhead of Explanations

Beyond faithfulness and stability, explanation latency is also an important practical consideration, as explanation generation can become a computational bottleneck when large volumes of audio recordings must be analyzed and audited.

Table 3 compares the computational cost of generating a frame-importance map. FLEX requires only a single forward propagation and no backpropagation, since importance scores are computed directly from distances between frame

Table 3. Computational cost of producing one frame-importance map. C_f/C_b: computational cost of forward/backward Propagation; N=20: number of SHAP samples/baselines. Latency is measured as the median wall-clock time per explanation over 100 utterances.

Method	Gradient-Free	Propagations	Latency (ms)
DeepSHAP	No	N x $(C_f + C_b)$	420.5
GradientSHAP	No	N x $(C_f + C_b)$	226.1
GATR	No	$C_f + C_b$	71.1
Grad-CAM	No	$C_f + C_b$	67.8
FLEX (Ours)	**Yes**	C_f	**27.1**

Table 4. Early degradation EER (%) under positive (Pos) and negative (Neg) masking across varying prototype counts (K) on the ITW dataset. Results are reported at 10%, 20%, and 30% masking ratios for SSL-Conformer+TCM.

K	**SSL-Conformer+TCM (Baseline EER: 7.85)**							
	K_b	K_s	10%		20%		30%	
			Pos ↑	Neg ↓	Pos ↑	Neg ↓	Pos ↑	Neg ↓
2	1	1	9.74	8.03	**16.58**	**7.89**	**37.71**	**8.88**
4	2	2	9.49	7.85	13.65	7.64	28.75	9.35
8	3	5	8.61	7.77	10.99	7.48	24.29	9.56
16	4	12	8.77	7.78	13.14	7.68	31.81	9.28
32	6	26	9.13	7.75	14.52	7.55	33.79	8.98

embeddings and class prototypes. As a result, FLEX achieves the lowest explanation latency (27.1 ms), reducing explanation time by 2.5× and 15.5× compared with Grad-CAM and DeepSHAP, respectively. These results indicate that FLEX introduces minimal explanation overhead while maintaining strong faithfulness and competitive stability, making it suitable for deployment in real-world ADD systems.

3.5 Analysis Across Prototype Configurations

To analyze the effect of prototype granularity, we vary the number of class-specific prototypes K from 2 to 32 and evaluate positive (Pos) and negative (Neg) masking at 10%, 20%, and 30% ratios. Table 4 reports results on SSL-Conformer+TCM. At $K = 2$, Pos masking at 30% increases the EER to 37.71%, while Neg masking under the same setting remains close to the baseline at 8.88%, confirming that the proposed method effectively identifies decision-relevant frames. This Pos–Neg gap remains consistent across all K, indicating that the effectiveness of FLEX is largely insensitive to prototype granularity.

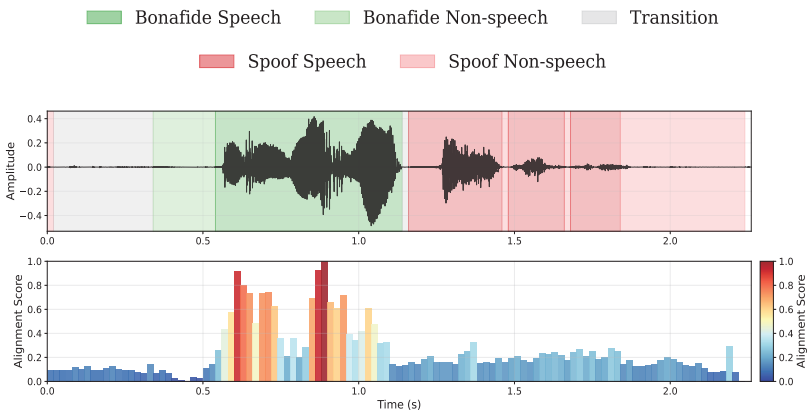


Fig. 4. Alignment Score $w(t)$ and ground-truth labels on a PartialSpoof sample `CON_E_0033629.wav`.

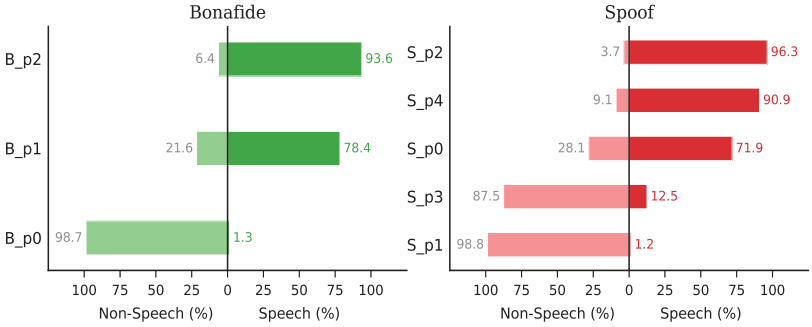


Fig. 5. Distribution of speech and non-speech regions within each class-conditional prototype on the PartialSpoof dataset.

3.6 A Case Study on PartialSpoof

We visualize the behavior of FLEX on the PartialSpoof dataset [20], which provides frame-level annotations of bonafide and spoof segments.

Figure 4 shows the alignment score $w(t)$ together with the waveform and ground-truth frame labels. Since the analyzed utterance was predicted as bonafide, the alignment scores are computed with respect to the bonafide prototype set. Higher values of $w(t)$ indicate stronger alignment with the bonafide class region in the latent space and are primarily observed in bonafide speech regions. The score decreases noticeably in spoofed and transition segments, while bonafide non-speech frames tend to exhibit lower values than bonafide speech frames, reflecting acoustic region-dependent alignment.

Figure 5 analyzes how frames are associated with each prototype based on minimum distance assignment. For each frame, we identify the closest prototype and examine the distribution of annotated speech and non-speech regions within each prototype group. We observe that some bonafide prototypes (e.g., B_p2) are primarily associated with bonafide speech frames, while others (e.g., B_p0) are more frequently linked to bonafide non-speech frames. A similar pattern is observed for spoof prototypes, suggesting that frames associated with the same prototype share consistent acoustic characteristics. These findings indicate that individual prototypes capture meaningful acoustic patterns, providing an interpretable basis for understanding model decisions at the frame level — a property directly useful for forensic analysts seeking to localize and examine manipulated audio segments.

4 Conclusion

As audio deepfakes pose growing threats to digital security and forensic integrity, interpretable detection systems are essential for trustworthy deployment. This paper presents FLEX, a post-hoc framework for frame-level explanations in audio deepfake detection based on latent decision geometry. By quantifying the alignment between frame-level representations and class-conditional prototypes, FLEX provides interpretable explanations without modifying or retraining the underlying ADD model.

Experimental results demonstrate that FLEX achieves strong faithfulness, competitive stability under noisy conditions, and low computational overhead. Furthermore, our case study suggests that FLEX captures meaningful acoustic patterns, such as speech and non-speech regions, and can provide insight into which acoustic regions contribute most strongly to the model's decisions. Together, these findings demonstrate that latent decision geometry provides a practical foundation for trustworthy audio deepfake detection. More importantly, FLEX shows that meaningful prototype-based explanations can be constructed post-hoc from frozen ADD models, suggesting that the latent representations learned for detection inherently contain structures that can be leveraged for explanation without additional training or model modification.

Acknowledgments. This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center)(IITP-2026-RS-2020-II201602, 50%) grant funded by the Korea government(Ministry of Science and ICT) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00457415, 50%).

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems **33**, 12449–12460 (2020)

2. Chefer, H., Gur, S., Wolf, L.: Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 397–406 (2021)
3. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: deep learning for interpretable image recognition. Advances in neural information processing systems **32** (2019)
4. Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al.: Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing **16**(6), 1505–1518 (2022)
5. Doan, T.P., Dinh-Xuan, H., Ryu, T., Kim, I., Lee, W., Hong, K., Jung, S.: Trident of poseidon: A generalized approach for detecting deepfake voices. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. pp. 2222–2235 (2024)
6. Ge, W., Patino, J., Todisco, M., Evans, N.: Explaining deep learning models for spoofing and deepfake detection with shapley additive explanations. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 6387–6391. IEEE (2022)
7. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3681–3688 (2019)
8. Grinberg, P., Kumar, A., Koppisetti, S., Bharaj, G.: What does an audio deepfake detector focus on? a study in the time domain. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
9. Liu, T., Zhang, L., Das, R.K., Ma, Y., Tao, R., Li, H.: How do neural spoofing countermeasures detect partially spoofed audio? arXiv preprint arXiv:2406.02483 (2024)
10. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
11. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
12. Monke, H., Sae-Chew, B., Fresz, B., Huber, M.F.: From confusion to clarity: Protoscore-a framework for evaluating prototype-based xai. In: Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency. pp. 2215–2231 (2025)
13. Müller, N.M., Czempin, P., Dieckmann, F., Froghyar, A., Böttinger, K.: Does audio deepfake detection generalize? arXiv preprint arXiv:2203.16263 (2022)
14. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. International journal of computer vision **128**(2), 336–359 (2020)
15. 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)
16. Truong, D.T., Tao, R., Nguyen, T., Luong, H.T., Lee, K.A., Chng, E.S.: Temporal-channel modeling in multi-head self-attention for synthetic speech detection. arXiv preprint arXiv:2406.17376 (2024)
17. Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., Sahidullah, M., Vestman, V., Kinnunen, T., Lee, K.A., et al.: Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech. Computer Speech & Language **64**, 101114 (2020)

18. Xiao, Y., Das, R.K.: Xlsr-mamba: A dual-column bidirectional state space model for spoofing attack detection. IEEE Signal Processing Letters (2025)
19. Xue, M., Huang, Q., Zhang, H., Cheng, L., Song, J., Wu, M., Song, M.: Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition. arXiv preprint arXiv:2208.10431 (2022)
20. Zhang, L., Wang, X., Cooper, E., Evans, N., Yamagishi, J.: The partialspoof database and countermeasures for the detection of short fake speech segments embedded in an utterance. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 813–825 (2022)

Towards Robust Audio Deepfake Detection: Consistency and Separability Analysis of Bonafide Acoustic Cues

Seonghak Lee[1][0009−0006−7079−5066] and Souhwan Jung[2][0000−0003−2676−3412]

Soongsil University, Seoul, Republic of Korea[1,2]
seonghak@soongsil.ac.kr[1]
souhwanj@ssu.ac.kr[2]

Abstract. Existing Audio Deepfake Detection (ADD) systems frequently rely on dataset or synthesis-specific cues, limiting their ability to generalize across unseen speech generation methods and recording conditions. This study systematically analyzes handcrafted acoustic cue candidates that are likely to appear relatively consistently across various bonafide domains, rather than relying directly on spoof artifacts. To this end, a total of 105 acoustic features were analyzed across 11 groups from a source-filter perspective, and it was found that 31 features satisfied bonafide consistency and 30 features satisfied spoof consistency. However, only 18 features satisfied both consistency criteria, and none satisfied the final separability criterion. Detector-level experiments also showed high false positive rates under cross-domain conditions. These results indicate that handcrafted acoustic cues are useful for diagnostic analysis of consistency and separability in ADD, but are insufficient as standalone detector features. This study presents the first systematic analysis of the consistency-separability trade-off among diverse handcrafted acoustic cue candidates under cross-domain ADD conditions, and provides diagnostic evidence for future self-supervised representation probing.

Keywords: Audio Deepfake Detection · Handcrafted Acoustic Features · AI Security

1 Introduction

Recent advances in text-to-speech (TTS) and voice conversion (VC) make synthetic speech increasingly difficult to distinguish from natural speech. Audio Deepfake Detection (ADD) has therefore become important for speech authentication, media verification, and online content reliability. Benchmark resources such as ASVspoof [1] and models including AASIST [2] and SSL-based systems [3] have achieved strong benchmark performance.

However, benchmark performance does not always generalize to unseen synthesis methods or real-world recording environments [4, 5]. Models may obtain low errors on training domain but fail under unseen synthesis methods, codec effects, in-the-wild recordings, or emotional speech. This suggests that ADD

models may rely on system-specific artifacts rather than general distinctions between bonafide and spoof speech.

Many existing approaches focus on artifacts remaining in spoof speech. Such artifacts can reflect distortions from specific vocoders, acoustic models, or post-processing steps, but they can also change with synthesis technology, recording conditions and transmission processes. Therefore, high bonafide-spoof separability alone is not sufficient evidence of a general acoustic cue. The source of the difference must also be examined.

To address this issue, this study analyzes handcrafted acoustic cue candidates that may remain relatively consistent across diverse bonafide domains, rather than relying directly on spoof artifacts. This analysis examines whether bonafide-spoof differences are associated with domain-consistent acoustic cues or with domain-dependent artifacts. We construct 105 handcrafted acoustic features across 11 source-filter-based groups and evaluate them along three axes: bonafide-domain consistency, spoof-domain consistency, and bonafide-spoof separability. A detector-level cross-domain experiment is also conducted to examine whether feature-level findings translate into detector-level behavior.

The main contributions are as follows:

- First, this paper constructs and organizes 105 handcrafted acoustic cue candidates for ADD into 11 source-filter-based feature groups and presents a cross-domain analysis framework.
- Second, it separately measures bonafide consistency, spoof consistency, and bonafide-spoof separability, thereby characterizing the consistency-separability trade-off at the handcrafted cue level.
- Third, detector-level experiments show that feature-level consistency does not necessarily lead to low false-positive rates, clarifying the practical limitations of handcrafted acoustic cues.

These results motivate future work on probing how SSL representations encode specific acoustic cues.

2 Methodology

2.1 Acoustic Feature Taxonomy

Handcrafted acoustic cue candidates are grouped according to speech production principles and signal characteristics, rather than being treated only as independent scalar features. This taxonomy is intended to make each feature interpretable in terms of the acoustic phenomenon it measures. Group-level organization also helps examine whether high separability reflects a shared speech-production-related acoustic difference or a dataset or recording-dependent effect.

The feature groups are designed from the source-filter view of speech production, where speech is described in terms of a voice source and a vocal-tract filter [6]. Based on this view, G1–G2 capture source-related and waveform-level characteristics, G3–G6 describe spectral structure, G7–G8 measure temporal dynamics, and G9 represents aperiodicity and voice quality. G10–G11 are included

Table 1. Feature group definitions based on the source-filter (S-F) model of speech production. Total 105 features across 11 groups.

ID	Group Name	#	S-F Component	Representative Features
G1	LPC Residual	5	Source (glottal)	LPC prediction error, skewness, kurtosis, entropy, RMS
G2	Waveform HOS	19	Source (glottal)	Waveform skewness/kurtosis/crest, zero-crossing rate, voiced-frame statistics
G3	Spectral Flatness	7	Vocal tract	STFT Wiener entropy (mean, std, min, max, median, Q25, Q75)
G4	CQT Flatness	7	Vocal tract	CQT-domain Wiener entropy (same statistics as G3)
G5	Spectral Shape	14	Vocal tract	Spectral centroid, rolloff, crest factor, spectral irregularity
G6	Spectral Contrast	6	Vocal tract	Sub-band spectral contrast (6 frequency bands)
G7	Subband Dynamics	24	Vocal tract	Spectral flux per sub-band (low/mid/high) and sub-band energy ratio
G8	Amplitude Modulation	6	Source (glottal)	AM envelope autocorrelation at 1, 2, 4, 8 Hz (syllabic rhythm)
G9	Aperiodicity	7	Source (glottal)	Jitter (local, PPQ5), shimmer (local, APQ3, APQ5), CPP mean/std
G10	Conventional	8	Environment	Energy (dB), amplitude range, voiced ratio, voiced run entropy
G11	Prosody Macro	2	Supra-segmental	F0 declination slope, F0 rise fraction

as negative-control groups because they may be sensitive to recording conditions, voicing structure, speech content, or macro-prosodic variation.

G1 provides a baseline for evaluating traditional source-related cues under cross-domain conditions by using linear predictive coding (LPC) residual statistics. G2 measures waveform higher-order statistics directly from the time-domain signal without LPC inverse filtering. G3 and G4 are spectral flatness feature groups based on short-time Fourier transform (STFT) and constant-Q transform (CQT) respectively. CQT-based representations have also been used in spoofing countermeasures [7]. G5 describes spectral shape, while G6 measures band-wise spectral contrast. G7 captures subband energy dynamics, and G8 measures amplitude modulation patterns related to temporal energy movement. G9 includes voice-quality and aperiodicity measures, such as jitter, shimmer, and Cepstral Peak Prominence (CPP), which are commonly used in acoustic voice analysis [8]. Finally, G10 includes conventional energy and voicing statistics, while G11 includes macro-prosodic features, as summarized in Table 1.

These groups are not assumed to be robust cues. Instead, they define a structured set of acoustic cue candidates to be evaluated using domain consistency and bonafide-spoof separability criteria. Therefore, this taxonomy is used as an analysis unit, not as a performance-driven feature selection procedure.

Table 2. Dataset domains used in the main analysis. Each domain was balanced to 2,000 utterances.

ID	Dataset	Condition	Main variation / system factor	Role
Bonafide domains				
B1	ASVspoof 2019 LA train [1]	Clean studio	English neutral speech	Bonafide baseline
B2	VCTK [9]	Clean multi-speaker studio	Speaker and accent variation	Speaker diversity
B3	ITW bonafide [4]	In-the-wild online media	Real-world channel variation	Real-world bonafide
B4	ESD + CREMA-D [10, 11]	Controlled studio	Emotional speech	Emotion variation
B5	FLEURS subset [12]	Online speech corpus	Multilingual speech	Language variation
Spoof domains				
S1	ASVspoof 2019 LA train	Clean studio	Early TTS/VC attacks	Source-domain spoof
S2	ASVspoof 2021 DF [5]	Transmission processed	Neural vocoder attacks with channel effects	Spoof consistency
S3	ITW spoof	In-the-wild online media	Unknown modern TTS in real-world media	Real-world spoof
S4	EmoFake spoof [13]	Studio emotional TTS	Emotional TTS with controlled scripts	Emotion-controlled spoof
S5	MLAAD v8 [14]	Controlled multi-TTS	Multilingual multi-system neural TTS	Multi-system spoof
S6	WaveFake [15]	Controlled corpus synthesis	LJSpeech-based neural vocoder/TTS systems	Controlled vocoder spoof

2.2 Evaluation Protocol

The evaluation protocol is designed to evaluate whether each acoustic cue candidate simultaneously satisfies cross-domain consistency and bonafide-spoof separability. Each feature is evaluated along three axes: bonafide-domain consistency, spoof-domain consistency, and bonafide-spoof separability.

Domain consistency is evaluated by measuring how strongly a feature distribution depends on the domain label. To this end, epsilon-squared ϵ^2, an effect size based on the Kruskal–Wallis test [16] and normalized mutual information (NMI) [17] are used together. The Kruskal–Wallis test is a non-parametric rank-based test that does not assume Gaussian feature distributions. Because p-values can be misleading with large sample sizes, ϵ^2 is used to quantify the magnitude of the domain effect, while NMI measures how much domain identity is contained in each feature.

The Kruskal–Wallis effect size is computed as

$$\epsilon^2 = \frac{H_{\mathrm{KW}} - k + 1}{N - k}, \tag{1}$$

where H_{KW} denotes the Kruskal–Wallis statistic, k denotes the number of domains, and N denotes the total number of samples. NMI is computed as

$$\mathrm{NMI}(X; D) = \frac{2I(X; D)}{\mathcal{H}(X) + \mathcal{H}(D)}, \tag{2}$$

where X denotes the feature values, D denotes the domain labels, $I(X;D)$ denotes mutual information, and $\mathcal{H}(\cdot)$ denotes entropy.

Bonafide-domain consistency is evaluated using the B1–B5 bonafide domains listed in Table 2. Spoof-domain consistency is evaluated using the S1–S6 spoof domains listed in Table 2. In both cases, a feature is considered consistent when it satisfies $\epsilon^2 < 0.14$ and NMI < 0.15 [18]. These criteria exclude features whose distributions are strongly affected by bonafide-domain factors or by a specific synthesis system or spoof dataset.

Bonafide-spoof separability evaluates whether each feature can distinguish bonafide from spoof speech. Separability is measured using the area under the receiver operating characteristic curve (AUC) [19] and Cohen's d [20]. AUC measures threshold-independent ranking performance, while Cohen's d measures the standardized mean difference between bonafide and spoof feature distributions. Therefore, AUC captures separability across thresholds, whereas Cohen's d captures the magnitude of the class difference. Since the class associated with larger feature values can differ across features, AUC is computed in a direction-independent manner:

$$\mathrm{AUC}_{\mathrm{DI}} = \max\left(\mathrm{AUC}_{\mathrm{raw}}, 1 - \mathrm{AUC}_{\mathrm{raw}}\right), \tag{3}$$

where $\mathrm{AUC}_{\mathrm{raw}}$ is computed by treating spoof speech as the positive class. Cohen's d is computed as

$$d = \frac{\mu_{\mathrm{bonafide}} - \mu_{\mathrm{spoof}}}{s_p}, \tag{4}$$

where s_p denotes the pooled standard deviation:

$$s_p = \sqrt{\frac{(n_{\mathrm{bonafide}} - 1)s^2_{\mathrm{bonafide}} + (n_{\mathrm{spoof}} - 1)s^2_{\mathrm{spoof}}}{n_{\mathrm{bonafide}} + n_{\mathrm{spoof}} - 2}}. \tag{5}$$

Here, μ, s, and n denote the mean, standard deviation, and number of samples for each class. A feature satisfies bonafide-spoof separability when $\mathrm{AUC}_{\mathrm{DI}} \geq 0.70$ and $|d| \geq 0.8$ [19, 20]. A robust acoustic cue candidate must satisfy bonafide consistency, spoof consistency, and bonafide-spoof separability simultaneously.

3 Experiments

3.1 Experimental Setup

Table 2 summarizes the bonafide and spoof domains used in the main analysis. Each domain was balanced to 2,000 utterances to reduce sample-size bias. From each utterance, 105 handcrafted acoustic features were extracted from the 11 feature groups in Table 1 and evaluated using the protocol in Section 2.2. For detector-level evaluation, GMM LLR [21], SVM RBF [22], and LightGBM [23] were trained on B1 and S1, and tested on unseen B2–B5 bonafide domains and S2–S6 spoof domains. Performance was measured using AUC and FPR@TPR=0.95 [24, 19, 25].

Table 3. Step-by-step filtering results for 105 acoustic feature candidates. Pass count indicates the number of features that satisfy each individual step, while remaining count indicates the number of features that satisfy all criteria up to that step.

Step	Filter	Pass	Remaining
Step 1	Bonafide-domain consistency across B1–B5	31	31
Step 2	Spoof-domain consistency across S1–S6	30	18
Step 3	Bonafide–spoof separability	6	0

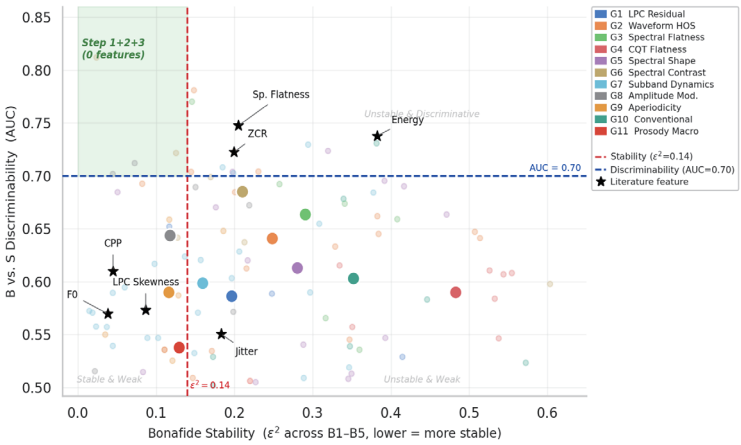


Fig. 1. Feature-level consistency–separability map. The x-axis shows bonafide-domain stability measured by ϵ^2, and the y-axis shows direction-independent bonafide-spoof AUC. Dashed lines indicate the stability and discriminability thresholds. Star markers indicate commonly cited acoustic features.

3.2 Feature-level Analysis

Bonafide-domain consistency. Table 3 summarizes the step-by-step filtering results. Among the 105 features, 31 satisfied bonafide-domain consistency. This indicates that some handcrafted features maintain relatively stable distributions across bonafide domains.

Spoof-domain consistency. The spoof-domain consistency criterion was satisfied by 30 features. However, only 18 features satisfied both bonafide-domain and spoof-domain consistency criteria. This shows that features stable in bonafide domains are not necessarily stable across spoof domains.

Bonafide-spoof separability. None of the 105 features satisfied the final separability criterion together with both consistency criteria. This result shows that single handcrafted features rarely satisfy both cross-domain consistency and bonafide-spoof separability.

Fig. 1 visualizes this feature-level trade-off. Features in the upper-left region would be both stable and discriminative, but no feature satisfies all consistency

Table 4. Final label distribution across 105 acoustic features. BF-Artifact denotes bonafide recording/artifact-sensitive features.

Label	Count	Criterion	Dominant groups
Robust	0	Passed consistency and separability criteria	—
Stable-Weak	0	Passed consistency and weak separability, but failed FPR criterion	—
Non-sep.	18	Passed both consistency criteria, but AUC < 0.70 or $\lvert d\rvert < 0.8$	G1, G8, G9, G11
Spoof-spec.	13	Passed bonafide consistency, but failed spoof consistency	G6, G7
BF-Artifact	74	Failed bonafide consistency: $\varepsilon^2 \geq 0.14$ or NMI ≥ 0.15	G2, G3, G5, G10
Total	**105**		

Table 5. Exp4: Cross-domain classification results. Train: B1+S1 (ASVspoof 2019 LA train, single controlled domain pair). Test: B2–B5 + S2–S6 (unseen bonafide/spoof domains). Feature subsets selected by: (A) all, (B) Step 1+2 stability ($\epsilon^2 < 0.14$ and NMI < 0.15), (C) greedy forward selection from B, (D) Step 3 separability ($\lvert d\rvert \geq 0.8$, AUC ≥ 0.70), (E) commonly cited ADD features (Jitter excluded). AUC ↑ and FPR@TPR= 0.95 ↓ are reported.

			GMM LLR		**SVM RBF**		**LightGBM**	
ID	**Subset**	N	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓
A	All	105	0.493	0.944	0.605	0.849	**0.808**	**0.699**
B	Stable-18	18	0.616	0.915	0.680	0.826	0.661	0.818
C	Greedy-12	12	0.671	0.897	0.692	0.813	0.691	0.807
D	Discrim-6	6	**0.802**	**0.773**	**0.799**	0.902	0.792	0.832
E	Literature-7	7	0.710	0.800	0.733	**0.791**	0.757	0.821

and separability criteria. CPP and LPC skewness remain relatively stable but weakly separable, whereas spectral flatness, zero-crossing rate, and energy show higher separability but fall outside the stable region.

Among the 18 stable features, `am_autocorr_8hz` achieved the highest single-feature AUC of 0.702, but its Cohen's d was 0.726 and did not satisfy the separability criterion. In contrast, `sp_contrast_b6` achieved a higher AUC of 0.812 and |d| = 1.128, but failed the spoof-domain consistency criterion. Other high-separability features showed the same pattern: they met the separability criterion but failed at least one consistency criterion. Detailed per-feature metrics are provided in Appendix and Table 6.

Table 4 summarizes the final feature-label distribution. No feature was classified as robust, while most features were sensitive to bonafide-domain variation or recording-related factors.

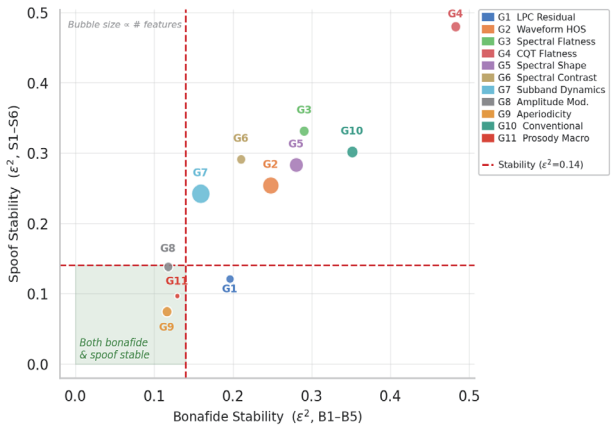


Fig. 2. Group-level bonafide and spoof stability. The x-axis shows bonafide-domain effect size across B1–B5, and the y-axis shows spoof-domain effect size across S1–S6. Bubble size is proportional to the number of features in each group. The lower-left region indicates groups that are stable in both bonafide and spoof domains.

3.3 Group-level Analysis

Bonafide-domain consistency and spoof-domain consistency. To interpret group-level behavior, this section summarizes the relationship between domain consistency and bonafide-spoof separability across the 11 feature groups. Fig. 2 summarizes this pattern at the feature-group level. Groups closer to the lower-left region have lower domain effects in both bonafide and spoof domains, whereas groups farther from this region are more domain-sensitive.

Bonafide-spoof separability. The consistency–separability trade-off was also evident at the group level. G2, G5, G6, G7, and G10 showed relatively high separability, but they also showed large bonafide-domain or spoof-domain effect sizes. Thus, high separability may reflect recording conditions, dataset composition, or synthesis systems rather than a domain-general acoustic cue. When the stability pattern in Fig. 2 is compared with the group-level separability results, the groups with higher separability are generally not the most stable groups.

In contrast, G8, G9, and G11 showed relatively low domain effect sizes but limited separability. The spectral flatness groups showed a similar limitation. CQT-based representations have been used in spoofing countermeasures [7]. However, in our analysis, the CQT-based flatness group was highly sensitive to both bonafide and spoof domains. G10 also showed high separability together with high domain effect sizes, suggesting that conventional energy and voicing features may reflect recording or dataset construction effects.

Detector-level evaluation. Table 5 reports detector-level cross-domain results for different feature subsets and classifiers. Although some subsets achieved relatively high AUC values, all settings still showed high FPR@TPR=0.95. The lowest FPR was 0.699 with LightGBM using all 105 features. This indicates

that feature-level separability does not necessarily lead to a low false-positive rate under cross-domain conditions.

As a result, group-level and subset-level results are consistent with the single-feature analysis. Groups with high separability were generally domain-sensitive, while groups with high consistency had limited separability. In addition, the detector-level subset results show that handcrafted feature subsets alone do not guarantee a robust detector for cross-domain ADD.

4 Conclusion

The results show a clear consistency–separability trade-off. Features that were stable across both bonafide and spoof domains generally showed limited bonafide-spoof separability, whereas high-AUC features often failed one or more domain consistency criteria. Therefore, high bonafide-spoof separability alone is not sufficient evidence of a domain-general acoustic cue. Although no handcrafted feature satisfied all strict criteria, three features were selected as compromise candidates for future work under relaxed criteria: `wav_frame_kurtosis_mean`, `sp_flatness_std`, and `am_autocorr_8hz`. The first two features did not fully satisfy the strict consistency criteria but showed relatively high separability, while `am_autocorr_8hz` showed the highest single-feature AUC among the strictly stable features.

These findings suggest that handcrafted acoustic cues are more useful as diagnostic tools than as standalone detector inputs because each feature is linked to a relatively interpretable acoustic phenomenon. For example, `wav_frame_kurtosis_mean` reflects waveform temporal impulsiveness, `sp_flatness_std` reflects temporal variability of tonal and noise-like spectral structure, and `am_autocorr_8hz` reflects syllabic-rate amplitude modulation. Therefore, even when these cues do not provide robust detection performance by themselves, they can help identify which acoustic factors remain relatively domain-stable and which factors achieve separability only under relaxed or domain-sensitive conditions.

Future work should examine how these three compromise features can be used in SSL-based ADD training. Rather than using them as primary classifiers, they can be used as auxiliary supervision signals for wav2vec 2.0 or XLS-R backbones to regularize the learned representation with supplementary naturalness-related cues [3]. Although this study is limited to predefined handcrafted features and cannot fully explain nonlinear cues learned by deep representations, it provides evidence that acoustic cues for cross-domain ADD should be evaluated not only by separability, but also by domain consistency.

Appendix

Additional Feature-level Analysis Results

Table 6 provides feature-level metrics for the subsets used in the classification experiment, including specific features cited in Section 3.2. Readers who wish to verify individual feature statistics can refer to this table.

Table 6. Feature-level analysis metrics for subsets used in the classification experiment. ε^2(B): KW ε^2 across B1–B5 (bonafide stability; pass if <0.14). ε^2(S): KW ε^2 across S1–S6 (spoof stability; pass if <0.14). Stability pass/fail is encoded by the S18 and G12 checkmark columns. AUC: direction-free bonafide vs. spoof discriminability. $|d|$: absolute Cohen's d. **Bold**: each metric independently bolded when AUC ≥ 0.70 or $|d| \geq 0.8$ (separability criterion). **S18** = Stable-18 (Step 1+2 pass); **G12** = Greedy-12 (subset of S18 selected by greedy forward selection; features added one by one to maximise cross-domain AUC, stopped at 12); **D6** = High-separability-6 (Step 3 pass); **L7** = Literature-7 (features cited in prior ADD literature).

Feature	**Grp**	ε^2(B)	ε^2(S)	**AUC**	$\lvert d\rvert$	**S18**	**G12**	**D6**	**L7**
Greedy-12 (G12 $\subseteq$ S18; greedy selection order)									
am_autocorr_8hz	G8	0.044	0.136	**0.702**	0.726	✓	✓	—	—
cpp_mean	G9	0.045	0.010	0.610	0.400	✓	✓	—	✓
shimmer_apq5	G9	0.117	0.092	0.659	0.536	✓	✓	—	—
shimmer_local	G9	0.128	0.081	0.587	0.237	✓	✓	—	✓
sb_flux_low_std	G7	0.088	0.114	0.547	0.174	✓	✓	—	—
cpp_std	G9	0.035	0.080	0.550	0.201	✓	✓	—	—
f0_decl_slope	G11	0.038	0.109	0.570	0.148	✓	✓	—	✓
spectral_flux_std	G5	0.083	0.102	0.515	0.054	✓	✓	—	—
sb_flux_mid_q75	G7	0.136	0.052	0.624	0.317	✓	✓	—	—
wav_crest_factor	G2	0.110	0.118	0.536	0.106	✓	✓	—	—
lpc_rms	G1	0.115	0.122	0.588	0.024	✓	✓	—	—
am_peak_freq	G8	0.021	0.039	0.516	0.053	✓	✓	—	—
Stable-18 only (S18 $\setminus$ G12)									
jitter_ppq5	G9	0.120	0.077	0.525	0.070	✓	—	—	—
lpc_pred_error	G1	0.116	0.026	0.652	0.023	✓	—	—	—
lpc_skewness	G1	0.087	0.066	0.573	0.009	✓	—	—	✓
sb_flux_low_max	G7	0.102	0.131	0.547	0.169	✓	—	—	—
sb_flux_mid_max	G7	0.044	0.120	0.540	0.079	✓	—	—	—
wav_peak_to_rms	G2	0.110	0.118	0.536	0.106	✓	—	—	—
High-separability-6 (D6; AUC $\geq 0.70 \wedge \lvert d\rvert \geq 0.8$)									
sp_contrast_b6	G6	0.024	0.210	**0.812**	**1.128**	—	—	✓	—
wav_frm_kurt_mean	G2	0.148	0.094	**0.781**	**0.934**	—	—	✓	—
sp_flatness_std	G3	0.146	0.182	**0.770**	**0.829**	—	—	✓	—
zcr_mean	G2	0.199	0.381	**0.722**	**0.820**	—	—	✓	—
wav_zcr	G2	0.199	0.381	**0.722**	**0.820**	—	—	✓	✓
sp_contrast_b5	G6	0.125	0.419	**0.722**	**0.815**	—	—	✓	—
Literature-7 only (L7 $\setminus$ (S18 $\cup$ D6))									
energy_db_min	G10	0.382	0.336	**0.738**	0.302	—	—	—	✓
sp_flatness_min	G3	0.205	0.328	**0.747**	0.623	—	—	—	✓

Acknowledgments. This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program(IITP-2026-RS-2024-00426853, 50%) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00457415, 50%).

References

1. Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., et al.: ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. Computer Speech & Language **64**, 101114 (2020). https://doi.org/10.1016/j.csl.2020.101114
2. Jung, J.W., Heo, H.S., Tak, H., Shim, H.J., Chung, J.S., Lee, B.J., Yu, H.J., Evans, N.: AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In: ICASSP 2022 – 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2405–2409. IEEE, Piscataway (2022). https://doi.org/10.1109/ICASSP43922.2022.9747766
3. Zhang, Q., Wen, S., Hu, T.: Audio deepfake detection with self-supervised XLS-R and SLS classifier. In: Proceedings of the 32nd ACM International Conference on Multimedia, pp. 6765–6773. ACM, New York (2024). https://doi.org/10.1145/3664647.3681345
4. Müller, N.M., Czempin, P., Diekmann, F., Froghyar, A., Böttinger, K.: Does audio deepfake detection generalize? In: Interspeech 2022, pp. 2783–2787. ISCA (2022). https://doi.org/10.21437/Interspeech.2022-108
5. Liu, X., Wang, X., Sahidullah, M., Patino, J., Delgado, H., Kinnunen, T., Todisco, M., Yamagishi, J., Evans, N., Nautsch, A., Lee, K.A.: ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 2507–2522 (2023). https://doi.org/10.1109/TASLP.2023.3285283
6. Tokuda, I.: The source–filter theory of speech. Oxford Research Encyclopedia of Linguistics (2021). https://doi.org/10.1093/acrefore/9780199384655.013.894
7. Todisco, M., Delgado, H., Evans, N.: Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. Computer Speech & Language **45**, 516–535 (2017). https://doi.org/10.1016/j.csl.2017.01.001
8. Brockmann-Bauser, M., de Paula Soares, M.F.: Do we get what we need from clinical acoustic voice measurements? Applied Sciences **13**(2), 941 (2023). https://doi.org/10.3390/app13020941
9. Yamagishi, J., Veaux, C., MacDonald, K.: CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92). University of Edinburgh, The Centre for Speech Technology Research (CSTR) (2019). https://doi.org/10.7488/ds/2645
10. Zhou, K., Sisman, B., Liu, R., Li, H.: Emotional voice conversion: Theory, databases and ESD. Speech Communication **137**, 1–18 (2022). https://doi.org/10.1016/j.specom.2021.11.006
11. Cao, H., Cooper, D.G., Keutmann, M.K., Gur, R.C., Nenkova, A., Verma, R.: CREMA-D: Crowd-sourced emotional multimodal actors dataset. IEEE Transactions on Affective Computing **5**(4), 377–390 (2014). https://doi.org/10.1109/TAFFC.2014.2336244

12. Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., Bapna, A.: FLEURS: Few-shot learning evaluation of universal representations of speech. In: 2022 IEEE Spoken Language Technology Workshop (SLT), pp. 798–805. IEEE, Piscataway (2023). https://doi.org/10.1109/SLT54892.2023.10023141
13. Zhao, Y., Yi, J., Tao, J., Wang, C., Dong, Y.: EmoFake: An initial dataset for emotion fake audio detection. In: Proceedings of the 23rd Chinese National Conference on Computational Linguistics, pp. 1286–1297. Chinese Information Processing Society of China, Taiyuan (2024)
14. Müller, N.M., Kawa, P., Choong, W.H., Casanova, E., Gölge, E., Müller, T., Syga, P., Sperl, P., Böttinger, K.: MLAAD: The multi-language audio anti-spoofing dataset. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE, Piscataway (2024). https://doi.org/10.1109/IJCNN60899.2024.10650962
15. Frank, J., Schönherr, L.: WaveFake: A data set to facilitate audio deepfake detection. In: NeurIPS 2021 Track on Datasets and Benchmarks (2021). https://doi.org/10.5281/zenodo.5642694
16. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. Journal of the American Statistical Association **47**(260), 583–621 (1952). https://doi.org/10.1080/01621459.1952.10483441
17. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. Journal of Machine Learning Research **11**, 2837–2854 (2010)
18. Tomczak, M., Tomczak, E.: The need to report effect size estimates revisited. An overview of some recommended measures of effect size. Trends in Sport Sciences **21**(1), 19–25 (2014)
19. Mandrekar, J.N.: Receiver operating characteristic curve in diagnostic test assessment. Journal of Thoracic Oncology **5**(9), 1315–1316 (2010). https://doi.org/10.1097/JTO.0b013e3181ec173d
20. Sullivan, G.M., Feinn, R.: Using effect size—or why the (p) value is not enough. Journal of Graduate Medical Education **4**(3), 279–282 (2012). https://doi.org/10.4300/JGME-D-12-00156.1
21. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digital Signal Processing **10**(1–3), 19–41 (2000). https://doi.org/10.1006/dspr.1999.0361
22. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(3), 27:1–27:27 (2011). https://doi.org/10.1145/1961189.1961199
23. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30, pp. 3146–3154 (2017)
24. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
25. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International Conference on Learning Representations (ICLR) (2017)

Speech Pseudonymization in Korean: Seamless PII Substitution via Localized Speech Editing

Sangjun Park[1](✉), Ngoc Hung Le[1], Kyujin Kim[1], and Souhwan Jung[1]

Soongsil University, Seoul, Korea
alexjuny@soongsil.ac.kr

Abstract. Personally Identifiable Information (PII) contained in voice data presents a critical privacy challenge in the age of widespread internet data sharing. Conventional pseudonymization techniques, such as beep masking and full-utterance Text-to-Speech (TTS) synthesis, fail to preserve data utility as they inherently destroy the original acoustic features, background context, and seamless boundary continuity. To overcome these limitations, we propose a novel speech pseudonymization framework utilizing an integrated Automatic Speech Recognition (ASR), Named Entity Recognition (NER), and Speech Editing (SE) pipeline. Our approach transcribes the audio, detects PII using a robust recognition pipeline, and selectively replaces only the target segments with pseudonyms via a context-aware, Neural Audio Codec (NAC)-based SE model. Furthermore, while the majority of speech de-identification research has predominantly focused on English, leaving morphologically complex languages like Korean relatively underexplored, we specifically optimized each module for the unique linguistic complexities of the Korean language through extensive fine-tuning. Experimental evaluations demonstrate that our proposed framework achieves an optimal balance of superior acoustic quality, seamless naturalness, and computational efficiency.

Keywords: Speech pseudonymization · Speech editing · PII detection

1 Introduction

Voice data is increasingly uploaded to cloud services, shared through online platforms, and reused for training speech technologies. Unlike ordinary text, speech contains both linguistic content and rich acoustic cues, including speaker identity, channel conditions, emotion, age, and background environment. If raw speech is released without protection, these cues can expose private information and allow sensitive recordings to be linked back to individuals. This risk is particularly important for Korean, an underrepresented language in speech privacy research despite its substantial use in real-world services such as call centers, mobile applications, and public audio archives.

Speech privacy threats stem from both the speaker's voice characteristics and the sensitive lexical content spoken, but since voice anonymization alone [1] cannot prevent the exposure of personally identifiable information (PII), robust

content protection is essential. However, existing content anonymization methods have significant limitations: simple operations like beeping [3] or noising [4] disrupt acoustic continuity and degrade data usefulness, whereas more advanced Automatic Speech Recognition (ASR), Named Entity Recognition (NER), and Text-to-Speech (TTS) pipelines [2] completely discard the original waveform, stripping away valuable speaker timbre, prosody, and background context that are crucial for downstream speech research.

Furthermore, speech pseudonymization in Korean remains relatively underexplored. Because existing pipelines are primarily English-centric [4,5], directly applying them often leads to inaccurate PII detection due to Korean-specific linguistic traits, such as particles and agglutination [6]. Additionally, the strict privacy regulations (e.g., Personal Information Protection Act in South Korea) and lack of publicly available Korean speech datasets containing annotated PII pose a significant challenge for research and evaluation. Therefore, building a robust pipeline requires language-aware detection models, combined with advanced audio editing techniques that preserve the original acoustic scene.

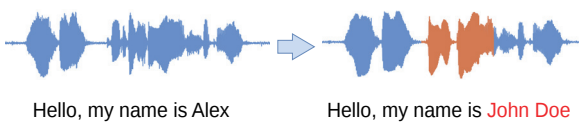


Fig. 1. An example of speech pseudonymization via speech editing.

To address this gap, we propose an integrated speech pseudonymization framework that achieves seamless PII substitution via neural speech editing. Built upon an ASR, NER, and Speech Editing (SE) pipeline, our approach emphasizes both accurate detection and plausible replacement. After transcribing the audio and detecting PII with a Korean-optimized NER module, the system substitutes the sensitive text with rule-based pseudonyms. Crucially, a context-aware Neural Audio Codec (NAC)-based SE model, leveraging SSR-Speech [7], then modifies only these targeted regions. By focusing on locality—editing solely the sensitive segments while leaving the rest of the utterance untouched—this framework provides a highly effective mechanism for localized PII substitution, demonstrating the feasibility of seamless text-guided audio editing while preserving the naturalness and utility of the original speech data. Figure 1 illustrates a conceptual example of this substitution process.

The main contributions of this paper are summarized as follows:

- **First Korean Speech Pseudonymization Framework:** We introduce, to the best of our knowledge, the first comprehensive framework for Korean speech pseudonymization, addressing an underrepresented language in speech privacy research.
- **Speech Editing for PII Replacement:** Instead of relying on disruptive masking or heavy full-utterance resynthesis, we utilize a (NAC)-based SE mechanism for PII replacement. Guided by an optimized secondary align-

ment module for precise temporal localization, our approach generates highly natural pseudonym segments with rapid inference capability (0.16 RTF).
- **Language-Aware ASR-NER-SE Pipeline:** We combine Korean-optimized ASR, NER, and SE models that operates in a speech-to-speech paradigm. The framework accepts raw audio inputs, seamlessly detects and processes sensitive entities, and directly outputs pseudonymized speech while robustly handling linguistically complex structures.
- **Evaluation with a Synthetic Korean PII Dataset:** Due to the scarcity of publicly available Korean speech datasets with annotated PII, we construct a synthetic evaluation dataset using Large Language Model (LLM)-generated PII-rich transcripts and TTS-generated audio, and show that the proposed SE-based method outperforms conventional pseudonymization baselines in quality, continuity, and efficiency.

2 Related Work

2.1 Speaker and Content Privacy in Speech

Speech privacy protection traditionally encompasses two distinct areas: speaker anonymization, which conceals identity via voice conversion [9,10] or signal processing [8], and content anonymization(pseudonymization) [2,3,4], aimed at hiding spoken PII. While speaker anonymization mitigates biometric identification risks, it leaves sensitive lexical information fully exposed. To address content privacy, early handcrafted methods—such as masking PII with silence, noise, or beep sounds—introduce audible discontinuities that severely degrade data utility. Consequently, an ideal approach must obscure sensitive content while seamlessly preserving the surrounding non-sensitive acoustic features.

2.2 ASR-NER-TTS Pipelines for Speech De-identification

A common strategy for content-aware speech de-identification is the ASR-NER-TTS pipeline [2], where speech is transcribed, sensitive entities are detected, and a new utterance is synthesized. While this approach can cleanly replaces PII at the text level and preserve speaker identity, full-utterance resynthesis introduces a major trade-off. As previously noted, discarding the original waveform strips away acoustic cues—such as emotional prosody and background environment—that are not captured in the transcript. This inherent limitation highlights the need for a robust speech pseudonymization system for Korean that features both accurate PII detection and a localized replacement mechanism designed to seamlessly edit only the sensitive segments while fully preserving the naturalness and utility of the original audio.

2.3 Localized Neural Speech Editing for PII Replacement

Text-guided speech editing provides a natural mechanism for speech pseudonymization because it modifies only selected regions of an utterance while leaving

the surrounding speech intact. Instead of synthesizing a complete recording from scratch, a speech editing model receives the original audio, the original transcript, and the edited text, then generates replacement audio for the target span. This local editing paradigm directly matches our objective: sensitive words should be replaced with plausible pseudonyms, while non-sensitive speech should remain unchanged.

For PII replacement, however, locality alone is not sufficient. The generated segment must blend with the speaker's voice, prosody, channel condition, and background environment; otherwise, the edit becomes unnatural and the utility of the speech is reduced. Recent neural SE models address this issue by conditioning generation on the acoustic context around the edited region. This makes them more suitable than beep masking, which creates obvious interruptions, and more faithful than TTS resynthesis, which discards the source waveform.

Our framework builds on SSR-Speech, an autoregressive speech editing model that operates in a discrete neural audio codec space. SSR-Speech first compresses the input waveform into discrete audio tokens. The target words are identified by aligning the original transcript with the modified text, and the corresponding token span is masked. A causal transformer then predicts replacement tokens conditioned on the target phoneme sequence and the unmasked acoustic context. Finally, the predicted tokens are combined with the preserved original tokens and decoded back into a waveform.

This design inherently aligns with speech pseudonymization requirements. Because only the masked PII region is regenerated, the model retains the surrounding acoustic evidence to preserve boundary continuity and speaker characteristics.

3 Methodology

Figure 2 presents the proposed Korean speech pseudonymization pipeline. The objective is to replace sensitive words with plausible alternatives while preserving the original acoustic evidence. To achieve this, the pipeline is organized around three tightly connected modules: ASR, NER, and SE. This design reflects the two central requirements of our task: reliable Korean PII detection despite agglutinative morphology, and natural audio replacement without full-utterance resynthesis. Section 3.1 describes PII detection and text substitution, determining *what* should be changed, while Section 3.2 describes speech editing, determining *how* the change is realized in the waveform.

3.1 PII Detection and Text Substitution

The first stage converts raw Korean speech into a pseudonymized text plan. The ASR model produces a transcript and word-level timestamps, which are passed to the NER model to detect sensitive entity spans (e.g., names, phone numbers, addresses). Finally, each detected entity is replaced according to predefined pseudonymization rules.

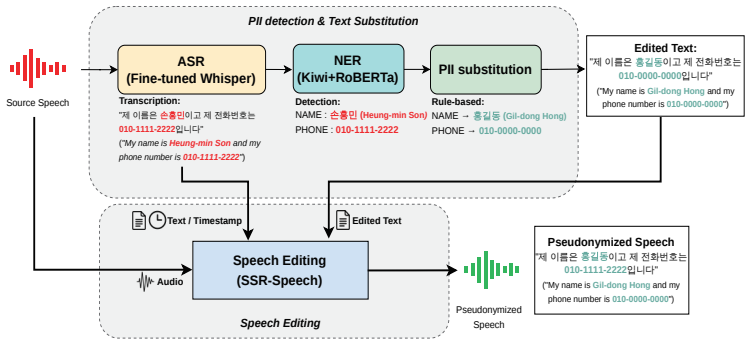


Fig. 2. Overview of the proposed speech pseudonymization system.

ASR Model Accurate transcription is the entry point of the entire pipeline; mistranscriptions directly cause the downstream NER module to fail. We utilize a large-scale transformer-based ASR architecture, specifically a Whisper model [11] fine-tuned primarily to enhance Korean speech recognition performance, while also ensuring robustness across diverse acoustic conditions such as telephone-quality and conversational speech. Beyond transcription accuracy, pseudonymization requires precise temporal alignment. Therefore, we employ WhisperX [12], which integrates a secondary forced-alignment module (e.g., wav2vec2) following the initial transcription stage. This aligns the transcript with accurate word-level start and end times, allowing the detected PII spans to be precisely mapped back to the corresponding audio regions for later masking.

NER Model The NER module identifies PII from the ASR transcript. When selecting the core architecture, we considered both GLiNER [13] and BERT [14]-based models. While open-vocabulary models like GLiNER offer highly flexible entity extraction based on user-specified prompts, our pseudonymization task strictly requires high-precision and high-recall detection for a fixed set of privacy-sensitive entity types. BERT-based models, which formulate NER as a dedicated token classification task, are inherently better suited for this strict requirement. To ensure robust and highly precise entity detection, we adopted *klue/RoBERTa-large* [6], a Korean-specific pre-trained BERT variant and extensively fine-tuned it into a dedicated NER model tailored for our pseudonymization task.

In Korean, core nouns are frequently agglutinated with grammatical particles within a single spacing-delimited unit *eojeol*. If a model relies on naive eojeol-level tokenization, it often struggles to separate the entity from the particle, mistakenly predicting the entire block as PII. To accurately capture the precise entity boundaries, we applied Kiwi [15], a morphological analyzer, as a preprocessing step. By tokenizing the input into fine-grained morphemes prior to training, the NER model successfully isolates the core noun (e.g., Jisung-Park) and assigns non-entity tags to the attached grammatical markers (e.g., *-igoyo* or

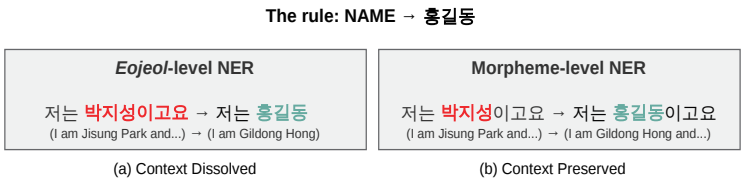


Fig. 3. A comparison of PII text substitution approaches in Korean: (a) Naive *eojeol*-level tokenization results in syntactic disruption by inadvertently discarding grammatical particles (e.g., *-igoyo*). (b) The proposed morpheme-level processing isolates only the target entity, successfully preserving the grammatical context and sentence flow for subsequent speech editing.

-indeyo), completely preventing the destruction of grammatical context during text substitution (see Fig. 3).

Because WhisperX provides timestamps at the eojeol level, the speech editing stage masks the entire word block in the audio. However, our morpheme-aware preprocessing ensures that the text transcript only substitutes the PII token, leaving the grammatical particles untouched. Guided by this accurate transcript, the SE model regenerates both the new pseudonym and the original particle together in the masked region.

PII Pseudonymization Rules After PII spans are detected, each entity is mapped to a pseudonym according to predefined replacement rules. These rules determine the target text that the speech editing model will synthesize. For example, all detected names are replaced with a fixed placeholder such as "Hong Gildong", and phone numbers with "010-1234-5678". Each entity type is mapped to a single consistent pseudonym, ensuring that repeated mentions within an utterance are substituted identically. The rule set is intentionally modular: users can modify the target entity types and replacement values depending on the privacy policy or application scenario. This rule-based substitution ensures the replacement process remains auditable and preserves the syntactic structure of the original utterance, making it highly suitable for text-guided speech editing.

3.2 Speech Editing

The second stage converts the pseudonymized text plan into audio. We adopt SSR-Speech [7], an autoregressive speech editing model operating in the discrete latent space of a neural audio codec. The original waveform is first encoded into discrete audio tokens. Using the PII timestamps obtained from the ASR alignment, tokens corresponding to sensitive words are masked, while the surrounding non-sensitive tokens are retained. The model then generates new tokens for the masked regions, conditioned on both the pseudonymous target text and the preserved acoustic context. Finally, the updated token sequence is decoded back into a continuous speech waveform.

By conditioning on the unmodified left and right context, the replacement segment naturally blends into the original utterance, maintaining speaker timbre, prosody, and background environment. Since only short PII spans are regenerated, this localized approach avoids unnecessary synthesis, maximizing acoustic utility and computational efficiency compared to methods that regenerate entire utterances. Because the original SSR-Speech model was trained primarily on English To further improve naturalness for Korean phonetics and prosody, we fine-tuned SSR-Speech on Korean conversational speech data.

4 Experiment Setup

4.1 Datasets

To train and evaluate the components of our framework, we utilized multiple specialized datasets:

- **ASR:** We fine-tuned the ASR model on approximately 6,000 hours of Korean speech data distributed via AIHUB. This mixture includes *Ksponspeech* [18] (conversational), *KtelSpeech* [19] (counseling/telephone), *youth* [29] and *adult* [30] (age-dependent), and *zeroth* [20] (clean read speech), ensuring robustness across diverse real-world conditions. For evaluation, we used the official test splits provided by each corpus, ensuring no overlap between training and evaluation data.
- **NER:** We used KDPII [17], a Korean PII dataset annotated with 33 entity types. The text was preprocessed using Kiwi [15], a Korean morphological analyzer, to extract morpheme-level units for precise boundary detection. The dataset was split into training and test sets at a 9:1 ratio for fine-tuning and evaluation.
- **Speech Editing (Training):** We fine-tuned SSR-Speech on Korean speech data to improve naturalness for Korean phonetics and prosody. Specifically, we used the *youth* [29] and *adult* [30] datasets from AIHUB, which were also used for ASR fine-tuning.
- **Speech Editing (Evaluation):** Due to privacy restrictions on releasing real PII audio, we constructed a synthetic dataset. We used the Gemma LLM [21] to generate transcripts containing NAME, PHONE, ADDRESS, and BIRTHDATE entities, and synthesized them into approximately 300 utterances using Qwen3-TTS [22].

4.2 Baselines

We evaluated our proposed models and the overall pipeline at three levels against representative open-source baselines:

- **ASR:** Whisper-large-v3 [11], Qwen3-ASR [23], Dolphin ASR [24], wav2vec2-large-xlsr-korean [25], and our proposed FT-Whisper.
- **NER:** GLiNER-based open-vocabulary models [13] (multi-v2.1, multi_pii-v1, ko, finetuned), baseline BERT-based models (ModernBERT [26], mDeBERTa [27], xlm-RoBERTa [16]), and our fine-tuned klue/RoBERTa [6].

– **Speech Substitution:** Original audio, beep masking [3], TTS-based splicing using Qwen3-TTS [22], and our proposed SE-based pseudonymization.

4.3 Evaluation Metrics

For ASR, we report Character Error Rate (CER), which is more stable for agglutinative languages than Word Error Rate (WER). For NER, we use Precision, Recall, and F1-Score, emphasizing Recall for strict privacy protection while maintaining Precision to avoid unnecessary edits. For speech substitution, we utilize NISQA-based [28] perceptual quality metrics: NISQA-MOS (overall quality), NISQA-DIS (discontinuity/boundary artifacts), and NISQA-COL (coloration/distortion), alongside Real-Time Factor (RTF) for computational efficiency. RTF is measured end-to-end, encompassing ASR transcription, WhisperX alignment, Kiwi morphological analysis, NER inference, pseudonym rule application, and speech editing, on a single NVIDIA RTX pro 6000 GPU.

5 Experiment

5.1 ASR Performance

Table 1. ASR fine-tuning results (CER, % ↓) compared with other open-source ASR models across various datasets.

Model	Ksponspeech	KtelSpeech	youth	adult	zeroth
Whisper-large-v3	17.99	7.04	26.88	20.63	6.86
Qwen3-asr	13.15	12.20	28.19	26.36	6.37
Dolphin	10.78	61.79	36.49	35.49	16.70
Wav2vec2-large-xlsr-korean	48.52	39.47	59.71	50.94	**1.90**
FT-Whisper (ours)	**6.09**	**2.76**	**18.77**	**13.92**	2.62

As presented in Table 1, our proposed *FT-Whisper* model achieved superior performance across most evaluation datasets. Notably, on the natural conversational *Ksponspeech* [18] dataset, FT-Whisper reduces the CER from 17.99% to 6.09%, representing an 11.90 percentage-point absolute reduction (66.1% relative reduction). While the Korean-tailored *wav2vec2* [25] model recorded the lowest CER (1.90%) on clean read-speech (*zeroth* [20]), its performance degraded severely in noisy or conversational environments. In contrast, our model demonstrated exceptional robustness across all challenging acoustic conditions, providing a reliable foundation for the downstream NER task.

5.2 NER Performance

As shown in Table 2, the detection performance on the KDPII test dataset varies significantly across architectures. GLiNER-based models [13] exhibited low Precision and F1-Scores; even after task-specific fine-tuning (*GLiNER-finetuned*), Precision plummeted to 10.13% despite a high Recall, indicating a severe false-positive issue unsuitable for strict PII detection. Conversely, BERT-based [14]

Table 2. Performance comparison of various NER models on the KDPII dataset.

Model	Precision	Recall	F1-Score
GLiNER_multi-v2.1	0.1492	0.5006	0.2299
GLiNER_multi_pii-v1	0.1047	0.5270	0.1746
GLiNER-ko	0.1504	0.4873	0.2298
GLiNER-finetuned	0.1013	0.8559	0.1811
ModernBERT-base	0.7725	0.7414	0.7566
mDeBERTa-v3-base	0.9105	0.9259	0.9181
xlm-RoBERTa-base	0.9164	0.9255	0.9209
xlm-RoBERTa-large	0.9285	0.9473	0.9378
klue/RoBERTa-large	**0.9389**	**0.9498**	**0.9443**

models consistently maintained high Precision while recording Recalls above 90%. Among them, our fine-tuned *klue/RoBERTa-large* [6] model achieved the highest overall F1-Score (0.9443). By leveraging morphological preprocessing to effectively handle Korean *eojeol* structures, this model successfully delivered the strict boundary precision required for our localized editing pipeline.

5.3 Speech Substitution Quality

Table 3 presents the evaluation results on our synthetic PII dataset.

Table 3. evaluation results (NISQA MOS, DIS, COL, and RTF) of different speech substitution methods on the TTS-generated synthetic dataset.

Method	NISQA-MOS	NISQA-DIS	NISQA-COL	RTF
Original Audio	4.6130	4.5066	4.3836	-
Beep Masking	2.9058	3.4273	2.5224	0.08
TTS-splice	4.3629	4.2831	4.2270	0.81
Ours (SE)	**4.4916**	**4.5134**	**4.2532**	0.16

To contextualize the performance of our SE model, we compared it against two conventional substitution baselines: beep masking and TTS splicing. Both baselines utilize our proposed ASR-NER pipeline to accurately detect the boundaries of sensitive PII. However, they differ in their audio replacement mechanisms: beep masking replaces the target segment with a standard pure tone of equal duration, whereas TTS splicing synthesizes the target pseudonym using a standard TTS model [22] and directly splices the generated waveform into the original audio track.

Our proposed SE-based approach achieved an overall NISQA-MOS [28] of 4.4916, closely matching the original audio (4.6130) and significantly outperforming beep masking (2.9058). Crucially, our method recorded a NISQA-DIS of 4.5134, demonstrating that the pseudonymous segments seamlessly blend into the surrounding context without the boundary artifacts of TTS-splicing (4.2831). The NISQA-COL score (4.2532) further indicates that spectral coloration artifacts introduced by the editing process are minimal.

Regarding efficiency, while beep masking is trivially fast (0.08 RTF), it destroys data utility. TTS-splicing suffers from high computational overhead (0.81

RTF) due to the full synthesis and post-processing required for waveform splicing. By autoregressively predicting only the discrete tokens within localized masked segments, our SE method achieved a highly efficient RTF of 0.16, striking an optimal balance between acoustic naturalness, privacy protection, and processing speed.

6 Conclusion

In this paper, we proposed a novel speech pseudonymization framework utilizing an integrated ASR-NER-SE pipeline to address the privacy challenges associated with sharing voice data. By effectively substituting sensitive personal information with pseudonyms through a context-aware speech editing model, our approach effectively substitutes targeted text spans with plausible pseudonyms without compromising the acoustic integrity of the original audio.

Crucially, our framework addresses the unique morphological complexities of the Korean language by incorporating a morpheme-aware detection module. To rigorously evaluate our pipeline, we constructed a novel synthetic PII speech dataset. Experimental evaluations demonstrate that, compared to conventional de-identification baselines such as beep masking and TTS splicing, our localized editing framework achieves superior perceptual quality and seamless boundary continuity. By achieving high NISQA scores across overall quality, boundary continuity, and coloration dimensions while maintaining high computational efficiency, this method maximizes data utility, highlighting its potential as a robust preprocessing step for reliable application in various downstream speech technologies.

While this study successfully validates the component-level efficacy and the acoustic continuity of localized speech editing on synthetic datasets, future work will explicitly address end-to-end privacy evaluations, including measurements of residual PII intelligibility and speaker linkability on real-world datasets. Furthermore, we plan to extend this framework to handle multi-speaker environments, addressing the complexities of overlapping speech and speaker diarization, and aim to broaden the applicability of our methodology to a robust multilingual pseudonymization system.

6.1 Acknowledgement

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT), (RS-2025-02215142, Development of Pseudonymization Technology for Suspected Criminal Information, 50%) and Korea Internet & Security Agency(KISA) grant funded by the Korea government(PIPC) (No.RS-2026-25530576, Development of Robust Real-time Deepfake Disruption Technology via Adversarial Noise Injection for Privacy Protection, 50%)

References

1. Miao, X., Tomashenko, N., Arefeen, R., Meyer, S., Panariello, M., Wang, X., Vincent, E., Evans, N., Yamagishi, J., Todisco, M.: The VoicePrivacy 2026 Challenge Evaluation Plan. https://hal.science/hal-05561895/. Accessed 29 June 2026
2. Hui Hui, B.S., Miao, X., Wang, X.: SecureSpeech: Prompt-based Speaker and Content Protection. In: Proceedings of the 2025 IEEE International Joint Conference on Biometrics (IJCB). IEEE (2025). https://doi.org/10.1109/IJCB65343.2025.11410690
3. Prajwal, T.S., Desu, L., Kumari, N., Kumar, R.: Content Moderation and File Anonymization using Agentic AI. In: Proceedings of the 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET). IEEE (2025)
4. Abdjul, R.N., Lestari, D.P., Purwarianti, A., Mawalim, C.O., Sakti, S., Unoki, M.: Indonesian Speech Content De-Identification in Low Resource Transcripts. In: Proceedings of the Second Workshop in South East Asian Language Processing. pp. 63–71. Association for Computational Linguistics (2025)
5. Baril, G., Cardinal, P., Koerich, A.L.: Named Entity Recognition for Audio De-Identification. In: Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2022). https://doi.org/10.1109/IJCNN55064.2022.9892285
6. Park, S., et al.: KLUE: Korean Language Understanding Evaluation. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (2021)
7. Wang, H., Yu, M., Hai, J., Chen, C., Hu, Y., Chen, R., Dehak, N., Yu, D.: SSR-Speech: Towards Stable, Safe and Robust Zero-shot Text-based Speech Editing and Synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1–5. IEEE (2025). https://doi.org/10.1109/ICASSP49660.2025.10889119
8. Patino, J., Tomashenko, N., Todisco, M., Nautsch, A., Evans, N.: Speaker Anonymisation using the McAdams Coefficient. In: Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH). pp. 1234–1238 (2021)
9. Miao, X., Wang, X., Cooper, E., Yamagishi, J., Tomashenko, N.: Speaker Anonymization Using Orthogonal Householder Neural Network. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 3681–3695 (2023)
10. Yao, J., et al.: NPU-NTU System for Voice Privacy 2024 Challenge. System Description, VoicePrivacy Challenge (2024)
11. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust Speech Recognition via Large-Scale Weak Supervision. In: Proceedings of the 40th International Conference on Machine Learning (ICML). PMLR, vol. 202, pp. 28492–28518 (2023)
12. Bain, M., Huh, J., Han, T., Zisserman, A.: WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. In: Proceedings of the 24th Annual Conference of the International Speech Communication Association (INTERSPEECH) (2023)
13. Zaratiana, U., Tomeh, N., Holat, P., Charnois, T.: GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 5364–5376. Association for Computational Linguistics (2024)

14. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019)
15. Lee, M.: Kiwi: Developing a Korean Morphological Analyzer Based on Statistical Language Models and Skip-Bigram. Korean Journal of Digital Humanities **1**(1), 109–136 (2024). https://doi.org/10.23287/KJDH.2024.1.1.6
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
17. Fei, L., Kang, Y., Park, S., Jang, Y., Lee, J., Kim, H.: KDPII: A New Korean Dialogic Dataset for the Deidentification of Personally Identifiable Information. IEEE Access **12**, 135626–135641 (2024). https://doi.org/10.1109/ACCESS.2024.3461804
18. Bang, J.-U., Yun, S., Kim, S.-H., Choi, M.-Y., Lee, M.-K., Kim, Y.-J., Kim, D.-H., Park, J., Lee, Y.-J., Kim, S.-H.: KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition. Applied Sciences **10**(19), 6936 (2020). https://doi.org/10.3390/app10196936
19. AI Hub: Counseling Speech (KtelSpeech) Dataset. https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=100 (2020). Accessed 30 June 2026
20. Jo, L., et al.: Zeroth-Korean: Open-Source Korean Speech Corpus. GitHub repository. https://github.com/goodatlas/zeroth. Accessed 30 June 2026
21. Gemma Team, et al.: Gemma: Open Models Based on Gemini Research and Technology. arXiv preprint arXiv:2403.08295 (2024)
22. Hu, H., et al.: Qwen3-TTS Technical Report. arXiv preprint arXiv:2601.15621 (2026)
23. Shi, X., Wang, X., Guo, Z., Wang, Y., Zhang, P., Zhang, X., Guo, Z., Hao, H., Xi, Y., Yang, B., Xu, J., Zhou, J., Lin, J.: Qwen3-ASR Technical Report. arXiv preprint arXiv:2601.21337 (2026)
24. Meng, Y., et al.: Dolphin: A Large-Scale Automatic Speech Recognition Model for Eastern Languages. arXiv preprint arXiv:2503.20212 (2025)
25. kresnik: wav2vec2-large-xlsr-korean. Hugging Face. https://huggingface.co/kresnik/wav2vec2-large-xlsr-korean. Accessed 30 June 2026
26. Warner, B., et al.: Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. pp. 2526–2547. Association for Computational Linguistics (2025)
27. He, P., Gao, J., Chen, W.: DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In: Proceedings of the International Conference on Learning Representations (ICLR) (2023)
28. Mittag, G., Naderi, B., Chehadi, A., Möller, S.: NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets. In: Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH) (2021)
29. AI Hub: Emotion-Tagged Free Conversation (Youth) Dataset. https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=71632 (2022). Accessed 30 June 2026

30. AI Hub: Emotion-Tagged Free Conversation (Adult) Dataset. https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=71631 (2022). Accessed 30 June 2026

Quiet-Subspace Diffusion-Model Watermarking: A Knowledge-Graded Attack Study

Jinwoo Lee[1], Daeseon Choi[1*], and Hyunsik Na[1]

Soongsil University, Republic of Korea
leejwian@gmail.com, sunchoi@ssu.ac.kr
nrnud7932@soongsil.ac.kr
*Corresponding author.

Abstract. We study parameter-space watermarking of diffusion models. At an early checkpoint, we construct a low-eigenvalue subspace E from the final-readout input Gram matrix and retain the reference readout θ_{ref}. Continued training writes a recipient-specific sign key along E; verification tests cosine alignment of the reference-relative projected weights. Among equal-dimensional readout subspaces, this carrier minimizes the worst-case fixed-feature local mean-square readout effect over unit-norm perturbations; among the tested carriers, it also has the smallest measured normalized drift in post-deployment continuation windows. Under an isotropic Gaussian-direction null in $k = 8$, the target exceeds the strict 10^{-4} threshold and none of the 255 nonmatching sign keys passes; this is directional calibration, not a false-positive-rate estimate over independently trained clean models. Across five paired continuations, only one unadjusted interval indicating degradation excludes zero: a $+0.2307$ final-checkpoint FID effect. In the evaluated G-unaware T1–T2 suite, only severely quality-degrading pruning evades, whereas geometry-aware attacks—exact projection, subspace noise, and cosine-targeting fine-tuning—expose lower-cost failures. These results establish a knowledge-graded empirical boundary, not a general white-box guarantee, and motivate—but do not validate—a joint cosine-and-magnitude detector.

Keywords: Diffusion models · Parameter-space watermarking · Model ownership · Robustness · White-box attacks.

1 Introduction

Diffusion models [5, 11, 17, 20] are expensive to train and easy to redistribute, creating a need for ownership evidence that survives ordinary model handling. *Output-space* watermarks alter generated content so a signature can be recovered from images [8, 25, 28]; *white-box parameter-space* watermarks write a signature into the weights so ownership can be tested directly on a suspect checkpoint [21]. Black-box or trigger-based DNN watermarks instead verify ownership through model responses [1, 29]. We study the white-box parameter-space family.

A parameter-space watermark should be *detectable*, *utility-preserving*, and *robust*. Placement creates a tension: directions updated heavily risk overwriting

the signature, whereas readout-sensitive directions may impair utility. We embed the key in the subspace spanned by the smallest-eigenvalue eigenvectors of an early-checkpoint readout-input Gram matrix and separately test its quietness under later optimization. Our contributions are:

1. **A low-eigenvalue readout watermark.** Among equal-dimensional readout subspaces, the carrier is minimax for the fixed-feature local mean-square readout effect of unit-norm perturbations and has the smallest measured normalized drift among tested carriers.
2. **Directional detection and matched utility evidence.** The target exceeds the strict Gaussian-direction 10^{-4} threshold, no nonmatching sign key passes, and five causally matched continuations quantify FID and memorization effects.
3. **A knowledge-graded attack study.** The evaluated G-unaware T1–T2 suite shows no quality-preserving evasion, whereas geometry-aware attacks expose lower-cost failures; this is an empirical boundary, not a general white-box guarantee.

2 Related Work

Output-space methods embed recoverable signals in generated content and test them under image-level transforms [8, 25, 28, 30]. White-box methods include direct weight regularization [21], adversarially trained covert RIGA [23], and collusion-secure multi-user DeepMarks [4]; activation- or representation-space methods read internal behavior [18], while black-box methods use model responses [1, 29]. Ours places a white-box parameter watermark in the final readout.

Recent diffusion-model schemes use output-verified AquaLoRA [7], semantic disentanglement for downstream fine-tuning [24], loss-landscape smoothing [26], or representation fingerprints [12]. Functional Subspace Watermarking solves a generalized eigenproblem for a stable LLM carrier [6]; ours instead uses a low-eigenvalue subspace of a final-readout feature Gram matrix and a reference-relative projection.

Removal studies cover fine-tuning, pruning, overwriting, and adaptive attacks [14, 19, 22]. Function-preserving neuron transforms, structural obfuscation, and DeepEclipse can defeat white-box verifiers without ordinary retraining [15, 16, 27], while pruning and quantization are standard compression operations [9]. We grade attacks by knowledge of the detector geometry G and preserve architecture and parameter indexing; structural attacks therefore define an important scope boundary rather than an evaluated attack family.

3 Threat Model and Preliminaries

3.1 Notation and Detection

Let $\theta \in \mathbb{R}^d$ be the flattened final readout weight vector, $\theta = \mathrm{vec}(W_{\mathrm{final}})$, a subset of the full parameters ϕ, and let θ_{ref} be its value at a fixed checkpoint. With an

orthonormal basis $E \in \mathbb{R}^{d\times k}$ of the watermark subspace, key $w \in \{-1,+1\}^k$, and stabilizer $\varepsilon = 10^{-12}$, the detector uses the projected change z and the statistics

$$z = E^\top(\theta - \theta_{\text{ref}}) \in \mathbb{R}^k, \qquad \texttt{low_norm} = \|z\|_2,$$
$$\texttt{score_cos} = \frac{\langle z, w\rangle}{\|z\|_2\|w\|_2 + \varepsilon}, \qquad \texttt{score_signed} = \frac{\langle z, w\rangle}{\|w\|_2 + \varepsilon}. \tag{1}$$

With $b = E^\top\theta_{\text{ref}}$, $z = E^\top\theta - b$, so detection needs only $G = (E, b)$, not θ_{ref} itself. `score_cos` is an ε-stabilized directional statistic, effectively scale-invariant away from zero; `low_norm` is the projected magnitude – some attacks change one without the other. To avoid assigning a direction to numerical zero, we fix a near-zero tolerance $\delta = 10^{-8}$: when $\|z\|_2 \le \delta$, the verifier returns non-detection and does not interpret `score_cos`; otherwise it applies the score threshold. Against a null sample of scores we report the plus-one-smoothed one-sided `empirical_p` $= (|\{s' \ge s\}| + 1)/(N_{\text{null}} + 1)$. Thresholds at a requested tail level q are empirical upper quantiles of the sorted null ($a = \max(1, \lfloor q \cdot N_{\text{null}} \rfloor)$, threshold $v[\,N_{\text{null}} - a\,]$, zero-based indexing); with few samples or ties the realized exceedance can exceed q, so these are operating points on the *sampled* directional null, not certified false-positive rates over independently trained clean models. The verifier flags a watermark only when $\|z\|_2 > \delta$ and `score_cos` meets a threshold.

3.2 Threat Model

We grade attackers by knowledge of $G = (E, b)$—needed to compute z—as G-unaware, geometry-aware, or reference-aware (the first two evaluated); reference-aware attackers additionally know the retained readout θ_{ref}.

T1 – **G-unaware transforms.** May know the algorithm, architecture, and carrier layer, but not G, w, or θ_{ref}: pruning and quantization.

T2 – **G-unaware fine-tuning.** Continued training on data with no detector targeting and no use of G, w, or θ_{ref}.

T3 – **Cosine-targeting fine-tuning.** Uses G to minimize `score_cos`^{2}. Our run uses the recipient key w, but for its unattacked initial checkpoint θ_0, $\widehat{w} = \text{sign}(E^\top\theta_0 - b) = w$; hence (G, θ_0) recovers the identical target direction in this run, so no key knowledge beyond G is needed.

T4 – **Exact projection.** Knows G and subtracts the projected component exactly.

T1–T2 are the evaluated G-unaware redistribution attacks; T3–T4 are geometry-aware. Geometry-aware subspace noise injects noise in the exact subspace E. We do not evaluate (i) reference-aware substitution of the final readout by θ_{ref}, which makes $z = 0$ by construction; (ii) G-unaware method-aware replacement, reinitialization, or recovery training of the known carrier layer; or (iii) generic model merging. Their utility costs are unmeasured.

Scope. The attacker sees every suspect parameter (model-level white box) and may know the architecture and training setup. The private owner material comprises G, w, and θ_{ref}. Evaluated attacks preserve the architecture and detector-read parameter indexing; function-preserving reparameterizations (channel permutation/rescaling) and recipient collusion are out of scope.

4 Method: A Low-Eigenvalue Readout Watermark

4.1 Subspace, Key, and Insertion

The matrix U is the empirical feature Gram matrix of the readout layer's own input at the generation checkpoint τ_{gen}: each local 3×3 patch of the feature tensor entering the final readout convolution is flattened to $h_m \in \mathbb{R}^{288}$ (32 input channels × 3×3 kernel positions); the final readout convolution has a single output channel, so the flattened readout weight shares these coordinates, $\theta = \text{vec}(W_{\text{final}}) \in \mathbb{R}^{1\cdot 32\cdot 3\cdot 3} = \mathbb{R}^{288}$ ($d = 288$). We take one patch per spatial position of all 1,024 training images after forward diffusion at independently sampled timesteps, and $U = H^\top H/M \in \mathbb{R}^{288\times 288}$ with the h_m as the rows of H and $M = 1{,}048{,}576$ (float64 accumulation); its low-eigenvalue eigenvectors are thus the patch directions along which the readout's input carries the least energy at τ_{gen}. Guarding against numerical asymmetry we take $U_{\text{sym}} = \frac{1}{2}(U + U^\top)$ with eigendecomposition $U_{\text{sym}} = V\Lambda V^\top$, eigenvalues ascending and $V = [v_1, \ldots, v_d]$ orthonormal. The watermark uses the k smallest-eigenvalue eigenvectors $E_{\text{low}} = [v_1, \ldots, v_k]$; ablations use the k largest (E_{high}) and QR-orthonormalized Gaussian random subspaces (E_{rand}). The recipient codebook stores sign keys $w \in \{-1, +1\}^k$; with $k = 8$ the recipient used here is $w = (+1, +1, -1, -1, +1, +1, -1, -1)$ and the target is $z_{\text{target}} = \alpha w$, $\alpha = 0.02$, so $\|z_{\text{target}}\|_2 = \alpha\sqrt{k} = 0.02\sqrt{8} \approx 0.0566$, matching the measured unattacked `low_norm` (Section 6.4).

The watermark is inserted by continued training, not by overwriting (Fig. 1): from θ_{ref} at τ_{gen}, each batch optimizes

$$L = L_{\text{diff}} + \lambda L_{\text{wm}}, \quad L_{\text{diff}} = \text{MSE}\big(\epsilon_t, f_\phi(x_t, t)\big), \quad L_{\text{wm}} = \tfrac{1}{k}\|E^\top(\theta - \theta_{\text{ref}}) - \alpha w\|_2^2, \tag{2}$$

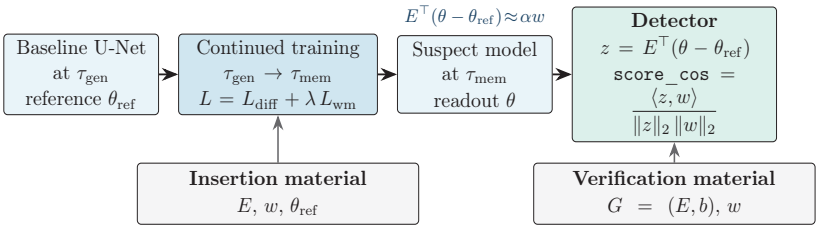


Fig. 1. Method overview: continued training writes w into E_{low}; verification computes $z = E^\top(\theta - \theta_{\text{ref}})$ and tests its cosine alignment with w. Insertion uses $(E, w, \theta_{\text{ref}})$ (with target scale α); verification uses (G, w), where $G = (E, b)$ and $b = E^\top\theta_{\text{ref}}$.

with λ constant at $\lambda_{\mathrm{wm}} = 1.0$. Training runs from τ_{gen} to τ_{mem}; the checkpoint at τ_{mem} is the suspect model, with $E^{\top}(\theta_{\tau_{\mathrm{mem}}} - \theta_{\mathrm{ref}}) \approx \alpha w$: the projected direction is aligned with the key while the diffusion loss keeps training the image model.

4.2 Design Rationale: The Low-Eigenvalue Subspace Is Quiet

Why might E_{low} be quiet? A change $\Delta\theta$ moves the single-output readout on patch h_m by $\Delta\theta^{\top} h_m$, with fixed-feature local mean-square readout effect $\frac{1}{M}\sum_m (\Delta\theta^{\top} h_m)^2 = \Delta\theta^{\top} U \Delta\theta$. By Courant–Fischer, among all k-dimensional readout subspaces, E_{low} minimizes the worst-case value of this effect over unit-norm changes. This bounds output sensitivity, not SGD drift, so we test drift empirically.

We measure how much ordinary, non-watermark training perturbs each candidate subspace via the normalized drift $\|E_g^{\top}(\theta_t - \theta_s)\|_2 / \|\theta_t - \theta_s\|_2$ (for $g \in \{\mathrm{low}, \mathrm{rand}, \mathrm{high}\}$) on the baseline trajectory, where θ_s is the baseline checkpoint at the start of the continuation window (step 50000; it differs from θ_{ref} at τ_{gen}, while the subspaces remain those built at τ_{gen}). These clean-baseline windows start at step $\tau_{\mathrm{mem}} = 50000$, matching the step range in which the deployed suspect lives and post-deployment fine-tuning (Section 6.3) operates; the subspaces E stay fixed at τ_{gen} for all later detection. Figure 2 shows that across windows of 1000/5000/10000 steps the low-eigenvalue subspace consistently receives the least natural drift – about a third of the random and a seventh or less of the high subspace – which is the operational reason to embed there.

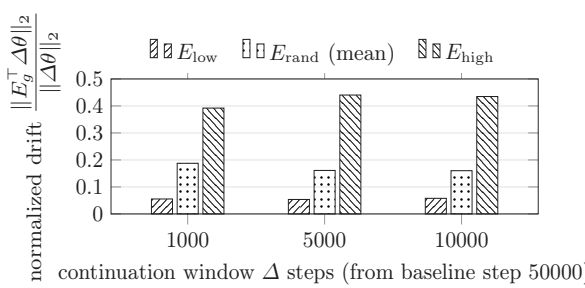


Fig. 2. Normalized natural drift over 1K/5K/10K-step windows from baseline step 50000 (E_{rand}: mean of five seeds): E_{low} 0.0554/0.0536/0.0578, E_{rand} 0.1876/0.1611/0.1602, and E_{high} 0.3921/0.4404/0.4349.

5 Experimental Setup

We train a diffusion U-Net on a fixed 1,024-image grayscale CelebA 32×32 subset [13] (single-channel, matching the single-output readout), writing the watermark into the final readout convolution weights, with $\tau_{\mathrm{gen}} = 20000$, $\tau_{\mathrm{mem}} = 50000$,

$k = 8$, $\alpha = 0.02$, and $\lambda_{\mathrm{wm}} = 1.0$. The base U-Net (width 32) uses Adam (lr 10^{-4}, batch 512); FID (`pytorch-fid`, Inception-v3) and memorization use 10,000 generated samples each. The τ_{gen}–τ_{mem} interval is motivated by the generation- and memorization-timescale picture of diffusion training [2], but we do *not* treat it as memorization-free because baseline nearest-neighbor memorization already rises across this window. Accordingly, Section 6.2 tests the *baseline-matched* incremental FID and memorization effects of the watermark loss using causally matched continuations. Quality is FID [10] (lower is better; ≈ 26 is near-baseline for the deployed model). Motivated by diffusion training-data extraction [3], memorization is Bonnaire et al.'s nearest/second-nearest training-image L_2 ratio below 1/3 in flattened 32×32 pixel space [2], reported in percent; paired effects are percentage-point differences. Attacks target the E_{low} model unless noted.

6 Results

6.1 Directional Calibration and Key Specificity

We calibrate the strict detector on an isotropic Gaussian-direction null, the only sampled null with 10^{-4} resolution. Scoring E_{low} against 10,000 Gaussian directions (not recipient sign keys) gives the empirical thresholds in Table 1; the strict 10^{-4} threshold is the sample maximum 0.9424, while the target has `score_cos`≈ 1.0. As limited negative controls, four clean and four clean-fine-tuned checkpoints scored against the target key are near zero or negative (maxima 0.000 and -0.027); the clean maximum is θ_{ref}, whose $z = 0$ triggers the near-zero non-detection rule.

Analytic Threshold and Scope. For an isotropic random direction against a fixed target in k dimensions, $(\texttt{score_cos} + 1)/2 \sim \mathrm{Beta}(\frac{k-1}{2}, \frac{k-1}{2})$. For $k = 8$, the analytic upper-tail-10^{-4} threshold is 0.9365; among 10^6 fresh directions, 96 exceed it. The empirical 0.9424 threshold is slightly more conservative, and every pass/fail decision in this paper is identical under either strict threshold. The target exceeds all 10,000 sampled directions, giving plus-one `empirical_p`$= 9.999 \times 10^{-5}$ and a one-sided 95% Clopper–Pearson upper bound of 3.0×10^{-4}. These are directional-null statements, not false-positive-rate estimates over independently trained clean models; the clean evidence is limited to the correlated checkpoints above.

Exhaustive Key Specificity. Since $k = 8$ admits 256 sign codewords, we enumerate all of them. Their cosines follow $1 - h/4$ at Hamming distance h (within $\lesssim 3 \times 10^{-4}$ of theory); the maximum nonmatching sign key score is 0.7501 ($h = 1$), at least 0.186 below both strict thresholds, so 0/255 pass. At the 1% level the claim is threshold-sensitive: the analytic 0.7498 threshold lies just below the eight h=1 codewords, whereas the empirical 0.7530 threshold excludes them. We therefore claim key specificity only at the strict operating point. Larger k tightens the directional null but not nearest-key separation ($1 - 2/k \to 1$), so multi-recipient use requires a minimum-distance codebook; $E_{\mathrm{high}}/E_{\mathrm{rand}}$ behave the same within $\sim 10^{-3}$. This calibration fixes the strict thresholds used below.

Table 1. Directional-null calibration for $k = 8$. Empirical thresholds use 10^4 Gaussian directions; analytic thresholds use the isotropic Beta law. The target score is 1.0000; the last column gives exceedances among 10^6 fresh directions.

tail level	analytic threshold	empirical threshold	MC check: 10^6 exc.
10^{-1}	0.4716	0.4725	99,964
10^{-2}	0.7498	0.7530	9,901
10^{-3}	0.8751	0.8879	992
10^{-4}	0.9365	0.9424	96

Table 2. Checkpoint-wise effects (watermark − clean) over five paired continuation seeds. The τ_{gen} row is zero by construction and omitted; exploratory 95% Student-t intervals are unadjusted for multiplicity. Bold marks the final FID interval excluding zero; the 25000 memorization interval also excludes zero.

step	FID effect [95% CI]	memorization effect (pp) [95% CI]
25000	+0.3714 [−0.1934, 0.9361]	−0.0456 [−0.0845, −0.0067]
30000	+0.1014 [−0.1869, 0.3897]	+0.0120 [−0.0665, 0.0905]
35000	−0.1599 [−0.6216, 0.3018]	+0.0464 [−0.0674, 0.1602]
40000	+0.1740 [−0.3157, 0.6636]	−0.0288 [−0.2458, 0.1882]
45000	−0.1407 [−0.5447, 0.2633]	−0.0552 [−0.3346, 0.2242]
50000	**+0.2307** [**+0.0101**, **+0.4513**]	−0.1328 [−0.3208, 0.0552]

6.2 Matched Quality and Memorization Effects

To isolate the watermark loss, five paired post-τ_{gen} continuations train from the same τ_{gen} base; within each pair both arms restart a fresh optimizer and share mini-batch order, diffusion timesteps and noise, dropout, and evaluation seeds, differing *only* in the watermark loss ($\lambda_{\text{wm}} = 0$ vs. 1). Paired generation-seed effects are first averaged within each continuation seed, the replication unit (Table 2). All unadjusted intervals include zero except the final-checkpoint FID interval (+0.2307, 95% CI $[0.0101, 0.4513]$) and the step-25000 memorization interval (−0.0456 pp, 95% CI $[-0.0845, -0.0067]$). With five pairs, no multiplicity adjustment, and no prespecified non-inferiority margin, we interpret these intervals descriptively and do not claim equivalence, harmlessness, or improved memorization. All five continuations share one pre-τ_{gen} base, so base-model variation remains unmeasured.

6.3 Robustness to Non-Adaptive Fine-Tuning (T2)

With fresh Adam (lr 10^{-4}), we continue training on a disjoint 1,024-image subset for 10,000 steps without detector targeting; all detector scores remain far above both strict thresholds: `score_cos` at 1K/5K/10K steps is 0.9993/0.9992/0.9985 (E_{low}), 0.9987/0.9968/0.9919 (E_{rand} mean), and 0.9962/0.9976/0.9795 (E_{high}), and each of the five individual random-subspace models stays above its own strict threshold (10K `score_cos` 0.989–0.997) – far above every threshold of

Section 6.1. We report `low_norm` retention as $\|z_t\|_2/\|z_0\|_2$ (vs. the unattacked checkpoint); at the same steps it is 1.0073/1.0018/1.0007, 0.9985/0.9633/0.9239, and 1.0495/0.9135/0.7935: E_{low} is essentially unchanged while E_{high} loses a fifth of its magnitude. FID rises transiently without removing any group (at 0/1K/5K/10K: E_{low} 26.2/44.2/50.6/40.6; E_{rand} 26.4/37.1/48.4/39.7; E_{high} 26.0/35.2/44.3/45.1; unpaired point estimates). Each model is one attack trajectory (no recorded seed), so the cross-subspace ranking is descriptive; none of the seven evaluated T2 trajectories removes the watermark, consistent with Section 4.2. All placements remain detectable; the observable low-eigenvalue advantage is magnitude retention, not a pass/fail distinction.

6.4 Attack Suite on the Low-Eigenvalue Model

The unattacked reference is `score_cos`= 1.0000 and `score_signed`=`low_norm`= 0.05657 at FID $\approx$ 26. Attacks follow Section 3.2. T3 continues training on the disjoint subset from Section 6.3 (Adam, lr 10^{-4}, 10,000 steps), optimizing $L = L_{\text{diff}} + \lambda_{\text{att}}\,\texttt{score_cos}^2$. Every strict pass/fail decision is identical under the analytic (0.9365) and empirical (0.9424) thresholds. FID averages five reference-statistic files per model, but generation seeds are not shared across rows, so small mid-20s differences are not ranked. Table 3 reports the tabulated families; geometry-aware subspace noise is reported separately, and Fig. 3 summarizes the score–FID trade-off.

Table 3. Attacks on E_{low} (reference: `score_cos`= 1.0000, `low_norm`= 0.05657, FID $\approx$ 26). T1 is G-unaware; T3/T4 are geometry-aware; each fine-tuning row is one trajectory. †For `low_norm`$\leq 10^{-8}$ the verifier returns non-detection. ‡The 50% pruning score is 0.0002/0.0061 below the analytic/empirical strict thresholds.

family	setting	`score_cos`	`low_norm`	FID
T1 pruning	0.1	0.9994	0.0575	24.42
T1 pruning	0.3	0.9980	0.0535	80.53
T1 pruning	0.5	$0.9363^{\ddagger}$	0.0588	200.14
T1 pruning	0.7	0.6581	0.0674	636.50
T1 quantization	FP16	1.0000	0.0566	26.06
T1 quantization	INT8	0.9999	0.0569	24.75
T1 quantization	INT4	0.9497	0.0602	450.43
T4 projection	$\gamma = 0.5$	1.0000	0.02829	26.05
T4 projection	$\gamma = 0.9$	1.0000	0.00566	26.32
T4 projection	$\gamma = 1.0$	non-det.†	4.7×10^{-9}	26.40
T3 cosine-FT	$\lambda_{\text{att}} = 0$	0.998453	0.056703	41.04
T3 cosine-FT	$\lambda_{\text{att}} = 0.01$	-9.4×10^{-7}	0.061751	52.15
T3 cosine-FT	$\lambda_{\text{att}} = 0.1$	-4.73×10^{-5}	0.061056	43.86
T3 cosine-FT	$\lambda_{\text{att}} = 1$	1.19×10^{-4}	0.060642	40.53
T3 cosine-FT	$\lambda_{\text{att}} = 10$	6.29×10^{-6}	0.060285	33.66

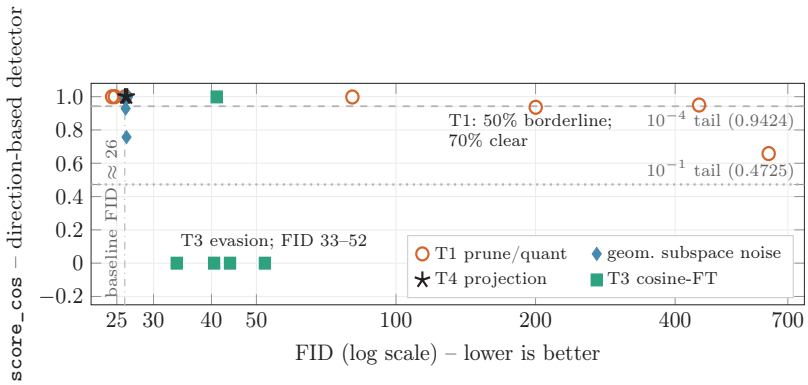


Fig. 3. `score_cos`–FID landscape on E_{low}. In the evaluated G-unaware T1–T2 suite, only severely quality-degrading pruning crosses the strict threshold; lower-FID evasions are geometry-aware. The subspace-noise sweep is one realization. The $\gamma = 1$ projection is omitted because `low_norm`$= 4.7 \times 10^{-9}$ triggers non-detection (FID 26.40).

G-Unaware Parameter Transformations (T1). Global magnitude pruning is applied to all floating-point tensors; FP16 is a round-trip cast, and INT8/INT4 are model-wide per-tensor fake quantizations. Pruning at 10/30% and all tested quantizations remain detected. The 50% pruning point is a borderline evasion (`score_cos`$= 0.9363$), while 70% clearly evades (`score_cos`$= 0.6581$); their FIDs are 200.14 and 636.50. Thus no evaluated T1 setting provides a quality-preserving evasion.

Geometry-Aware Subspace Noise. Gaussian noise is injected in the exact subspace coordinates, $\theta_\rho = \theta + E\xi$ with $\xi \sim \mathcal{N}(0, \sigma^2 I_k)$ and $\sigma = \rho\|z\|_2/\sqrt{k}$, for $\rho \in \{0.01, 0.03, 0.05, 0.1, 0.2, 0.5, 1.0\}$. In one realization, $\rho = 0.5$ evades both strict thresholds (`score_cos`$= 0.9288$) at near-baseline FID 26.12. The key-aligned component remains near its unattacked value (`score_signed`≈ 0.058 vs. 0.057), while orthogonal subspace energy raises `low_norm` to 0.0621 and reduces cosine alignment; at $\rho = 1.0$, the score 0.7571 still exceeds the 1% threshold. This attack requires exact G; the single realization is a stress test, not an evasion-rate estimate.

Exact Projection (T4). The attack subtracts a fraction γ of the projected component, $\theta_\gamma = \theta - \gamma E_{\text{low}} z$. For $\gamma < 1$, the direction remains unchanged (`score_cos`$= 1.0$) and the removed projected-magnitude fraction equals γ. At $\gamma = 1$, $E^\top(\theta_\gamma - \theta_{\text{ref}}) = z - z = 0$ exactly; the measured residual norm 4.7×10^{-9} is below δ, so the verifier returns non-detection without interpreting `score_cos` (FID 26.40). Overshoot reverses the sign. Exact projection requires G and is an attacker-capability upper bound, not evidence that generic attacks remove the watermark.

Cosine-Targeting Fine-Tuning (T3). T3 drives `score_cos` to $\sim 10^{-4}$ or below by suppressing the component along w: |`score_signed`| $\approx$ |`score_cos`| `low_norm`

falls below 7.3×10^{-6} while `low_norm` stays near 0.060, leaving projected energy nearly orthogonal to the key. Evading checkpoints have FID 33–52, but the $\lambda_{\text{att}} = 0$ control is already at 41.04 and each setting is one unseeded trajectory, so attack-specific FID cost is not isolated. The retained projected magnitude motivates considering an additional magnitude statistic; its validity requires separate calibration.

7 Discussion

Knowledge-Graded Boundary. Within the evaluated G-unaware T1–T2 suite, we observe no quality-preserving evasion: ordinary fine-tuning and mild model handling retain detection, while successful pruning coincides with severe FID degradation. Geometry-aware attacks expose lower-cost failures than destructive G-unaware pruning. This separation is limited to the evaluated suite and does not show that G is necessary for lower-cost evasion; method-aware full-readout perturbations may also inject energy into E at unmeasured utility cost.

Implications, Scope, and Verification Practice. Reliance on geometry secrecy is therefore a vulnerability: G enables exact projection, and θ_{ref} enables a reset to $z = 0$. In T3, G and the unattacked initialization also reveal the target sign direction (Section 3.2). Geometry-aware subspace noise and cosine-targeting fine-tuning preserve `low_norm`, motivating a joint `score_cos`/`low_norm` rule; it remains unvalidated without an independent clean-model magnitude null and joint false-positive-rate analysis, and exact projection defeats both statistics.

The evidence covers one grayscale CelebA 32×32 subset, one architecture and key, five continuations from one base checkpoint, a directional rather than independent-clean-model null, one subspace-noise realization, and one trajectory per fine-tuning setting. The suite also preserves architecture and parameter indexing; invariant transforms, structural obfuscation, and DeepEclipse motivate weight canonicalization [15, 16, 27]. Reference substitution is characterized only analytically, and readout replacement, reinitialization, recovery training, model merging, partial geometry, and collusion remain unevaluated. Operational ownership claims should use an authenticated pre-release commitment to canonical encodings of G, w, the model version, and the fixed decision rule. Future work should measure false-positive rates over independently trained models, repeat attacks across seeds, test partial-knowledge adversaries, and calibrate a joint detector.

8 Conclusion

We introduced a quiet-subspace watermark for diffusion models. Its low-eigenvalue carrier is minimax among equal-dimensional readout subspaces for the fixed-feature local mean-square readout effect of unit-norm perturbations and has the smallest measured normalized drift among tested carriers. Directional calibration and exhaustive key testing assess detection, causally matched continuations assess

utility, and attacks separate persistence under evaluated G-unaware handling from geometry-aware lower-cost failures. Quiet placement improves empirical persistence but does not secure a cosine verifier once its geometry is exposed.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00398353, Development of Countermeasure Technologies for Generative AI Security Threats).

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

References

1. Adi, Y., Baum, C., Cisse, M., Pinkas, B., Keshet, J.: Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In: USENIX Security 2018. pp. 1615–1631. USENIX Association (2018)
2. Bonnaire, T., Urfin, R., Biroli, G., Mézard, M.: Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In: NeurIPS. vol. 38, pp. 141266–141286. Curran Associates, Inc. (2025)
3. Carlini, N., Hayes, J., Nasr, M., et al.: Extracting training data from diffusion models. In: USENIX Security 2023. pp. 5253–5270. USENIX Association (2023)
4. Chen, H., Rouhani, B.D., Fu, C., Zhao, J., Koushanfar, F.: DeepMarks: A secure fingerprinting framework for digital rights management of deep learning models. In: ACM ICMR. pp. 105–113. ACM (2019). https://doi.org/10.1145/3323873.3325042
5. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS. vol. 34, pp. 8780–8794. Curran Associates, Inc. (2021)
6. Ding, Z., Li, J., Wu, S., Yao, J., Liu, H., Hu, L.: Functional subspace watermarking for large language models. CoRR **abs/2603.18793** (2026). https://doi.org/10.48550/arXiv.2603.18793
7. Feng, W., Zhou, W., He, J., et al.: AquaLoRA: Toward white-box protection for customized stable diffusion models via watermark LoRA. In: ICML. Proceedings of Machine Learning Research, vol. 235, pp. 13423–13444. PMLR (2024)
8. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. In: IEEE/CVF ICCV. pp. 22409–22420. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.02053
9. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: ICLR (2016)
10. 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. In: NeurIPS. vol. 30, pp. 6626–6637. Curran Associates, Inc. (2017)
11. Ho, J., Jain, A.N., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
12. Li, Z., Qu, H., Kuen, J., et al.: DiffIP: Representation fingerprints for robust IP protection of diffusion models. In: IEEE/CVF ICCV. pp. 17035–17045. IEEE (2025)
13. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE ICCV. pp. 3730–3738. IEEE (2015). https://doi.org/10.1109/ICCV.2015.425

14. Lukas, N., Jiang, E., Li, X., Kerschbaum, F.: SoK: How robust is image classification deep neural network watermarking? In: IEEE S&P. pp. 787–804. IEEE (2022). https://doi.org/10.1109/SP46214.2022.9833693
15. Pan, X., Zhang, M., Yan, Y., Wang, Y., Yang, M.: Cracking white-box DNN watermarks via invariant neuron transforms. In: ACM SIGKDD KDD. pp. 1783–1794. ACM (2023). https://doi.org/10.1145/3580305.3599291
16. Pegoraro, A., Segna, C., Kumari, K., Sadeghi, A.: DeepEclipse: How to break white-box DNN-watermarking schemes. In: USENIX Security 2024. pp. 5287–5304. USENIX Association (2024)
17. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE/CVF CVPR. pp. 10684–10695. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01042
18. Rouhani, B.D., Chen, H., Koushanfar, F.: DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In: ACM ASPLOS. pp. 485–497. ACM (2019). https://doi.org/10.1145/3297858.3304051
19. Shafieinejad, M., Lukas, N., Wang, J., Li, X., Kerschbaum, F.: On the robustness of backdoor-based watermarking in deep neural networks. In: ACM IH&MMSec. pp. 177–188. ACM (2021). https://doi.org/10.1145/3437880.3460401
20. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
21. Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S.: Embedding watermarks into deep neural networks. In: ACM ICMR. pp. 269–277. ACM (2017). https://doi.org/10.1145/3078971.3078974
22. Wang, T., Kerschbaum, F.: Attacks on digital watermarks for deep neural networks. In: IEEE ICASSP. pp. 2622–2626. IEEE (2019). https://doi.org/10.1109/ICASSP.2019.8682202
23. Wang, T., Kerschbaum, F.: RIGA: Covert and robust white-box watermarking of deep neural networks. In: The Web Conference. pp. 993–1004. ACM (2021). https://doi.org/10.1145/3442381.3450000
24. Wang, Z., Guo, J., Zhu, J., et al.: SleeperMark: Towards robust watermark against fine-tuning text-to-image diffusion models. In: IEEE/CVF CVPR. pp. 8213–8224. IEEE (2025)
25. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-Rings watermarks: Invisible fingerprints for diffusion images. In: NeurIPS. vol. 36, pp. 58047–58063. Curran Associates, Inc. (2023). https://doi.org/10.52202/075280-2529
26. Xie, Y., Min, R., Qin, Z., et al.: RoMa: A robust model watermarking scheme for protecting IP in diffusion models. In: NeurIPS. vol. 38, pp. 139640–139662. Curran Associates, Inc. (2025)
27. Yan, Y., Pan, X., Zhang, M., Yang, M.: Rethinking white-box watermarks on deep learning models under neural structural obfuscation. In: USENIX Security 2023. pp. 2347–2364. USENIX Association (2023)
28. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: IEEE/CVF ICCV. pp. 14428–14437. IEEE (2021)
29. Zhang, J., Gu, Z., Jang, J., et al.: Protecting intellectual property of deep neural networks with watermarking. In: ACM AsiaCCS. pp. 159–172. ACM (2018). https://doi.org/10.1145/3196494.3196550
30. Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N., Lin, M.: A recipe for watermarking diffusion models. CoRR **abs/2303.10137** (2023). https://doi.org/10.48550/arXiv.2303.10137

A Unified Ensemble Learning Framework for Multi-Class Intrusion Detection in Wireless Sensor Networks

Thi-Thu-Huong Le[1[0000−0002−8366−9396]], Mulyasari Wulan Silvani[2[0009−0009−6564−3668]], Andro Aprila Adiputra[2[0009−0000−9467−2841]], Hyunjin Jang[2[0009−0004−2649−8350]], Jaehan Cho[2[0009−0008−4878−9123]], and Howon Kim[2[0000−0001−8475−7294]]

[1] Blockchain Platform Research Center, Pusan National University, Busan 46241, South Korea
lehuong7885@gmail.com

[2] School of Computer Science and Engineering, Pusan National University, Busan 46241, South Korea
wulansilvani@gmail.com,androaprila@pusan.ac.kr,flyho77@naver.com, jhcho0208@gmail.com, howonkim@pusan.ac.kr

Abstract. Reliable multi-class intrusion detection across heterogeneous network environments remains difficult because class distributions, feature semantics, and attack patterns change substantially between enterprise traffic and wireless sensor networks (WSNs). We first formalize this setting as four connected sub-problems: imbalanced multi-class classification, domain-conditional feature representation, directional traffic-asymmetry modelling, and Pareto-efficient accuracy–efficiency model selection. Guided by this formulation, we present a reproducible ensemble framework combining Random Forest (RF), Extra Trees (ET), XGBoost (XGB), LightGBM (LGB), and CatBoost (CAT) with domain-aware feature engineering, robust normalization, and training-only SMOTE-based imbalance correction, evaluated on UNSW-NB15 (10-class, severely imbalanced) and WSN-DS (5-class WSN routing attacks). RF obtains the best UNSW-NB15 Macro-F1 of 82.86%, while LGB reaches 99.94% Macro-F1 on WSN-DS and offers the strongest accuracy–efficiency trade-off. Against five deep learning baselines, the ensembles are markedly more reliable on the imbalanced tabular setting, improving UNSW-NB15 Macro-F1 by 28.6 points over the best neural baseline.

Keywords: Network Intrusion Detection · Ensemble Learning · Random Forest · LightGBM · SMOTE · UNSW-NB15 · WSN-DS · Class Imbalance

1 Introduction

Networked infrastructure increasingly combines high-throughput enterprise networks, Internet-of-Things devices, and resource-constrained WSNs. This heterogeneity enlarges the attack surface and weakens the assumptions behind single-environment intrusion detection systems (IDSs) [15,23]. Machine-learning-based

network IDSs are attractive because they can generalize from traffic observations rather than rely only on manually written signatures [6,17]. However, practical multi-class IDS deployment remains constrained by three connected issues.

First, severe class imbalance causes conventional learners to optimize majority-class accuracy while missing rare but security-critical classes such as backdoors, shellcode, and worms [22]. Second, enterprise flow records and WSN routing-layer records have different feature semantics, so a model that is effective in one environment may not be the best choice in another. Third, raw traffic features do not always expose the source–destination asymmetry produced by scanning, flooding, and one-sided attack sessions. These limitations motivate a pipeline that is both dataset-aware at the feature level and unified at the model-evaluation level.

Tree-based ensembles are a strong candidate for this setting because they combine non-linear decision boundaries, robust handling of tabular features, and interpretable feature importance. Yet many IDS studies evaluate only one benchmark, omit systematic feature engineering, leave the underlying imbalance and feature-domain problems informal, or skip comparison against modern deep tabular models under matched preprocessing. This paper addresses those gaps with the following contributions:

- We propose a configurable ensemble IDS pipeline (RF, ET, XGB, LGB, CAT) under one preprocessing/evaluation harness and provide an explicit algorithmic specification for training, validation, and deployment; this harness instantiates the domain-conditional feature map of Eq. (2);
- We introduce twelve source-destination asymmetry features (ratios, differences, totals, duration-normalized intensity) whose discriminative signal (Eq. (3)) we verify post hoc via feature importance;
- We conduct a cross-environment evaluation on UNSW-NB15 and WSN-DS with exact metrics, confusion matrices, per-class analysis, feature importance, and timing/complexity figures that expose the Pareto-efficient model subset of Eq. (4); and
- We conduct a controlled comparison with MLP, DNN, 1D-CNN, TabTransformer, and FT-Transformer baselines, positioned against concurrent 2024–2026 WSN/IoT security and routing studies (Sect. 2) that target fusion-based or federated detectors and energy-aware routing rather than a unified cross-environment ensemble comparison.

2 Related Work

Early IDS studies established that no single classifier dominates all attack categories, motivating ensemble-based detection [26]. Buczak and Guven [6] identify class imbalance, outdated benchmarks, and inconsistent evaluation protocols as central barriers to deployable ML-based IDS. UNSW-NB15 was introduced to address limitations of older datasets and has become a common benchmark for both binary and multi-class intrusion detection [18,19]. Representative multi-class results include RepTree at 81.28% accuracy [3] and ANN+XGBoost at 77.51%

accuracy [13]; however, these studies often omit Macro-F1, which is essential for assessing minority classes. WSN-DS provides a complementary routing-layer benchmark for WSN attacks [2], and earlier WSN IDS work has shown that tree-based methods can be effective for DoS and routing attack detection [16,21].

Deep learning models such as CNNs, recurrent networks, and transformer-style tabular architectures have been applied to IDS [25,12,11]. These models can perform well when abundant labels and stable feature distributions are available, but tabular IDS data are often imbalanced and heterogeneous. Empirical surveys report that tree-based ensembles remain highly competitive in such settings, particularly when class imbalance and interpretability matter [1]. The present work therefore evaluates both families under the same experimental protocol.

Recent 2024–2026 studies extend WSN/IoT security through fusion, federated, and hybrid detectors: Zhang et al. [29] aggregate local WSN models, Bukhari et al. [4] combine SCNN–Bi-LSTM with federated learning, and Villafranca and Cano [27] stack DNNs with XGBoost. Current routing work also emphasizes topology and energy variables, e.g., ETX optimization and fuzzy unequal clustering [24,28]. These studies do not jointly benchmark tree ensembles and deep tabular models on enterprise-flow and WSN-routing datasets under one preprocessing harness, which is the gap targeted here.

3 Proposed Framework

The proposed pipeline has five stages: data ingestion and feature engineering, preprocessing, stratified splitting with training-only imbalance correction, ensemble training, and evaluation. Figure 1 summarizes the data flow and makes explicit that SMOTE is applied only to the training branch, not to validation or test data.

Algorithm 1 gives the executable logic behind Fig. 1. The key safeguards are that encoders, scalers, and SMOTE are fitted only on the training split, and that the final model is selected under the Macro-F1–efficiency trade-off defined later in Eq. (4).

Algorithm 1 Domain-aware ensemble IDS training and deployment

Require: labelled dataset $\mathcal{D}$, environment $\mathcal{E} \in \{\text{network}, \text{WSN}\}$, model set $\mathcal{M} = \{\text{RF}, \text{ET}, \text{XGB}, \text{LGB}, \text{CAT}\}$
Ensure: selected detector f_{m^*}, fitted preprocessing objects, and test metrics
Build domain representation $\mathbf{z}_i \leftarrow \phi(\mathbf{x}_i; \mathcal{E})$ using Eq. (2)
Split $\{(\mathbf{z}_i, y_i)\}$ into stratified train/validation/test partitions
Fit categorical encoders and robust scaling statistics on training data only
Transform validation and test data with the fitted training objects
Apply SMOTE only to the transformed training partition
for each model $m \in \mathcal{M}$ **do**
    Train f_m with class weights $w_c = N/(K \cdot N_c)$ when supported
    Measure Macro-F1, accuracy, balanced accuracy, training time, inference time, and model size
end for
Select m^* from the Pareto-efficient subset in Eq. (4); use Macro-F1 as the primary criterion
Evaluate f_{m^*} on the untouched test partition and generate class-level and feature-importance evidence
return f_{m^*} and the fitted preprocessing objects for deployment on future records

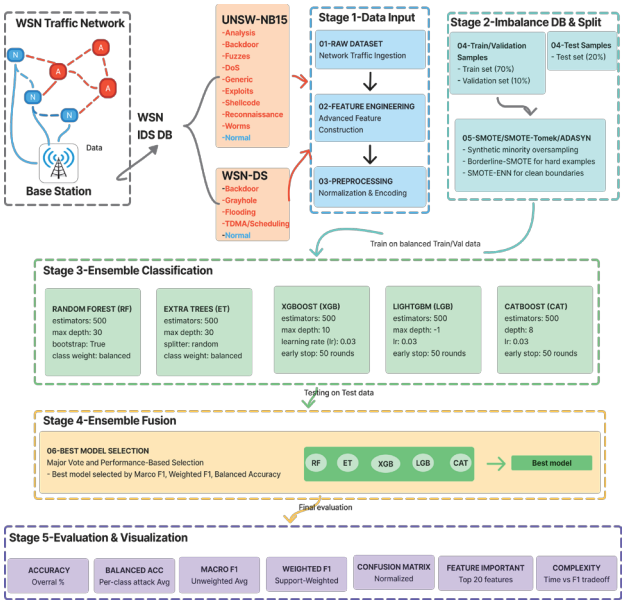


Fig. 1: Proposed multi-class ensemble IDS pipeline. The same harness is used for both datasets, while feature engineering is activated when source–destination flow fields are available.

3.1 Problem Statement

We formalize cross-environment intrusion detection as four connected sub-problems, matching the issues raised in Sect. 1.

(P1) Imbalanced multi-class classification. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{1, \ldots, K\}$ (K=10 for UNSW-NB15, K=5 for WSN-DS), with class sizes $N_c = |\{i : y_i = c\}|$. The dataset is imbalanced when $\rho = \max_c N_c / \min_c N_c \gg 1$ (Eq. 1), reaching $\rho \approx 535$ on UNSW-NB15 (Table 1). We therefore train $f_\theta : \mathbb{R}^d \to \{1, \ldots, K\}$ to maximize Macro-F1 rather than accuracy, since accuracy is dominated by majority classes under large ρ:

$$\rho = \frac{\max_c N_c}{\min_c N_c}, \qquad \theta^* = \arg\max_\theta \; \frac{1}{K} \sum_{c=1}^{K} F1_c(\theta), \qquad F1_c = \frac{2\, P_c R_c}{P_c + R_c}, \tag{1}$$

where P_c, R_c are class-c precision and recall on a held-out split.

(P2) Domain-conditional feature representation. UNSW-NB15 and WSN-DS expose disjoint raw feature spaces $\mathbf{x}^{\text{net}} \in \mathbb{R}^{d_1}$ (flow fields, e.g. $s_{\text{bytes}}, d_{\text{bytes}}$) and $\mathbf{x}^{\text{wsn}} \in \mathbb{R}^{d_2}$ (sensor-topology/energy fields), $d_1 \neq d_2$. We define a domain-conditional feature map

$$\phi(\mathbf{x}; \mathcal{E}) = \begin{cases} \left[\, \mathbf{x}^{\text{net}},\ g(\mathbf{x}^{\text{net}}) \,\right], & \mathcal{E} = \text{network}, \\ \mathbf{x}^{\text{wsn}}, & \mathcal{E} = \text{WSN}, \end{cases} \tag{2}$$

where $\mathcal{E}$ is the environment indicator and $g(\cdot)$ is the source–destination asymmetry transform of Eq. (5), so the same classifier family operates on $\phi(\mathbf{x};\mathcal{E})$ for either environment.

(P3) Directional traffic asymmetry. Scanning and flooding attacks produce structurally asymmetric source/destination volumes, whereas benign sessions are closer to bidirectional balance. We quantify this with the deviation of the ratio terms in Eq. (5) from 1:

$$\Delta(\mathbf{x}) = \left|\, bytes_ratio - 1 \,\right| + \left|\, pkts_ratio - 1 \,\right|, \tag{3}$$

expected to be small for Normal traffic and large for one-sided attack sessions; this is later verified empirically via feature importance (Sect. 5) rather than assumed.

(P4) Accuracy–efficiency model selection. No single classifier is jointly optimal in Macro-F1, training time $T(\theta)$, inference time $I(\theta)$, and model size $M(\theta)$. We seek the Pareto-efficient subset

$$\mathcal{P} = \left\{ \theta \,\middle|\, \nexists\, \theta' \text{ s.t. Macro-F1}(\theta') \geq \text{Macro-F1}(\theta) \,\wedge\, \big(T, I, M\big)(\theta') \preceq \big(T, I, M\big)(\theta) \right\}, \tag{4}$$

with $\preceq$ component-wise $\leq$ and at least one strict inequality; $\mathcal{P}$ is reported empirically via Fig. 5 rather than assumed analytically, since T, I, M depend on implementation and hardware.

P1–P2 motivate the preprocessing and feature design (Sect. 3), P3 is checked via feature importance and P4 via the timing/complexity comparison (Sect. 5).

3.2 Feature Engineering and Preprocessing

Raw flow records are heavy-tailed and often hide the directional imbalance generated by attacks. For UNSW-NB15 flow features, we apply $\log(1+x)$ to highly skewed continuous fields (*dur*, *sbytes*, *dbytes*, *sload*, *dload*, *rate*) and construct twelve compound features:

$$\begin{aligned}
bytes_ratio &= \frac{s_{\text{bytes}}}{d_{\text{bytes}}+1}, & bytes_diff &= s_{\text{bytes}} - d_{\text{bytes}},\\
total_bytes &= s_{\text{bytes}} + d_{\text{bytes}}, & load_ratio &= \frac{s_{\text{load}}}{d_{\text{load}}+1},\\
load_diff &= s_{\text{load}} - d_{\text{load}}, & pkts_ratio &= \frac{s_{\text{pkts}}}{d_{\text{pkts}}+1},\\
pkts_diff &= s_{\text{pkts}} - d_{\text{pkts}}, & total_pkts &= s_{\text{pkts}} + d_{\text{pkts}},\\
ttl_diff &= s_{\text{ttl}} - d_{\text{ttl}}, & window_ratio &= \frac{s_{\text{win}}}{d_{\text{win}}+1},\\
bytes_per_pkt &= \frac{total_bytes}{total_pkts+1}, & bytes_per_dur &= \frac{total_bytes}{dur+1}.
\end{aligned} \tag{5}$$

The +1 smoothing term prevents division by zero for one-sided sessions, a frequent pattern in scanning and flooding. Categorical variables are encoded within

the training split and numeric variables are scaled with `RobustScaler`; the fitted statistics are then applied to validation and test data to avoid leakage.

Class imbalance, quantified by ρ in Eq. (1), is corrected with SMOTE [7], but only after the stratified split and only on the training partition. Validation and test partitions therefore remain an unbiased evaluation surface for the Macro-F1 objective θ^* in Eq. (1). Class weights $w_c = N/(K \cdot N_c)$ are additionally supplied to classifiers that support them, giving rarer classes proportionally higher loss contribution during training.

3.3 Ensemble Classifiers and Metrics

Five tree-based models are evaluated in the same harness: RF [5], ET [9], XGB [8], LGB [14], and CAT [20]. All models use 500 estimators unless the underlying implementation controls the number of boosting rounds separately. The primary metric is Macro-F1, the unweighted mean of per-class F1 scores, because it gives equal weight to rare and frequent classes. Accuracy and balanced accuracy are reported as secondary metrics.

4 Datasets

UNSW-NB15 [18] contains 257,673 labelled records across 10 classes in the official train/test partitions. Table 1 corrects the class-count logic by using the combined partition counts consistently. The three largest classes represent 76.2% of all records, whereas the four rarest classes account for only 2.6%. This makes Macro-F1 more informative than accuracy alone.

Table 1: UNSW-NB15 class distribution in the official train+test partitions.

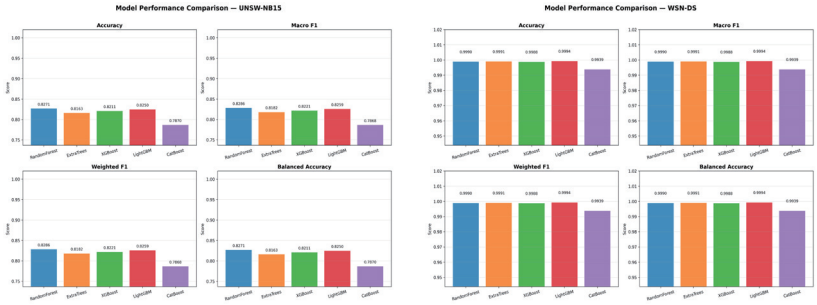
Class	**Count**	**Type**	**Class**	**Count**	**Type**
Normal	93,000	Benign	Reconnaissance	13,987	Attack
Generic	58,871	Attack	Analysis	2,677	Attack
Exploits	44,525	Attack	Backdoor	2,329	Attack
Fuzzers	24,246	Attack	Shellcode	1,511	Attack
DoS	16,353	Attack	Worms	174	Attack

WSN-DS [2] is used as a complementary WSN routing-layer benchmark. It contains five labels: Normal, Blackhole, Grayhole, Flooding, and TDMA scheduling violations. Its sensor-specific features, such as cluster size, cluster-head identity, and expended energy, describe different mechanisms from enterprise flow statistics. Accordingly, the experiments should be interpreted as within-dataset evaluations under a common pipeline, not as direct cross-dataset transfer learning.

5 Experimental Results

5.1 Overall Ensemble Performance

Fig. 2 visualizes the result ranking across all reported metrics. RF is the best UNSW-NB15 model with 0.8286 Macro-F1; LGB is close behind at 0.8259. On WSN-DS, LGB leads with 0.9994 Macro-F1. The two datasets therefore select different best models, which supports the need for a configurable rather than single-model IDS pipeline.



(a) UNSW-NB15 confusion matrix (b) WSN-DS confusion matrix

Fig. 2: Visual comparison of ensemble accuracy, Macro-F1, weighted F1, and balanced accuracy. The figure connects the numerical ranking in Table 2 to the result plots supplied with the experiments.

5.2 Comparison with Deep Learning Baselines

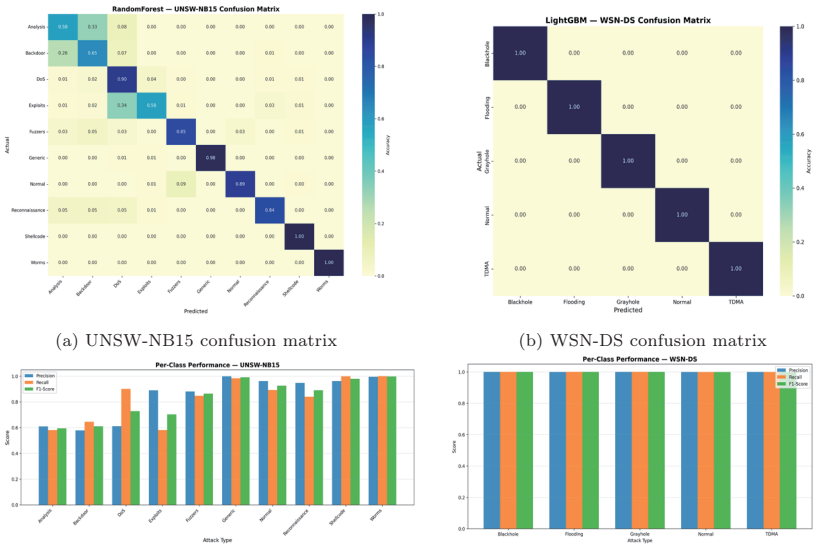
Table 2 compares the ensembles with MLP, DNN, 1D-CNN, TabTransformer (TT) [12], and FT-Transformer (FTT) [11]. All neural models use Adam with learning rate 10^{-3}, cosine annealing, batch size 512, early stopping, and class-weighted cross-entropy. The UNSW-NB15 gap is the key result: RF improves Macro-F1 by 28.6 percentage points over the best neural model (CNN, 0.5430). On WSN-DS, neural models obtain high accuracy but still trail the best ensemble in Macro-F1.

5.3 Per-Class Behaviour and Error Structure

Figure 3 links aggregate scores to class-level behaviour. On UNSW-NB15, the diagonal is strong for Generic, Shellcode, and Worms, but Analysis, Backdoor, and Exploits show more confusion because their low-volume traffic patterns are less separable. The per-class bars show that the aggregate Macro-F1 is limited by these minority or overlapping classes rather than by the majority Normal and Generic categories. On WSN-DS, both the confusion matrix and per-class bars are almost perfect, confirming that the sensor-specific features produce clearer class separation.

Table 2: Ensembles versus deep learning baselines. Best Macro-F1 per dataset in **bold**; best neural value is underlined.

Type	Model	UNSW-NB15			WSN-DS		
		Acc.	Macro-F1	AUC	Acc.	Macro-F1	AUC
Ensemble	RF	**0.8271**	**0.8286**	–	0.9990	0.9990	–
	ET	0.8163	0.8182	–	0.9991	0.9991	–
	XGB	0.8211	0.8221	–	0.9988	0.9988	–
	LGB	0.8250	0.8259	–	**0.9994**	**0.9994**	–
	CAT	0.7870	0.7868	–	0.9939	0.9939	–
Deep Learning	MLP	0.7480	0.4280	0.9542	0.9892	0.9120	0.9955
	DNN	0.7439	0.3960	0.9548	0.9797	<u>0.9140</u>	0.9940
	CNN	<u>0.7629</u>	<u>0.5430</u>	0.9552	0.9831	0.9090	0.9955
	TT	0.7226	0.3900	0.9398	0.9823	0.9070	0.9944
	FTT	0.6654	0.3510	0.9306	0.9706	0.8490	0.9897



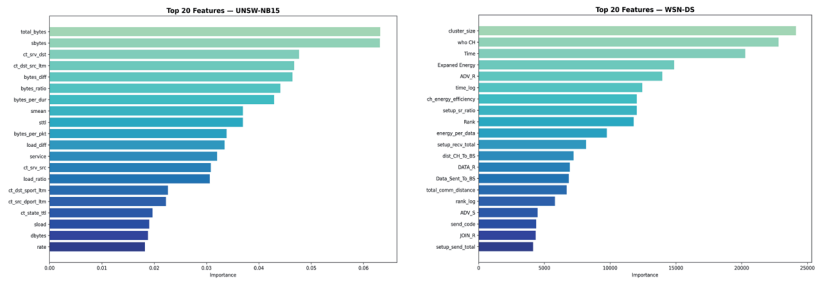
(a) UNSW-NB15 confusion matrix (b) WSN-DS confusion matrix

(c) UNSW-NB15 per-class precision/recall/F1 (d) WSN-DS per-class precision/recall/F1

Fig. 3: Class-level analysis for the best model on each dataset. This figure supports the claim that UNSW-NB15 remains minority-class limited, whereas WSN-DS is highly separable under the evaluated feature representation.

5.4 Feature Importance

The feature-importance plots in Fig. 4 explain why the same pipeline behaves differently across datasets. For UNSW-NB15, engineered asymmetry features such as *total_bytes*, *bytes_diff*, *bytes_ratio*, and *bytes_per_dur* appear among the highest-ranked features, confirming that source–destination traffic imbalance is discriminative. For WSN-DS, the dominant features are sensor-topology and energy variables such as cluster size, cluster-head identity, and expended energy; the wired-flow asymmetry features are not the source of WSN performance. This prevents an overgeneralized conclusion and clarifies that the feature engineering benefit is domain-specific.



(a) UNSW-NB15 (b) WSN-DS

Fig. 4: Top-20 feature importance for the best-performing model on each dataset. The plots support the feature-engineering claim rather than leaving it as a textual assertion.

5.5 Computational Complexity and Timing

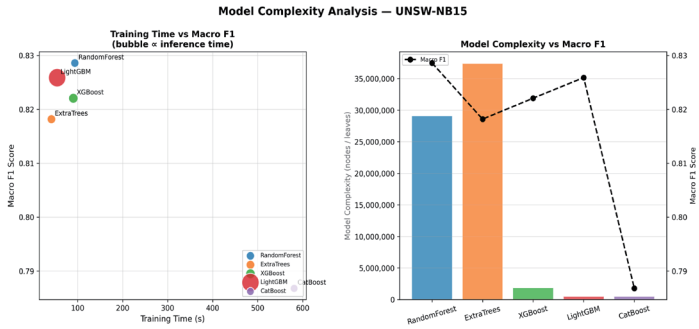
The timing discussion is now tied directly to Fig. 5. On UNSW-NB15, RF gives the highest Macro-F1, but LGB lies on a more efficient Pareto region because it reaches similar Macro-F1 with lower training time and substantially smaller model complexity. CAT is dominated in this setting because it is both slower and less accurate. On WSN-DS, LGB again offers the best accuracy–efficiency trade-off. These plots support the deployment recommendation: RF is preferred when maximum UNSW-NB15 Macro-F1 is the goal, whereas LGB is preferred when latency, model size, or WSN edge deployment are important.

5.6 Comparison with Published UNSW-NB15 Baselines

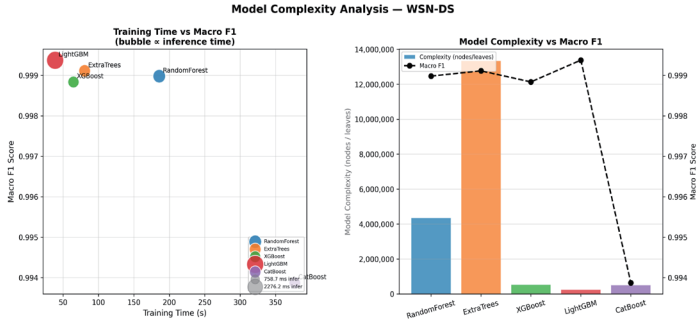
Table 3 positions the proposed result against representative published UNSW-NB15 baselines. Because earlier works often report only accuracy, the comparison is conservative: the proposed method is compared on accuracy where possible, and Macro-F1 is reported to expose minority-class behaviour.

6 Discussion and Limitations

The results indicate that tree-based ensembles are well matched to imbalanced tabular IDS data. Their node-level splitting can isolate locally informative minority-



(a) UNSW-NB15



(b) WSN-DS

Fig. 5: Pareto model complexity and timing analysis. In each panel, training time is plotted against Macro-F1 and bubble size represents inference time; the bar-line chart relates model complexity to Macro-F1.

Table 3: Comparison with representative UNSW-NB15 baselines. †Binary result is shown for reference only.

Reference	Method	Setting	Acc. (%)	Macro-F1 (%)
Moustafa & Slay [19]	ANN	Binary†	81.34	–
Belouch et al. [3]	RepTree	Multi-class	81.28	–
Kasongo & Sun [13]	ANN+XGBoost	Multi-class	77.51	–
Proposed	**RF ensemble pipeline**	**Multi-class**	**82.71**	**82.86**

class regions, whereas neural models trained with global mini-batch gradients still struggle with Analysis, Backdoor, and other low-frequency UNSW-NB15 categories. The contrast between UNSW-NB15 and WSN-DS also shows that a single headline accuracy is insufficient: WSN-DS is close to saturated under the available sensor features, while UNSW-NB15 remains limited by overlapping attack semantics.

The study has three limitations: experiments are within-dataset rather than direct cross-dataset transfer tests; feature engineering is flow-level and does not use raw payload or temporal sequence models; and SMOTE may create vectors that are not always behaviourally realistic for semantically specific attacks. Future work will evaluate stacked ensembles, concept-drift adaptation [10], CIC-IDS-2017, and edge deployment of LGB on WSN cluster-head hardware.

7 Conclusion

We presented a unified ensemble learning framework for multi-class intrusion detection across enterprise and WSN environments, grounded in an explicit formalization of imbalanced classification, domain-conditional feature representation, traffic asymmetry, and accuracy–efficiency model selection. The revised paper makes the logical chain explicit: aggregate metrics are supported by model-comparison plots, minority-class claims are supported by confusion matrices and per-class bars, feature-engineering claims are supported by importance plots, and timing claims are supported by the complexity figures. RF is the strongest UNSW-NB15 detector with 0.8286 Macro-F1, while LGB is the best WSN-DS and efficiency-oriented model with 0.9994 Macro-F1. Compared with five deep learning baselines and with concurrent 2024–2026 fusion-based, federated, and routing-focused WSN/IoT security studies, the ensemble pipeline provides a stronger and more transparent baseline for imbalanced tabular IDS research.

Acknowledgments. This work was supported by the Korea Institute of Energy Technology Evaluation and Planning(KETEP) grant funded by the Korea government(MOTIE) (RS-2024-00423446, Demonstration of Technology for Efficiency Improvement in Entire Cycle of Manufacturing Processes of Root industries for Small and Medium-sized Enterprises).

References

1. Ahmad, Z., Khan, A.S., Shiang, C.W., Abdullah, J., Ahmad, F.: Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. **32**, e4150 (2021)
2. Almomani, I., Al-Kasasbeh, B., Al-Akhras, M.: WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. J. Sensors **2016**, 4731953 (2016)
3. Belouch, M., El Hadaj, S., Idhammad, M.: A two-stage classifier approach using RepTree algorithm for network intrusion detection. Int. J. Adv. Comput. Sci. Appl. **8**(6) (2017)

4. Bukhari, S.M.S., Zafar, M.H., Abou Houran, M., Moosavi, S.K.R., Mansoor, M., Muaaz, M., Sanfilippo, F.: Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. Ad Hoc Netw. **155**, 103407 (2024)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
6. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. **18**(2), 1153–1176 (2016)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
8. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proc. 22nd ACM SIGKDD, pp. 785–794 (2016)
9. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
10. Gomes, H.M. et al.: Adaptive random forests for evolving data stream classification. Mach. Learn. **106**(9–10), 1469–1495 (2017)
11. Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A.: Revisiting deep learning models for tabular data. In: NeurIPS (2021)
12. Huang, X., Khetan, A., Cvitkovic, M., Karnin, Z.: TabTransformer: Tabular data modeling using contextual embeddings. arXiv:2012.06678 (2020)
13. Kasongo, S.M., Sun, Y.: Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J. Big Data **7**(1) (2020)
14. Ke, G. et al.: LightGBM: A highly efficient gradient boosting decision tree. In: NeurIPS (2017)
15. Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S.: Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. IEEE Commun. Surv. Tutor. **18**(1), 184–208 (2016)
16. Le, T.T.H., Park, T., Cho, D., Kim, H.: An effective classification for DoS attacks in wireless sensor networks. In: Proc. ICUFN 2018, pp. 689–692 (2018)
17. Liao, H.J., Lin, C.H.R., Lin, Y.C., Tung, K.Y.: Intrusion detection system: A comprehensive review. J. Netw. Comput. Appl. **36**(1), 16–24 (2013)
18. Moustafa, N., Slay, J.: UNSW-NB15: A comprehensive data set for network intrusion detection systems. In: MilCIS 2015, pp. 1–6 (2015)
19. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J.: A Glob. Perspect. **25**(1–3), 18–31 (2016)
20. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: Unbiased boosting with categorical features. In: NeurIPS (2018)
21. Raza, S., Wallgren, L., Voigt, T.: SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw. **11**(8), 2661–2674 (2013)
22. Ren, H., Tang, Y., Dong, W., Ren, S., Jiang, L.: DUEN: Dynamic ensemble handling class imbalance in network intrusion detection. Expert Syst. Appl. **229**, 120420 (2023)
23. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A survey of network-based intrusion detection data sets. Comput. Secur. **86**, 147–167 (2019)
24. Senouci, O., Benaouda, N.: Supervised machine learning-based ETX optimization for energy-efficient routing in IoT-enabled WSNs. Ad Hoc Netw. **178**, 103972 (2025)
25. 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: IEEE NetSoft 2018, pp. 202–206 (2018)

26. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Intrusion detection by machine learning: A review. Expert Syst. Appl. **36**(10), 11994–12000 (2009)
27. Villafranca, A., Cano, M.D.: A hybrid inference pipeline for IDS: Combining DNNs and XGBoost through stacking for real-world intrusion detection. Ad Hoc Netw. **188**, 104227 (2026)
28. Zaier, A., Lahmar, I., Yahia, M., Lloret, J.: Interval type 2 fuzzy unequal clustering and sleep scheduling for IoT-based WSNs. Ad Hoc Netw. **175**, 103867 (2025)
29. Zhang, H., Upadhyay, D., Zaman, M., Jain, A., Sampalli, S.: SC-MLIDS: Fusion-based machine learning framework for intrusion detection in wireless sensor networks. Ad Hoc Netw. **175**, 103871 (2025)

Skip-VFL: Collaborative Intrusion Detection Across Feature-Fragmented Edge Networks via Vertical Federated Learning

Alimov Abdulboriy Abdulkhay ugli[1] and Ji Sun Shin[2]

[1] Department of Computer and Information Security, and the Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea adamalimov98@gmail.com
[2] Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea jsshin@sejong.ac.kr

Abstract. Distributed network environments increasingly rely on collaborative machine learning for threat intelligence; however, edge devices capture fragmented, heterogeneous feature spaces that conventional Federated Learning cannot aggregate. Furthermore, deploying these systems for Network Intrusion Detection is severely hindered by extreme data imbalances. To overcome these constraints, we propose `Skip-VFL`, a resilient Vertical Federated Learning architecture designed for collaborative anomaly detection across feature-fragmented networks. `Skip-VFL` organizes distributed telemetry into shared core, overlapping, and unique feature sets. To bridge architectural gaps without exposing raw data, a novel **Skip-Layer Mechanism** aligns distributed latent spaces while retaining tier-specific adapters (ϕ_k) locally, ensuring stringent privacy and reduced communication overhead. To mitigate extreme class imbalances, `Skip-VFL` employs a two-stage distributed training protocol: **Latent Space Alignment** for robust representation clustering, followed by **Weighted Classification** utilizing Inverse Class Frequency (ICF) to amplify sensitivity to rare network anomalies. Extensive evaluations on the NSL-KDD, UNSW-NB15, and ToN-IoT datasets demonstrate that Skip-VFL avoids the precision collapse exhibited by conventional cross-entropy baselines under severe imbalance, achieving anomaly recall as high as 89.1% and accuracy up to 99.2% on select benchmarks under 1% class imbalance while maintaining robust convergence.

Keywords: Vertical Federated Learning · Network Intrusion Detection · Extreme Class Imbalance · Prototype Learning

1 Introduction

The rapid proliferation of Internet-of-Things (IoT) devices and increasingly complex network infrastructures has significantly expanded the attack surface of modern distributed systems, necessitating robust and scalable Network Intrusion Detection Systems (NIDS) [17]. Traditional NIDS rely on centralized data

collection and processing, which is increasingly impractical due to privacy regulations such as GDPR and CCPA [18]. Federated Learning (FL) [9] addresses this challenge by enabling collaborative model training without sharing raw data.

However, conventional Horizontal Federated Learning (HFL) [22] assumes homogeneous feature spaces across participants, an assumption that rarely holds in modern edge networks. In practice, network observations are vertically partitioned across heterogeneous devices, such as edge sensors, gateways, and security servers, each capturing distinct subsets of features. Vertical Federated Learning (VFL) [10] provides a suitable framework for such environments, but its application to network intrusion detection remains challenged by two fundamental issues: *feature heterogeneity* and *extreme class imbalance*.

First, network telemetry is inherently heterogeneous, with different monitoring nodes observing diverse feature subsets. Existing VFL approaches [8] generally assume flat feature structures, resulting in inefficient representation learning and increased communication overhead. Second, intrusion detection datasets exhibit severe class imbalance [16], where rare attack instances are easily overshadowed by abundant benign traffic, causing models to converge toward majority-class solutions and significantly reducing anomaly recall.

To address these challenges, we propose `Skip-VFL`, a tiered Vertical Federated Learning framework for collaborative anomaly detection across feature-fragmented edge networks. `Skip-VFL` introduces a *Skip-Layer* mechanism that preserves privacy by withholding client-specific feature adapters while enabling global representation learning. Additionally, we propose a two-stage optimization strategy consisting of prototype-based latent space alignment followed by inverse class frequency (ICF) weighted classification to improve minority anomaly detection under severe imbalance conditions.

Extensive experiments on NSL-KDD [1], UNSW-NB15 [13], and ToN-IoT [12] demonstrate that `Skip-VFL` achieves robust and stable performance under highly imbalanced and heterogeneous settings.

The main contributions of this work are summarized as follows:

- We propose `Skip-VFL`, a tiered VFL architecture that supports collaborative learning over heterogeneous feature spaces while preserving client-specific feature distributions through a novel Skip-Layer mechanism.
- We introduce a two-stage training framework combining prototype-based representation learning and inverse class frequency weighting to mitigate severe class imbalance in distributed intrusion detection.
- We evaluate `Skip-VFL` on three benchmark intrusion detection datasets, demonstrating improved anomaly recall, stability, and convergence under extreme imbalance conditions.

2 Preliminaries

2.1 Vertical Federated Learning (VFL)

Vertical Federated Learning (VFL) addresses scenarios where participating entities hold distinct feature columns over a shared sample ID space [10]. Traditional

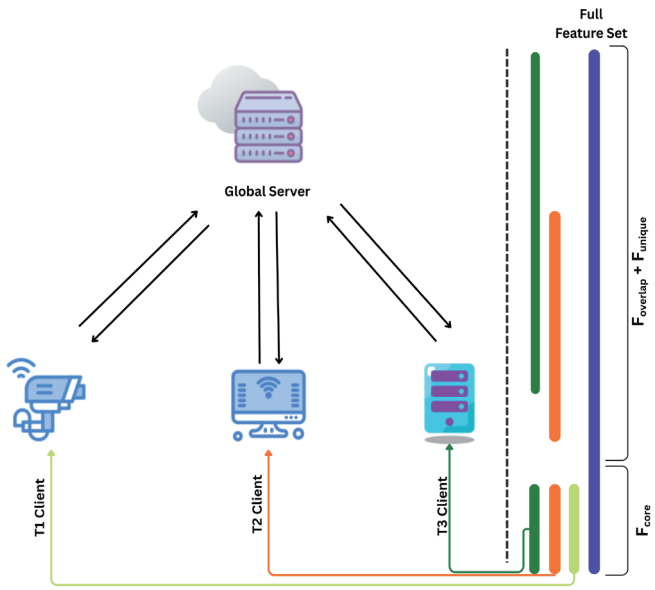


Fig. 1. Overview of Tiered Feature Partitioning ($\mathcal{F}_{\text{core}}$, $\mathcal{F}_{\text{overlap}}$, $\mathcal{F}_{\text{unique}}$) between Clients in `Skip-VFL` Framework.

VFL operates under an active–passive paradigm: a single active party possesses ground-truth labels and drives supervision, while passive parties hold unlabelled features.

Our framework departs from this layout. Unlike canonical single-active-party VFL, Skip-VFL adopts a multi-label-owner relaxation [14], reflecting the self-labeling nature of distributed network telemetry. This setup reflects realistic edge conditions where distributed nodes (e.g., IoT gateways) generate independent, local ground-truth annotations.

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the federation of clients. Each client $k \in \mathcal{K}$ holds a local dataset:

$$\mathcal{D}_k = \left\{ (\mathbf{x}_k^{(i)}, y^{(i)}) \right\}_{i=1}^{N_k} \tag{1}$$

where $\mathbf{x}_k^{(i)} \in \mathbb{R}^{d_k}$ is the local feature vector, d_k is the local feature dimension, and $y^{(i)} \in \mathcal{Y}$ is the corresponding class label. Let $\mathcal{F}$ represent the global feature space with total cardinality $D = |\mathcal{F}|$. Each client observes a partial subset $\mathcal{F}_k \subseteq \mathcal{F}$ such that $d_k = |\mathcal{F}_k|$, where feature dimensions across clients can partially overlap as illustrated in Figure 1. Clients train collaboratively by communicating only intermediate latent representations or localized model updates to the server, preserving raw feature privacy.

2.2 Supervised Contrastive Learning Foundations

Contrastive representation learning maps data into a high-dimensional metric space by pulling positive pairs closer while pushing negative pairs apart [4, 5].

Supervised Prototypes Loss As a core regularizer within `Skip-VFL`, the Supervised Prototypes objective eliminates the quadratic instance-to-instance comparison overhead inherent to standard contrastive pipelines [11]. Each client embedding z_i is mapped directly against learnable, global class-conditional anchors $\mathcal{P} = \{P_1, \ldots, P_C\}$ representing the class centroids:

$$\mathcal{L}_{\text{proto}} = -\log \frac{\exp(z_i \cdot P_{y_i}/\tau)}{\sum_{j=1}^{C} \exp(z_i \cdot P_j/\tau)} \tag{2}$$

This establishes a stable, high-density target space that stops minority anomaly trajectories from collapsing into the dominant benign distribution.

3 Related Work

Collaborative anomaly detection over distributed networks must navigate two compounding structural bottlenecks: input feature heterogeneity across decentralized silos and severe representation skew caused by sparse anomaly instances.

Existing Vertical Federated Learning (VFL) security architectures generally address feature mismatch and topological fragmentation using dimension adapters, structural model slicing, or split learning variants.

- **Dimension Adapters and Slicing:** Yadav et al. [21] utilize Autoencoders (AEs) as dimension-matching adapters to compress private client samples into a static optimal latent space l^*. Similarly, Zhao et al. [23] propose a dual-AE paradigm split across client-side (bottom) and server-side (front) structures to handle privacy-preserving anomaly tracking. To bypass uniform dimension requirements entirely, Novikova et al. [15] execute model slicing, partitioning the global model into specialized feature columns mapped directly to local client properties.
- **Hybrid Security Paradigms:** Beyond standard neural architectures, Kadhe et al. [6] evaluate vertically trained Random Forests utilizing privacy-preserving ensemble learning methods. To mitigate label scarcity across fragmented spaces, *FedHSSL* [3] introduces a semi-supervised knowledge propagation pipeline designed for asymmetric configurations where a single active silo holds the ground-truth annotations. Conversely, Kesici et al. [7] combine conventional VFL with Split Learning to secure nodes against false data injection attacks, anchoring label evaluation strictly at the server level.

Class distribution skews dramatically degrade federated convergence bounds if left unmitigated, leading to representation collapse on minority distributions [20].

Algorithmic remedies typically re-engineer empirical risk objectives or map features to shared anchor spaces.

Wang et al. [19] address this via a ratio loss function that dynamically scales global model optimization weights based on individual client sample distributions. Closer to our metric alignment strategy, *Proto-EVFL* [2] establishes a state-of-the-art prototype-based VFL architecture. *Proto-EVFL* utilizes a probabilistic dual prototype learning scheme that matches unaligned sample features to global centroids by calculating optimal transport costs guided by historical class priors.

3.1 Summary of Departure

While the aforementioned frameworks deliver robust solutions for either structural feature matching or label imbalance in isolation, they generally treat these compounding anomalies as independent problems. Furthermore, many contemporary VFL platforms remain bound to an active-passive labeling constraint. In contrast, `Skip-VFL` introduces a unified, tiered architecture that explicitly solves arbitrary cross-client feature fragmentation while stabilizing the underlying manifold against extreme, long-tailed anomaly boundaries ($\leq 1\%$) without imposing centralized label dependencies.Unlike conventional VFL settings where labels reside at a single party, we assume locally available binary labels at each participating node to reflect autonomous intrusion detection deployments.

4 Methodology

This section presents the technical details of the `Skip-VFL` framework. We focus on the multi-tiered ($T1, T2, T3$) architectural design, the novel aggregator mechanism that handles feature heterogeneity, and the two-stage training protocol designed to mitigate extreme class imbalance through supervised contrastive learning. We have provided the detailed workflow in Algorithm 1 and Figure 2

4.1 Problem Formulation and Tiered Feature Partitioning

We define a Vertical Federated Learning (VFL) environment over a distributed network of K heterogeneous clients. Each client $k \in \{1, \ldots, K\}$ maintains a local dataset $\mathcal{D}_k = \{(\mathbf{x}_k^{(i)}, y^{(i)})\}_{i=1}^{N_k}$, where local features $\mathbf{x}_k^{(i)} \in \mathbb{R}^{d_k}$ exhibit high dimensionality variance d_k across network tiers. The local ground-truth labels $y^{(i)} \in \{0, 1\}$ represent a highly skewed binary distribution (0 for benign traffic, 1 for anomalies) subjected to extreme class imbalance boundaries:

$$\frac{N_{y=1}}{N} \leq \gamma, \quad \text{where } \gamma \in \{0.01, 0.05\} \tag{3}$$

To capture the physical fragmentation of real-world telemetry systems, the global feature space $\mathcal{F}$ is separated into three distinct hierarchical pools based on collection complexity:

Algorithm 1 `Skip-VFL`: Two-Stage Training Procedure

Initialize global backbone Ψ, projection head g, classifier Ω, and prototypes $\mathcal{P}$
for each communication round $r = 1, \ldots, R$ **do**
  for each client k in parallel **do**
    Synchronize shared parameters $(\Psi, g, \Omega, \mathcal{P})$
    Construct local model $E_{\theta_k} = \phi_k \circ \psi_k$
    if $r \leq S_1$ **then**
      Generate latent embeddings z_k
      Optimize prototype-based contrastive objective
      Update $\{\phi_k, \psi_k, g_k, \mathcal{P}_k\}$
      Send only $\{\psi_k, g_k, \mathcal{P}_k\}$ to server
    else
      Freeze feature extractor (ϕ_k, ψ_k, g_k)
      Train classifier using ICF-weighted loss
      Send classifier parameters $\{\omega_k\}$ to server
    end if
  end for
  Aggregate shared parameters using weighted averaging
  Broadcast updated global parameters
end for=0

- **Core Features ($\mathcal{F}_{\mathbf{core}}$):** Fundamental network indicators (e.g., protocol types, packet dimensions) captured globally across all monitoring deployments.
- **Overlapping Features ($\mathcal{F}_{\mathbf{overlap}}$):** Intermediate operational metrics (e.g., flow durations, flag metrics) shared between structural monitoring checkpoints and high-performance nodes.
- **Unique Features ($\mathcal{F}_{\mathbf{unique}}$):** High-dimensional, specialized domain signals (e.g., deep packet inspection anomalies, host system logs) restricted to advanced security infrastructures.

Clients map to a **Tiered Client Architecture** composed of three operational classes: Tier 1 (T_1, Edge Devices), Tier 2 (T_2, Gateways), and Tier 3 (T_3, Security Servers). To validate cross-network robustness, we implement **Stochastic Subset Sampling**: local nodes within higher tiers instantiate unique feature footprints by extracting randomized feature slices from the $\mathcal{F}_{\text{overlap}}$ and $\mathcal{F}_{\text{unique}}$ distribution matrices.

4.2 Skip-Layer Aggregation

To resolve input feature dimension variance $d_k = |\mathcal{F}_k|$ across tiers without relying on resource-intensive zero-padding or structural masking matrices, each client decomposes their local encoder network E_{θ_k} into a decoupled, asymmetric dual-component design:

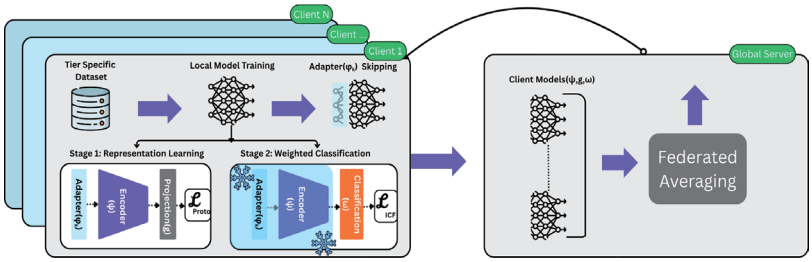


Fig. 2. Two-stage Training Logic and Federated Averaging Pipeline

1. **Local Adapter (ϕ_k):** An isolated initial projection layer $L_1^{(k)} : \mathbb{R}^{d_k} \to \mathbb{R}^h$ that translates heterogeneous raw feature spaces into a uniform global latent width ($h = 512$). This layer is kept strictly private and is permanently withheld from server communication.
2. **Shared Backbone (ψ_k):** Deeper semantic processing blocks $\psi_k : \mathbb{R}^h \to \mathbb{R}^d$ that are structurally isomorphic across all participating nodes in the federation.

The global aggregation operator $\mathcal{A}$ evaluates exclusively over the shared backbone parameters at each synchronization exchange:

$$\Psi_{\text{global}} = \sum_{k=1}^{K} \alpha_k \psi_k, \quad \text{where } \alpha_k = \frac{n_k}{n} \tag{4}$$

By decoupling and isolating ϕ_k, the framework preserves raw telemetry security boundary contexts, supports elastic edge connectivity, and minimizes the global communication payload size over band-limited edge links.

4.3 Prototype-Based Two-Stage Training

To mitigate representation collapse under severe class imbalance, `Skip-VFL` replaces conventional sample-to-sample contrastive learning with a prototype-based objective defined over learnable global class centroids $\mathcal{P} = p_0, p_1 \subset \mathbb{R}^d$, representing benign and anomalous traffic. Given the ℓ_2-normalized embedding $z_i^{(k)} = E_{\theta_k}(\mathbf{x}_i^{(k)})$ generated by client k, the objective reduces to Equation 5,

$$\mathcal{L}_{\text{proto}} = -\log \frac{\exp(z_i^{(k)} \cdot p_{y_i}/\tau)}{\sum_{j=0}^{1} \exp(z_i^{(k)} \cdot p_j/\tau)}, \tag{5}$$

where τ denotes the temperature parameter. This objective simultaneously encourages intra-class compactness and inter-class separation, preventing minority anomaly representations from collapsing into dominant benign regions.

Training proceeds in two stages. During **Stage 1** ($1 \to S_1$), clients jointly optimize the shared backbone and local adapters using $\mathcal{L}_{\text{proto}}$, aligning heterogeneous feature spaces into a common latent manifold. The private adapters ϕ_k remain local, while only shared model parameters are aggregated through the Skip-Layer mechanism.

During **Stage 2** ($S_1 + 1 \to R$), the learned feature extractor is frozen and a lightweight classifier head is optimized using an Inverse Class Frequency (ICF) weighted cross-entropy loss:

$$\mathcal{L}_{\text{ICF}} = -\sum_{j \in 0,1} w_j y_j \log(\hat{y}_j), \qquad w_j = \frac{n}{2n_j}. \tag{6}$$

This decoupled optimization strategy preserves the learned latent geometry while improving minority-class sensitivity with minimal additional communication overhead.

5 Experimental Settings

5.1 Experimental Setup

We evaluate `Skip-VFL` on three benchmark intrusion detection datasets: NSL-KDD [1], UNSW-NB15 [13], and ToN-IoT [12], which collectively represent both traditional enterprise and modern IoT network environments. Dataset statistics after preprocessing are summarized in Table 1.

Table 1. Dataset statistics after preprocessing.

Dataset	Features	Train	Test	Anomaly (%)
NSL-KDD	41	118,813	29,704	48.1
UNSW-NB15	34	206,138	51,535	63.9
ToN-IoT	38	168,834	42,209	76.3

To evaluate robustness under severe data scarcity, we construct two additional imbalance settings with anomaly ratios $\gamma \in \{0.05, 0.01\}$, corresponding to 5% and 1% anomaly prevalence [11].

We simulate a distributed VFL environment consisting of $K = 5$ clients arranged in a fixed tier distribution of $[1 : 2 : 2]$ for edge devices (T_1), gateways (T_2), and security servers (T_3). Feature heterogeneity is introduced through stochastic subset sampling, where clients within the same tier receive distinct feature subsets. Each client maintains local binary labels $y_i^{(k)}$, while only shared model parameters participate in global aggregation.

To ensure reproducibility across different datasets, we standardize the feature allocation percentages while maintaining client-level heterogeneity. For each client k, the local feature set $\mathbf{x}_i^{(k)}$ is constructed via stochastic subset sampling from the hierarchical pools:

1. **Tier-1 Clients**: Receive only the $\mathcal{F}_{\text{core}}$ features.
2. **Tier-2 Clients**: Receive $\mathcal{F}_{\text{core}}$ plus a randomized 60% subset of the $\mathcal{F}_{\text{overlap}}$ pool.
3. **Tier-3 Clients**: Receive $\mathcal{F}_{\text{core}}$ plus a randomized 90% subset of the $\mathcal{F}_{\text{overlap}}$ pool and a randomized 50% subset of the $\mathcal{F}_{\text{unique}}$ pool.

This sampling approach ensures that no two clients within the same tier possess identical feature sets, forcing the model to generalize across non-identical feature distributions. In line with our decentralized label requirement, the binary ground truth label $y_i^{(k)}$ is appended to the local feature vector of every participating client. Overall strategy can be seen in Table 2.

Table 2. Client Tier Assignment and Feature Partitioning Logic in Feature-Heterogeneous VFL

Tier	Ratio	Count	Device Simulation	Feature Assignment	Model
T1 (Low)	20%	1	Resource-Constrained IoT Device	$\mathbf{F}_{Core}$ Only	DNN (Shared)
T2 (Mid)	40%	2	Standard Network Gateway/Device	$\mathbf{F}_{Core}+\approx$ 60% of $\mathbf{F}_{Overlap}$	DNN (Shared)
T3 (High)	40%	2	Enterprise Security Server	$\mathbf{F}_{Core}+\approx$ 90% Overlap+ $\approx$ 50% Unique	DNN (Shared)

6 Experimental Results

To the best of our knowledge, this framework represents the first vertical federated learning (VFL) architecture designed to simultaneously reconcile severe class imbalance ($\leq 1\%$) and structural feature heterogeneity in network security applications. Given the absence of pre-existing frameworks addressing this specific intersection, we establish a customized benchmark denoted as Federated Cross-Entropy (**FedCE**). FedCE implements a conventional VFL training regime leveraging a standard cross-entropy objective. To ensure a structurally equitable comparison, the FedCE baseline incorporates an identical skip-layer configuration to maintain client feature privacy while enabling end-to-end global head optimization. This setup isolates the evaluation metrics to directly contrast prototype-based contrastive representation mapping against traditional supervised paradigms, with full results reported in Table 3

6.1 Intrusion Detection Performance on Network Benchmarks

In extreme data imbalance regimes, traditional performance metrics like global accuracy or precision can become heavily biased toward majority-class distri-

Table 3. Detailed Performance Metrics Across Different Imbalance Scenarios

Dataset	Scenario	Metric	FedCE	SupCon	Minority	(Skip-VFL)
NSL-KDD	Normal	Accuracy	91.88%	94.01%	94.08%	**94.26%**
		W. Precision	92.23%	94.01%	94.08%	**94.26%**
		Anomaly Recall	86.66%	93.79%	93.74%	**93.86%**
		W. F1-Score	91.85%	94.01%	94.08%	**94.26%**
	5% Imbalance	Accuracy	83.94%	95.03%	94.97%	**95.14%**
		W. Precision	95.24%	96.90%	96.89%	**96.99%**
		Anomaly Recall	85.57%	88.29%	88.41%	**89.40%**
		W. F1-Score	88.04%	95.66%	95.62%	**95.76%**
	1% Imbalance	Accuracy	84.18%	**95.32%**	94.82%	95.09%
		W. Precision	98.90%	99.03%	99.03%	**99.04%**
		Anomaly Recall	86.45%	87.10%	**88.39%**	88.39%
		W. F1-Score	90.52%	**96.88%**	96.60%	96.75%
UNSW-NB15	Normal	Accuracy	78.02%	84.88%	**86.71%**	85.76%
		W. Precision	80.78%	84.77%	**86.73%**	85.75%
		Anomaly Recall	74.10%	89.42%	**93.70%**	93.13%
		W. F1-Score	78.44%	84.80%	**86.45%**	85.46%
	5% Imbalance	Accuracy	27.64%	87.38%	**89.54%**	86.58%
		W. Precision	94.83%	94.96%	**95.27%**	95.18%
		Anomaly Recall	**97.65%**	76.79%	78.02%	81.49%
		W. F1-Score	37.28%	90.22%	**91.67%**	89.74%
	1% Imbalance	Accuracy	27.07%	87.35%	92.53%	**99.22%**
		W. Precision	98.90%	98.80%	98.78%	**99.21%**
		Anomaly Recall	**96.79%**	76.47%	70.59%	59.89%
		W. F1-Score	41.34%	92.37%	95.29%	**99.22%**
ToN-IoT	Normal	Accuracy	82.49%	**88.77%**	88.65%	88.74%
		W. Precision	81.61%	**89.88%**	89.83%	89.78%
		Anomaly Recall	91.36%	99.63%	**99.68%**	99.53%
		W. F1-Score	81.86%	**87.50%**	87.34%	87.50%
	5% Imbalance	Accuracy	65.91%	**96.39%**	96.30%	96.36%
		W. Precision	94.65%	**96.67%**	96.59%	96.62%
		Anomaly Recall	**87.83%**	71.67%	70.72%	70.72%
		W. F1-Score	75.42%	**96.51%**	96.43%	96.48%
	1% Imbalance	Accuracy	60.52%	**97.92%**	64.98%	83.94%
		W. Precision	98.96%	**99.07%**	98.97%	98.92%
		Anomaly Recall	**96.04%**	78.22%	96.04%	89.11%
		W. F1-Score	74.40%	**98.38%**	77.79%	90.37%

Note: W. Precision and W. F1-Score refer to weighted metrics to reflect overall classification stability, while Anomaly Recall specifically reports the minority (attack) class detection rate to rigorously .evaluated here across both balanced (Normal) and extreme imbalance (5%, 1%) conditions.

butions, masking catastrophic misclassifications of minority anomaly instances. Consequently, our performance evaluation prioritizes minority attack recall and structural F1-scores as primary indicators of network monitoring reliability. The comprehensive empirical results across the NSL-KDD, UNSW-NB15, and ToN-IoT datasets under various imbalance boundaries ($\gamma \in \{1\%, 5\%, \text{Normal}\}$) are compiled in Table 3.

Performance Under Extreme Class Imbalance (1% and 5%) As the anomaly ratio drops to an extreme $\gamma = 1\%$, standard supervised pipelines like the FedCE baseline degrade sharply, though not always in the way raw recall alone would suggest. On UNSW-NB15, for instance, FedCE's accuracy collapses to just 27.07% and its weighted F1-score to 41.34%, even though its raw anomaly recall remains high at 96.79%: the model over-predicts the anomaly class, buying recall at the cost of precision rather than reliably isolating genuine intrusions. `Skip-VFL` avoids this failure mode on the same split, reaching a peak 99.22% accuracy and 99.22% weighted F1-score, ahead of SupCon (87.35% / 92.37%) and the Minority baseline (92.53% / 95.29%) – though this stability comes with a lower raw recall of 59.89%. On the ToN-IoT 1% split the picture is more mixed: SupCon attains the strongest overall accuracy (97.92%) and F1-score (98.38%), while `Skip-VFL` (83.94% accuracy, 90.37% F1-score) still improves substantially over FedCE (74.40% F1) and the Minority baseline (77.79% F1), and maintains a markedly higher anomaly recall (89.11%) than SupCon's 78.22%. On NSL-KDD at 1% imbalance, SupCon leads on accuracy (95.32%) and F1-score (96.88%), while the Minority baseline and `Skip-VFL` tie for the highest anomaly recall (88.39%); `Skip-VFL` nonetheless secures the top weighted precision (99.04%). Taken together, no single baseline dominates uniformly at $\gamma = 1\%$, but `Skip-VFL` and SupCon both consistently avoid the severe precision or recall collapse exhibited by FedCE, with the strongest individual performer varying by dataset.

This nuanced pattern continues under the $\gamma = 5\%$ imbalance configuration. On NSL-KDD, `Skip-VFL` leads across every metric, achieving 95.14% accuracy, 96.99% weighted precision, 89.40% anomaly recall, and a 95.76% F1-score. On UNSW-NB15, the Minority baseline instead provides the strongest overall profile (89.54% accuracy, 91.67% F1-score), narrowly ahead of `Skip-VFL` (86.58% accuracy, 89.74% F1-score); FedCE remains the weakest performer by a wide margin, its F1-score collapsing to 37.28% despite an inflated recall of 97.65% driven by systematic over-prediction of the anomaly class. On the ToN-IoT 5% partition, SupCon attains the top accuracy (96.39%) and F1-score (96.51%), with `Skip-VFL` close behind (96.36% / 96.48%); notably, FedCE again posts the highest raw anomaly recall (87.83%) here, reinforcing that elevated recall alone is an unreliable indicator of detection quality when accuracy and precision collapse in tandem.

Performance in Balanced (Normal) Scenarios To verify that our imbalance mitigations do not introduce optimization trade-offs in balanced data en-

vironments, we evaluate the frameworks under baseline (Normal) conditions. As shown in Table 3, all contrastive variants remain highly competitive. On NSL-KDD, `Skip-VFL` sweeps every metric, securing the peak accuracy (94.26%), precision (94.26%), anomaly recall (93.86%), and weighted F1-score (94.26%). On UNSW-NB15, the Minority baseline leads narrowly across all four metrics (86.71% accuracy, 86.73% precision, 93.70% recall, 86.45% F1-score), with `Skip-VFL` trailing by less than one point on each (85.76% / 85.75% / 93.13% / 85.46%). In the balanced ToN-IoT configuration, SupCon secures the top accuracy (88.77%) and precision (89.88%), with `Skip-VFL` closely matching (88.74% / 89.78%) and tying SupCon's weighted F1-score at 87.50%; the Minority baseline attains the highest anomaly recall (99.68%), with `Skip-VFL` close behind at 99.53%. These narrow margins indicate that `Skip-VFL`'s imbalance-handling machinery imposes no meaningful accuracy penalty relative to the strongest baselines even when class distributions are already balanced, and that its principal advantage manifests specifically as imbalance intensifies.

7 Conclusion

In this paper, we presented **Skip-VFL**, a tiered prototype-based Vertical Federated Learning framework for robust anomaly detection in heterogeneous and imbalanced network environments. Through evaluations on the NSL-KDD, UNSW-NB15, and ToN-IoT datasets, we demonstrated that Skip-VFL effectively addresses feature heterogeneity and extreme class imbalance through its Skip-Layer architecture and two-stage prototype-guided training strategy. Experimental results show that Skip-VFL achieves robust anomaly detection performance and competitive F1-scores across diverse network settings while preserving data privacy. Future work will focus on extending the framework to multi-class intrusion detection and evaluating its effectiveness in more heterogeneous and large-scale distributed environments.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03038540.

References

1. Bala, R., Nagpal, R.: A review on kdd cup99 and nsl nsl-kdd dataset. International Journal of Advanced Research in Computer Science **10**(2) (2019)
2. Guo, W., Duan, Y., Hu, Z., Tong, Y., Zhuang, F., Zhang, X., Dong, J., Wu, R., Liu, T., Sun, Y.: Proto-evfl: Enhanced vertical federated learning via dual prototype with extremely unaligned data. arXiv preprint arXiv:2507.22488 (2025)
3. He, Y., Kang, Y., Zhao, X., Luo, J., Fan, L., Han, Y., Yang, Q.: A hybrid self-supervised learning framework for vertical federated learning. IEEE Transactions on Big Data (2024)

4. Hu, H., Wang, X., Zhang, Y., Chen, Q., Guan, Q.: A comprehensive survey on contrastive learning. Neurocomputing **610**, 128645 (2024)
5. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
6. Kadhe, S.R., Ludwig, H., Baracaldo, N., King, A., Zhou, Y., Houck, K., Rawat, A., Purcell, M., Holohan, N., Takeuchi, M., et al.: Privacy-preserving federated learning over vertically and horizontally partitioned data for financial anomaly detection. arXiv preprint arXiv:2310.19304 (2023)
7. Kesici, M., Pal, B., Yang, G.: Detection of false data injection attacks in distribution networks: A vertical federated learning approach. IEEE Transactions on Smart Grid **15**(6), 5952–5964 (2024)
8. Khan, A., ten Thij, M., Wilbik, A.: Vertical federated learning: A structured literature review. Knowledge and Information Systems pp. 1–39 (2025)
9. Li, L., Fan, Y., Tse, M., Lin, K.Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020)
10. Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.Q., Yang, Q.: Vertical federated learning: Concepts, advances, and challenges. IEEE transactions on knowledge and data engineering **36**(7), 3615–3634 (2024)
11. Mildenberger, D., Hager, P., Rueckert, D., Menten, M.J.: A tale of two classes: Adapting supervised contrastive learning to binary imbalanced datasets. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10305–10314 (2025)
12. 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)
13. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 military communications and information systems conference (MilCIS). pp. 1–6. IEEE (2015)
14. Mugunthan, V., Goyal, P., Kagal, L.: Multi-vfl: A vertical federated learning system for multiple data and label owners. arXiv preprint arXiv:2106.05468 (2021)
15. Novikova, E., Doynikova, E., Golubev, S.: Federated learning for intrusion detection in the critical infrastructures: Vertically partitioned data use case. Algorithms **15**(4), 104 (2022)
16. Rodda, S., Erothi, U.S.R.: Class imbalance problem in the network intrusion detection systems. In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT). pp. 2685–2688. Ieee (2016)
17. Tsiknas, K., Taketzis, D., Demertzis, K., Skianis, C.: Cyber threats to industrial iot: a survey on attacks and countermeasures. IoT **2**(1), 163–186 (2021)
18. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A practical guide, 1st ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
19. Wang, L., Xu, S., Wang, X., Zhu, Q.: Addressing class imbalance in federated learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 10165–10173 (2021)
20. Xiao, C., Wang, S.: An experimental study of class imbalance in federated learning. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–7. IEEE (2021)
21. Yadav, K., Chaubey, H., Sah, R.: Enhancing iot intrusion detection through vertical federated learning and latent space mapping. In: 2024 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 642–649. IEEE (2024)

22. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Horizontal federated learning. In: Federated learning, pp. 49–67. Springer (2022)
23. Zhao, J., Zhu, H., Zhang, J., Tang, J., Wang, F., Li, H.: Splitad: A lightweight and privacy-enhancing vertical federated anomaly detection framework based on hierarchical autoencoders. Information Sciences **714**, 122211 (2025). https://doi.org/https://doi.org/10.1016/j.ins.2025.122211, https://www.sciencedirect.com/science/article/pii/S0020025525003433

Time-Series Context Representation Learning for Label-Bias-Robust Semi-Supervised Anomaly Detection in ICS

Jung Kyu Seo[0009-0007-7096-383X], Ilhwan Ji[0009-0003-9027-1305] and Jung Taek Seo*[0000-0003-0971-8548]

Gachon University, Seongnam-daero 1342, Seongnam-si, Republic of Korea
{ghfkd0311, ilhwan1013, seojt}@gachon.ac.kr

Abstract. Industrial Control Systems (ICS) monitor and control the physical processes of critical infrastructure, and their exposure to cyberattacks has grown with increasing external connectivity. In such environments, normal operational data are abundant, but attack data and reliable labels are scarce, which makes it difficult to build reliable detection models. When only a small number of labels are available, the detection boundary tends to become biased toward those labels, degrading detection performance. This paper proposes a semi-supervised anomaly detection model based on time-series contextual representation learning to mitigate the detection-boundary bias caused by label scarcity. The model learns contextual features and self-supervised representations from normal operational data and calibrates the detection boundary using a small number of normal and attack labels. On the SWaT, WADI, and HAI datasets with 500 attack labels, it achieved F1-scores of 0.9519, 0.9765, and 0.9113, respectively, reducing label bias relative to existing semi-supervised models and proving effective in ICS environments where labels are limited.

Keywords: ICS, Anomaly Detection, Label Bias, Semi-supervised learning.

1 Introduction

Industrial Control Systems(ICS) control processes in critical infrastructure such as water treatment, power, and manufacturing, and their connectivity with external systems has increased with the recent convergence of Information and Communication Technology(ICT)[1, 2]. Stuxnet went beyond mere information leakage, causing malfunctions in physical processes and equipment damage in ICS environments[3]. To counter such threats, AI-based anomaly detection has been studied. It learns normal data to detect anomalous states that deviate from the learned normal range, and is categorized according to label use into supervised, unsupervised, and semi-supervised learning[4]. However, supervised learning suffers from the difficulty of obtaining sufficient attack labels, while unsupervised learning struggles to distinguish normal variations from anomalous states as operating conditions change. Semi-supervised learning can leverage a small number of labels, but if these labels fail to represent the full range of attack characteristics and operating conditions, the detection boundary becomes biased toward a few labels and cannot sufficiently reflect gradual deviations from normal behavior or

changes in inter-variable relationships. Therefore, ICS with limited labeled data require an anomaly detection method that reflects time-series context while alleviating label-induced boundary bias. To this end, this paper proposes a semi-supervised anomaly detection model based on time-series contextual representation learning for ICS environments. From operational data, the model extracts the change from the previous step and short- and long-term context deviations to construct contextual features and learns inter-variable relationships and temporal structures through self-supervised representation learning based on masked reconstruction and next-step prediction on normal data. It then combines the two feature sets and calibrates the detection boundary using normal training samples with a small number of normal and attack labels, thereby alleviating its bias toward a few attack labels. The model was evaluated on the SWaT (Secure Water Treatment), WADI (Water Distribution), and HAI (HIL-based Augmented ICS) datasets, varying the number of attack labels from 10 to 500 and comparing it with existing supervised, unsupervised, and semi-supervised models[5-7]. In addition, the boundary learner was compared with classifiers XGBoost, Random Forest (RF), MLP, and ResNet to validate the choice of LightGBM. It achieved F1-scores of 0.9519 on SWaT, 0.9765 on WADI, and 0.9113 on HAI under the 500-attack-label condition, showing performance equal to or higher than existing models across most label conditions, with lower performance variance.

The main contributions of this paper are as follows:

- We propose a semi-supervised anomaly detection model based on time-series contextual representation learning for ICS multivariate time-series data.
- We combine self-supervised representation learning from normal operational data with small label-based boundary calibration to alleviate performance degradation caused by label bias.
- We validate the proposed model on the SWaT, WADI, and HAI datasets, confirming its effectiveness against existing supervised, unsupervised, and semi-supervised anomaly detection models.

This paper is organized as follows. Section 2 reviews related works on AI-based anomaly detection for ICS. Section 3 describes the proposed method. Section 4 presents experimental results and comparative analysis. Section 5 concludes the paper and discusses future research directions.

2 Related Works

This section reviews AI-based anomaly detection studies for ICS environments.

2.1 Supervised Learning-Based Anomaly Detection

Supervised learning-based anomaly detection trains a model to distinguish normal from anomaly states using labeled normal and attack data. Mokhtari et al.[8] proposed an intrusion detection method based on SCADA measurement data using the HIL-based HAI Security Dataset; comparing KNN, Random Forest, and others, Random Forest achieved the highest performance, reaching approximately 0.9976 in accuracy, F1-score, and AUC. Zhong et al.[9] proposed STAND to analyze the effect of limited label

information on time-series anomaly detection performance, and quantitatively demonstrated the effect of label utilization on datasets such as SWaT and WADI. These supervised approaches can achieve high detection performance when sufficient labels are available. However, in real ICS environments, it is difficult to intentionally collect diverse attack data. Moreover, detection performance may be limited for new attack types or changes in operating conditions not included in training[10].

2.2 Unsupervised Learning-Based Anomaly Detection

Unsupervised learning-based anomaly detection typically learns normal patterns from normal operational data and computes an anomaly score based on the degree to which input data deviates from these patterns. Inoue et al.[11] compared a DNN-based method combining LSTM and feed-forward layers with a One-Class SVM using the SWaT dataset, demonstrating the applicability of anomaly detection based solely on normal data. Kravchik et al.[12] used a one-dimensional CNN-based model to learn the temporal patterns of SWaT data and detected anomalies based on the difference between predicted and observed values. Unsupervised approaches have the advantage of being applicable in ICS environments where attack labels are scarce. However, their performance is highly sensitive to the threshold setting for anomaly scores, and false alarms may increase when the distribution of normal data shifts due to changes in operating modes. In addition, they may fail to detect attacks concealed to resemble normal behavior and anomalies that develop gradually.

2.3 Semi-Supervised Learning-Based Anomaly Detection

Semi-supervised learning-based anomaly detection detects anomalies by jointly utilizing normal or unlabeled data together with a small amount of labeled data. Ruff et al.[13] proposed DeepSAD, which jointly utilizes a small number of normal and anomaly labels. DeepSAD forms a discriminative boundary between normal and anomaly states by training anomaly samples to move away from the center. Xu et al.[14] proposed RoSAS, which exploits the relative relationship between normal and anomaly states in a small-label setting, and Liu et al.[15] proposed SFSD, which combines feature selection with a deviation network for ICS attack detection. Semi-supervised learning can reflect label information in the detection boundary without requiring large-scale labels, making it suitable for ICS environments by alleviating both the high label dependency of supervised learning and the ambiguous detection boundary of unsupervised learning. However, if the labels fail to adequately represent attack types and operating conditions, the detection boundary may become biased toward specific labels. Furthermore, existing studies do not sufficiently reflect the temporal context of normal operational data and inter-variable relationships, which may limit detection performance for anomalies that develop gradually within the normal range or for new attack types. Therefore, ICS environments require a semi-supervised anomaly detection method that learns temporal context and inter-variable relationships from normal operational data and uses a small number of labels for boundary calibration. Accordingly, this paper proposes a semi-supervised anomaly detection model based on time-series context utilization to alleviate performance degradation caused by label bias.

3 Proposed Semi-Supervised Anomaly Detection Model Based on Time-Series Context Utilization for ICS Environments

This section proposes an anomaly detection model based on time-series contextual representation learning for ICS multivariate time-series data. The model combines self-supervised representation learning from normal operational data with small-label-based semi-supervised boundary learning, alleviating performance degradation caused by label-induced boundary bias or insufficient reflection of time-series context. First, from the raw sensor and actuator values, it extracts the current state, the change from the previous step, and short- and long-term operational context deviations to construct contextual features. It then learns inter-variable relationships and time-series structures through masked reconstruction and next-step prediction on normal data, combines the contextual and self-supervised features, and calibrates the detection boundary using normal training samples with a small number of normal and attack labels. The model comprises data preprocessing, contextual feature extraction, self-supervised representation learning, and semi-supervised boundary learning and anomaly detection. Through this, it performs stable anomaly detection in ICS environments where label acquisition is limited.

3.1 Overview of the Anomaly Detection Model

The proposed model first learns time-series contextual representations from normal operational data, then calibrates the detection boundary using a small number of normal and attack labels. Fig. 1 shows an overview of the proposed model.

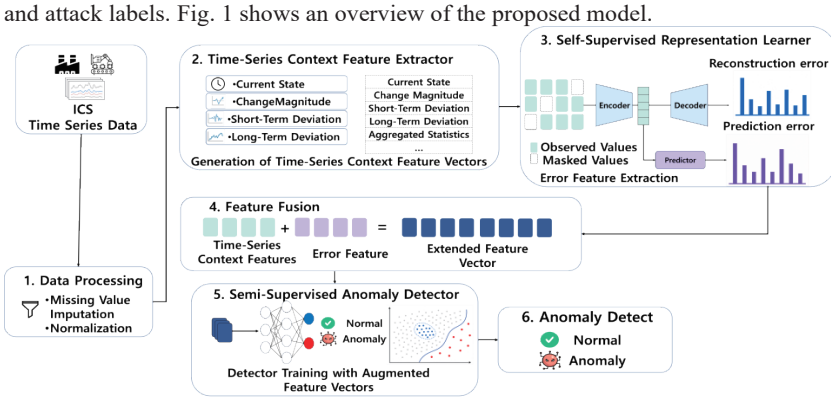


Fig. 1. Overview of the proposed semi-supervised anomaly detection model

As shown in Fig. 1, the model consists of a data preprocessor, a time-series contextual feature extractor, a self-supervised representation learner, and a semi-supervised anomaly detector. The role of each component is as follows.

- **Data preprocessor:** Selects sensor and actuator features from raw ICS data, corrects missing and anomalous values, and standardizes them based on normal operational data.

- **Time-series contextual feature extractor:** Extracts the current state, the change from the previous step, and short- and long-term context deviations to represent instantaneous changes and deviation from normal behavior.
- **Self-supervised representation learner:** Performs masked reconstruction and next-step prediction on normal operational data to learn inter-variable relationships and temporal transition structures, producing latent representations, reconstruction errors, and prediction errors.
- **Semi-supervised anomaly detector:** Learns the detection boundary from features combining the contextual and self-supervised features, computes an anomaly score at each step, and classifies it as normal or anomalous.

Through this, the proposed model reflects the temporal context of normal operational data while alleviating the problem of the detection boundary being biased toward a small number of attack labels.

3.1 Data Preprocessor

The data preprocessor converts ICS multivariate time-series data into input for the anomaly detection model. Feeding raw data directly can cause training errors or bias toward specific variables due to missing values and noise; therefore, preprocessing is performed to ensure the stability of contextual feature extraction and self-supervised representation learning. First, numerical sensor and actuator variables are selected, and unnecessary variables such as time and index are removed. The data are then sorted chronologically, and missing and anomalous values are corrected to maintain continuity. Next, standardization reduces differences in units and value ranges, preventing any single variable from dominating training. Finally, the preprocessed data represent the relative degree of change at each step with respect to the normal operating state.

3.2 Time-Series Contextual Feature Extractor

The time-series contextual feature extractor constructs, from the preprocessed data, the temporal change information needed to distinguish normal from anomalous states. Anomalies in ICS take various forms changes in sensor–actuator relationships, mismatches between control commands and process responses, and gradual deviations from normal behavior so the current value alone cannot capture slowly developing anomalies. The proposed features therefore jointly reflect the current state, the change from the previous step (instantaneous anomaly signs), and short- and long-term context deviations (deviation from recent behavior and accumulated deviation from the normal trend). In addition, aggregate statistical features summarizing the mean and maximum of these features capture anomaly signs appearing simultaneously across multiple variables, complementing the limitations of static, single-step features.

3.3 Self-Supervised Representation Learner

The self-supervised representation learner learns the inter-variable relationships and temporal transition structures inherent in ICS time-series data using only normal operational data, since attack data and reliable labels are difficult to secure in real environments. It consists of an encoder, a decoder, and a next step predictor, taking the

contextual features as input: the encoder transforms them into a latent representation, the decoder reconstructs them, and the predictor estimates the subsequent operating state. Training jointly performs masked reconstruction and next step prediction. In the normal state, inter variable relationships and temporal structures remain consistent, so reconstruction and prediction errors are small; in the anomalous state, these relationships are distorted, increasing the errors. Once trained, the learner generates a latent representation, reconstruction error, and prediction error for each time step, indicating the degree of deviation from the normal operational structure.

3.4 Semi-Supervised Boundary Learning

Semi-supervised boundary learning combines the time-series contextual and self-supervised features to learn a detection boundary between normal and anomalous states. The final detection features form a single vector combining the contextual features and aggregate statistics with the latent representation, reconstruction error, and prediction error, jointly capturing the current operating state, temporal changes, and deviation from the normal structure. A binary classifier learns the boundary from normal training samples and a small number of normal and attack labels: the normal samples provide the distribution of the normal operating state, while even limited attack labels convey the direction in which anomalous states deviate from normal behavior. This yields a clearer boundary than unsupervised methods while reducing the boundary's bias toward a few attack labels.

3.5 Anomaly Detector

The anomaly detector determines whether each step is anomalous using the learned semi-supervised boundary model. The anomaly score is defined as the likelihood that the final detection features at a given step belong to an anomalous state. The LightGBM based model takes these features as input and computes the score; a higher score indicates the step is closer to a cyberattack or anomalous state. The threshold is a normal anomalous criterion set from the small amount of labeled data available during training: a step is classified as anomalous if its score is at or above the threshold, and normal otherwise. In this way, the model computes anomaly scores from the time-series context of normal operational data and uses labeled data for threshold calibration, alleviating performance degradation caused by label bias.

4 Experimentation and Validation

4.1 Experimental Environment

All experiments were conducted on an AMD Ryzen 7 5800X at 3.80 GHz, 64.0 GB RAM, Windows 11, and an NVIDIA GeForce RTX 3070. In preprocessing, the standardization parameters were learned from normal operational data and applied identically to the training and test data, and the short- and long-term windows for contextual feature extraction were set to 30 and 120. The self-supervised representation learner an encoder, a decoder, and a next-step predictor used a masking ratio of 0.15, a hidden dimension of 128, a latent dimension of 16, a learning rate of 0.001, and a batch size of 1,024. For

boundary learning, a LightGBM-based binary classifier took the final detection features as input, configured with 900 trees, a learning rate of 0.025, 127 leaves, and class weights for class imbalance. The number of normal training samples was 10,000, and the number of attack labels varied from 10 to 50 in increments of 10 and from 50 to 500 in increments of 50. The threshold was set to the value yielding the highest F1-score on the training-time labeled data, without using test labels. Performance was evaluated using accuracy, precision, recall, F1-score, and AUROC, with F1-score as the primary metric. For the boundary-learner comparison, XGBoost, Random Forest (RF), MLP, and ResNet were evaluated on the same detection features, training samples, and test split: XGBoost with 900 estimators, a learning rate of 0.025, a maximum depth of 6, subsampling and column sampling ratios of 0.90, and a class-imbalance weight; RF with 500 trees and balanced subsampling-based class weights; the MLP with hidden dimensions of 256 and 128, batch normalization, and dropout of 0.15; and the ResNet with three residual blocks and a hidden dimension of 256 both neural classifiers trained with AdamW at a learning rate of 0.001, weight decay of 0.0001, a batch size of 4,096, and balanced sampling.

4.2 Datasets

To validate the proposed model, we used SWaT, WADI, and HAI benchmark datasets(D) widely used in the field of ICS anomaly detection[5-7]. These are multivariate time-series datasets in which sensor and actuator values are recorded chronologically, with normal and attack state labels. Self-supervised representation learning was performed on the normal operational data (Train), while semi-supervised boundary learning and threshold setting were carried out on the Validation data, consisting of 10,000 normal samples and 10 to 500 attack labels. The Test data were kept identical across all label conditions to evaluate performance as the number of labels varied.

- **SWaT:** Collected from an ICS testbed simulating a water treatment process. The Train data consists of 495,000 normal samples; the Test data consists of 264,762 time steps (237,179 normal, 27,583 attack).
- **WADI:** Collected from an ICS testbed targeting a water distribution system. The Train data consists of 784,571 normal samples; the Test data consists of 102,734 time steps (97,696 normal, 5,038 attack).
- **HAI:** Collected from a HIL-based ICS testbed, including interactions among complex processes and diverse sensor and actuator variables (version 23.05). The Train data consists of 896,400 normal samples; the Test data consists of 169,559 time steps (163,810 normal, 5,749 attack).

4.3 Performance Evaluation

This section evaluates the anomaly detection performance of the proposed model as the number of attack labels varies. As shown in Table 1, performance was measured on the same test set while varying the number of labels(C) from 10 to 500, verifying whether the model detects anomalous states under limited attack-label conditions. Performance was evaluated using Accuracy(A), precision(P), recall(R), F1-score(F1), and AUROC, with the anomalous state as the positive class. Accuracy is the proportion of data correctly classified as normal or anomalous, precision is the proportion of truly anomalous

data among those predicted as anomalous, recall is the proportion of truly anomalous data correctly detected, and F1-score is the harmonic mean of precision and recall.

Table 1. Performance of the proposed model(SWaT:S, WADI: W, HAI: H)

D	C	10	20	30	40	50	100	150	200	250	300	350	400	450	500
S	A	0.9568	0.9622	0.9627	0.9628	0.9681	0.9714	0.9737	0.9794	0.9820	0.9847	0.9875	0.9889	0.9894	0.9903
	P	0.9975	0.9960	0.9906	0.9940	0.9926	0.9905	0.9891	0.9876	0.9862	0.9879	0.9853	0.9848	0.9824	0.9817
	R	0.5873	0.6399	0.6486	0.6469	0.6988	0.7330	0.7555	0.8129	0.8386	0.8640	0.8932	0.9075	0.9148	0.9239
	F1	0.7393	0.7792	0.7839	0.7837	0.8202	0.8425	0.8567	0.8918	0.9064	0.9218	0.9370	0.9446	0.9474	0.9519
	AUROC	0.8486	0.8645	0.8755	0.9166	0.9221	0.9408	0.9678	0.9792	0.9870	0.9920	0.9937	0.9944	0.9967	0.9975
W	A	0.9546	0.9633	0.9665	0.9737	0.9822	0.9933	0.9961	0.9964	0.9970	0.9972	0.9974	0.9972	0.9973	0.9977
	P	0.9748	0.9832	0.9878	0.9863	0.9858	0.9816	0.9777	0.9804	0.9741	0.9707	0.9669	0.9615	0.9628	0.9657
	R	0.0768	0.2553	0.3206	0.4704	0.6457	0.8803	0.9416	0.9446	0.9647	0.9730	0.9807	0.9821	0.9819	0.9877
	F1	0.1424	0.4053	0.4840	0.6370	0.7803	0.9282	0.9594	0.9622	0.9694	0.9718	0.9738	0.9717	0.9723	0.9765
	AUROC	0.8660	0.9220	0.9663	0.9805	0.9935	0.9975	0.9983	0.9985	0.9993	0.9994	0.9997	0.9997	0.9996	0.9996
H	A	0.9683	0.9711	0.9718	0.9732	0.9743	0.9793	0.9830	0.9863	0.9885	0.9905	0.992	0.9931	0.9936	0.9939
	P	0.9584	0.9709	0.9477	0.9697	0.9742	0.9628	0.9573	0.9541	0.9311	0.9257	0.9275	0.9185	0.9035	0.9058
	R	0.0682	0.1508	0.1795	0.2169	0.2494	0.4053	0.5222	0.6262	0.7149	0.7841	0.8297	0.8748	0.9071	0.9169
	F1	0.1273	0.2611	0.3018	0.3545	0.3972	0.5704	0.6757	0.7561	0.8088	0.8490	0.8759	0.8961	0.9053	0.9113
	AUROC	0.7166	0.7838	0.8063	0.8445	0.8572	0.9351	0.9582	0.9810	0.9834	0.9888	0.9909	0.9916	0.9948	0.9949

Table 1 shows that the F1-score of the proposed model consistently improved as the number of attack labels increased from 10 to 500 across all datasets. SWaT maintained relatively high performance even in the extremely low label regime, recording 0.7393 with only 10 labels and reaching 0.9519 at 500 labels. WADI and HAI showed low recall with fewer than 50 labels but improved rapidly thereafter, reaching 0.9282 and 0.5704 at 100 labels and 0.9765 and 0.9113 at 500 labels, respectively, indicating that a certain minimum number of labels is required to stabilize the detection boundary in complex process environments. Precision remained above 0.9 in most conditions, and AUROC steadily increased with labels, exceeding 0.99 on all datasets at 500 labels. By learning the time-series context of normal operational data and calibrating the detection boundary with a few labels, the proposed model achieved effective anomaly detection even under label-limited conditions.

Table 2. F1-score comparison according to classifiers

D	M	10	20	30	40	50	100	150	200	250	300	350	400	450	500
S	LightGBM	0.7393	0.7792	0.7839	0.7837	0.8202	0.8425	0.8567	**0.8918**	0.9064	**0.9218**	**0.9370**	**0.9446**	**0.9474**	**0.9519**
	XGBoost	0.7429	0.7523	0.8042	0.8252	**0.8377**	**0.8572**	**0.8711**	0.8916	**0.9086**	0.9208	0.9320	0.9353	0.9376	0.9418
	RF	**0.7718**	0.7887	**0.8053**	0.8197	0.8303	0.8463	0.8648	0.8780	0.8926	0.9172	0.9155	0.9223	0.9261	0.9299
	ResNet	0.7517	0.7949	0.7978	0.8155	0.8137	0.8459	0.8558	0.8660	0.8676	0.8828	0.8838	0.8763	0.8760	0.8788
	MLP	0.7518	**0.7970**	0.8025	**0.8303**	0.8293	0.8534	0.8610	0.8750	0.8711	0.8822	0.8715	0.8851	0.8895	0.8791
W	LightGBM	0.1424	0.4053	0.4840	0.6370	0.7803	0.9282	**0.9594**	0.9622	0.9694	0.9718	**0.9738**	0.9717	0.9723	**0.9765**
	XGBoost	0.2684	0.6310	0.6530	0.7675	**0.8610**	**0.9469**	0.9565	**0.9634**	**0.9727**	**0.9725**	0.9731	**0.9763**	**0.9740**	0.9684
	RF	0.2475	**0.6373**	**0.6679**	**0.8400**	0.8384	0.9216	0.9397	0.9430	0.9569	0.9676	0.9701	0.9739	0.9750	0.9753
	ResNet	0.4253	0.5674	0.4264	0.5399	0.5130	0.7138	0.7944	0.8634	0.8911	0.9014	0.9188	0.9073	0.9292	0.9229
	MLP	**0.4498**	0.5925	0.5430	0.5772	0.5250	0.7815	0.8580	0.8899	0.8936	0.8677	0.9117	0.9050	0.9265	0.9233
H	LightGBM	0.1273	0.2611	0.3018	0.3545	0.3972	0.5704	0.6757	0.7561	0.8088	**0.8490**	**0.8759**	**0.8961**	**0.9053**	**0.9113**
	XGBoost	0.2107	0.3038	0.3582	**0.4443**	**0.4824**	**0.6390**	**0.7426**	**0.7986**	**0.8229**	0.8453	0.8608	0.8737	0.8854	0.8891
	RF	**0.2944**	**0.3290**	**0.3692**	0.2995	0.4213	0.5001	0.5833	0.6252	0.6714	0.7299	0.7578	0.7903	0.7927	0.8168
	ResNet	0.1867	0.2097	0.3010	0.3112	0.3840	0.5914	0.6425	0.6705	0.7375	0.7718	0.7768	0.7953	0.7671	0.7495
	MLP	0.1880	0.2190	0.2637	0.2698	0.3451	0.5891	0.6803	0.6309	0.7504	0.7646	0.7810	0.8054	0.8062	0.7503

Table 2 compares the F1-scores obtained when the boundary learning stage is implemented with LightGBM, XGBoost, Random Forest (RF), ResNet, and MLP under

identical conditions. LightGBM achieved the highest F1-scores under most label conditions, recording 0.9519, 0.9765, and 0.9113 on SWaT, WADI, and HAI at 500 labels, and outperformed all alternatives across the entire label range on HAI. The neural classifiers were less consistent under label limited conditions, as the labeled samples available for boundary learning fall far short of what deep classifiers typically require consistent with the proposed design of employing the deep network for self-supervised representation learning on abundant unlabeled data while performing boundary learning with a lightweight model. Consequently, LightGBM provided the most stable detection performance, with low training cost and native support for class weighting under class imbalance.

4.4 Comparative Study and Label-Bias Sensitivity Analysis

This section validates the detection performance of the proposed model by comparing it with existing semi-supervised anomaly detection models(M) DeepSAD[13], RoSAS[14], SFSD[15] as well as a supervised model, STAND[9], and an unsupervised model, TranAD[16]. For a fair comparison, all baseline models were directly re-implemented and evaluated in the same experimental environment as the proposed model, using identical preprocessing, data splits, and evaluation protocols, rather than citing results reported in prior studies. Table 3 compares the F1-score of the proposed model with those of the existing supervised, unsupervised, and semi-supervised anomaly detection models. The comparison was conducted under conditions of 50 or more labels, where all baseline models could be trained stably.

Table 3. Performance comparison F1-score

D	M	50	100	150	200	250	300	350	400	450	500
S	**DeepSAD[13]**	**0.8356**	**0.8515**	0.8566	0.8822	0.9011	0.8822	0.9016	0.8996	0.8734	0.9179
	RoSAS [14]	0.8069	0.845	0.8277	0.8536	0.8555	0.8629	0.8626	0.8752	0.8787	0.8890
	SFSD [15]	0.8133	0.8482	0.8548	0.8673	0.8640	0.878	0.8865	0.8905	0.8914	0.8862
	TranAD[16]	0.8143									
	STAND[9]	0.7122									
	Ours	0.8202	0.8425	**0.8567**	**0.8918**	**0.9064**	**0.9218**	**0.9370**	**0.9446**	**0.9474**	**0.9519**
W	**DeepSAD**	0.7535	0.7527	0.8720	0.9466	0.9508	0.9572	0.9556	0.9554	0.7460	0.9412
	RoSAS	0.5773	0.7540	0.8251	0.9152	0.9067	0.9224	0.9278	0.9549	0.9440	0.9390
	SFSD	0.5004	0.7815	0.8602	0.8994	0.9019	0.8952	0.9315	0.9071	0.9404	0.9385
	TranAD	0.4360									
	STAND	0.8305									
	Ours	**0.7803**	**0.9282**	**0.9594**	**0.9622**	**0.9694**	**0.9718**	**0.9738**	**0.9717**	**0.9723**	**0.9765**
H	**DeepSAD**	**0.5095**	0.5431	**0.6966**	0.7143	0.7465	0.7825	0.7875	0.8305	0.7759	0.8227
	RoSAS	0.2121	0.4166	0.5285	0.6239	0.6320	0.6595	0.6411	0.7001	0.6687	0.7078
	SFSD	0.3817	0.5632	0.6515	0.5066	0.6998	0.7470	0.6536	0.7464	0.6998	0.7723
	TranAD	0.5278									
	STAND	0.5967									
	Ours	0.3972	**0.5704**	0.6757	**0.7561**	**0.8088**	**0.8490**	**0.8759**	**0.8961**	**0.9053**	**0.9113**

Table 3 shows that the F1-score of the proposed model improved as the number of attack labels increased across all three datasets. SWaT recorded 0.8202 with 50 labels and 0.9519 with 500, WADI achieved 0.9282 with 100 labels and 0.9765 with 500, and HAI improved from 0.3972 to 0.9113. This indicates that, by learning the time-series context of normal operational data and calibrating the detection boundary with few labels, the proposed model achieves effective anomaly detection even under label-limited conditions. Table 4 compares the label-bias sensitivity of the proposed and existing semi-supervised models from 50 to 500 attack labels.

Table 4. Label-Bias Sensitivity Comparison results

D	M	50	100	150	200	250	300	350	400	450	500
S	DeepSAD	0.0085	0.0158	0.0247	0.0262	0.0100	0.0169	0.0131	0.0139	0.0172	0.0120
	RoSAS	**0.0068**	0.0196	0.0169	0.0202	**0.0084**	0.0148	0.0139	0.0152	0.015	0.0104
	SFSD	0.0127	**0.0083**	0.0128	**0.0124**	0.0174	**0.0048**	**0.0061**	**0.0039**	**0.0024**	0.0112
	Ours	0.0084	0.0105	**0.0112**	0.0173	0.0103	0.0114	0.0085	0.0077	0.0076	**0.0053**
W	DeepSAD	0.0903	0.0349	0.0527	0.0558	0.0324	0.0291	0.0227	0.0229	0.0118	0.0123
	RoSAS	0.0735	0.0394	0.0478	0.0205	0.0171	0.0167	0.0154	0.0174	0.0087	0.0078
	SFSD	0.0725	0.0365	0.0264	0.0604	0.0463	0.0111	0.039	0.0223	0.0228	0.0120
	Ours	**0.0280**	**0.0124**	**0.0081**	**0.0048**	**0.0055**	**0.0042**	**0.0059**	**0.0064**	**0.0054**	**0.0050**
H	DeepSAD	0.0531	0.0489	0.0252	0.0218	0.0879	0.0361	0.0362	0.0339	0.0750	0.0175
	RoSAS	0.0535	0.0257	0.0156	0.0304	0.0311	0.0285	0.0432	0.0227	0.0395	0.0407
	SFSD	0.0580	0.0554	0.0198	0.0765	0.0279	0.0224	0.0348	0.0369	0.0356	0.0192
	Ours	**0.0401**	**0.0230**	**0.0155**	**0.0126**	**0.0173**	**0.0105**	**0.0133**	**0.0105**	**0.0118**	**0.0111**

Table 4 shows the label-bias sensitivity comparison, where sensitivity is defined as the average of the seed-wise standard deviations of F1, recall, and precision across five label seeds. The proposed model showed lower sensitivity than existing models under all conditions on WADI and HAI, and stable results under most conditions on SWaT. This relates to dataset complexity. SWaT's simple process structure allows a stable boundary with only a few labels, whereas on WADI and HAI, with many variables and complex structures, a few labels cannot represent the full range of attack characteristics, so label-dependent models vary substantially with the label combination. By learning contextual representations from normal operational data and using labels only for boundary calibration, the proposed model is more robust against label bias as dataset complexity grows.

4.5 Ablation Study

This section conducts an ablation study on the SWaT, WADI, and HAI datasets to examine changes in detection performance according to feature composition.

Table 5. Performance comparison according to stage-wise feature composition (1)

D	M		10	20	30	40	50	100	150	200	250	300	350	400	450	500
S	Raw	F1	0.7480	0.7487	0.7507	0.7517	0.7529	0.7146	0.6926	0.7584	0.7444	0.7614	0.7655	0.7412	0.7674	0.7694
		R	0.6184	0.6184	0.6184	0.6187	0.6181	0.5601	0.5314	0.6318	0.6022	0.6457	0.6442	0.6003	0.6523	0.6401
		P	0.9462	0.9485	0.9551	0.9576	0.9628	0.9868	0.9943	0.9486	0.9744	0.9277	0.9431	0.9686	0.9320	0.9640
	Raw +SSL	F1	0.7498	0.7649	0.7844	0.8192	0.8202	0.8464	0.8635	0.8791	0.9028	0.9109	0.9167	0.9205	0.9248	0.9259
		R	0.6025	0.6228	0.6492	0.6987	0.7000	0.7411	0.7704	0.7962	0.8416	0.8556	0.8667	0.8774	0.8828	0.8872
		P	0.9924	0.9909	0.9907	0.9898	0.9901	0.9865	0.9822	0.9813	0.9736	0.9738	0.9727	0.9680	0.9709	0.9682
	Context	F1	0.7381	0.7573	0.7866	0.7687	0.8141	0.8324	0.8536	0.8917	0.9088	0.9199	0.9315	0.9413	0.9447	0.9496
		R	0.5853	0.6112	0.6514	0.6862	0.6897	0.7175	0.7496	0.8111	0.8415	0.8603	0.8824	0.9004	0.9073	0.9163
		P	0.9991	0.9954	0.9928	0.8737	0.9933	0.9913	0.9912	0.9900	0.9877	0.9883	0.9865	0.9862	0.9854	0.9856
	Context +SSL	F1	0.7393	0.7793	0.7849	0.7834	0.8202	0.8425	0.8567	0.8918	0.9064	0.9218	0.9370	0.9446	0.9474	0.9519
		R	0.5874	0.6401	0.6500	0.6464	0.6988	0.7330	0.7555	0.8129	0.8386	0.8640	0.8932	0.9075	0.9148	0.9239
		P	0.9974	0.9957	0.9906	0.9940	0.9926	0.9905	0.9891	0.9876	0.9862	0.9879	0.9853	0.9848	0.9824	0.9817
W	Raw	F1	0.0071	0.1000	0.1013	0.1209	0.1242	0.1799	0.3318	0.3434	0.3766	0.3575	0.3911	0.4443	0.4381	0.4703
		R	0.0036	0.0528	0.0536	0.0655	0.0675	0.1008	0.2156	0.2253	0.2660	0.2410	0.2805	0.3541	0.3412	0.3847
		P	1.0000	0.9399	0.9184	0.7857	0.7745	0.8328	0.7202	0.7220	0.6449	0.6925	0.6461	0.5963	0.6120	0.6051
	Raw +SSL	F1	0.2837	0.5508	0.6229	0.6959	0.6948	0.8748	0.9076	0.9126	0.9320	0.9322	0.9471	0.9433	0.9434	0.9450
		R	0.1655	0.3821	0.4579	0.5403	0.5401	0.7936	0.8519	0.8684	0.9025	0.9150	0.9418	0.9454	0.9543	0.9599
		P	0.9905	0.9862	0.9738	0.9774	0.9739	0.9746	0.9710	0.9615	0.9633	0.9499	0.9524	0.9411	0.9327	0.9305
	Context	F1	0.1433	0.2910	0.4101	0.6346	0.7607	0.9301	0.9595	0.9627	0.9717	0.9701	0.9738	0.9734	0.9720	0.9762
		R	0.0772	0.1707	0.2590	0.4680	0.6195	0.8805	0.9405	0.9468	0.9661	0.9730	0.9815	0.9819	0.9835	0.9891
		P	0.9949	0.9851	0.9834	0.9854	0.9852	0.9856	0.9793	0.9791	0.9775	0.9672	0.9662	0.9651	0.9608	0.9636
	Context +SSL	F1	0.1424	0.4053	0.4840	0.6370	0.7803	0.9282	0.9594	0.9622	0.9694	0.9718	0.9738	0.9717	0.9723	0.9765
		R	0.0810	0.2483	0.3257	0.4708	0.6457	0.8803	0.9416	0.9446	0.9647	0.9730	0.9807	0.9821	0.9819	0.9877
		P	0.9761	0.9866	0.9897	0.9863	0.9858	0.9816	0.9777	0.9804	0.9741	0.9707	0.9669	0.9615	0.9628	0.9657

Table 6. Performance comparison according to stage-wise feature composition (2)

D	M		10	20	30	40	50	100	150	200	250	300	350	400	450	500
H	Raw	F1	0.3036	0.2997	0.3136	0.2916	0.3270	0.3496	0.3409	0.3192	0.3035	0.3436	0.3752	0.3649	0.3730	0.3824
		R	0.1946	0.1945	0.2046	0.1854	0.2226	0.2484	0.2425	0.2282	0.2105	0.2593	0.2947	0.3021	0.3070	0.3289
		P	0.6899	0.6534	0.6716	0.6825	0.6154	0.5898	0.5739	0.5307	0.5441	0.5089	0.5163	0.4606	0.4751	0.4565
	Raw +SSL	F1	0.1491	0.1988	0.2494	0.3189	0.4182	0.5534	0.6379	0.6829	0.7301	0.7668	0.7918	0.8047	0.8143	0.8252
		R	0.0807	0.1108	0.1442	0.1924	0.2668	0.3952	0.4910	0.5498	0.6140	0.6740	0.7144	0.7471	0.7805	0.8005
		P	0.9727	0.9666	0.9221	0.9310	0.9660	0.9228	0.9101	0.9008	0.9003	0.8892	0.8880	0.8719	0.8511	0.8516
	Context	F1	0.1476	0.2576	0.2759	0.3312	0.3779	0.5374	0.6650	0.7432	0.7970	0.8271	0.8559	0.8887	0.9044	0.9088
		R	0.0798	0.1487	0.1609	0.1995	0.2343	0.3719	0.5084	0.6112	0.6999	0.7502	0.7979	0.8593	0.9015	0.9148
		P	0.9766	0.9618	0.9666	0.9737	0.9761	0.9683	0.9609	0.9479	0.9253	0.9216	0.9229	0.9203	0.9072	0.9030
	Context +SSL	F1	0.1273	0.2611	0.3018	0.3545	0.3972	0.5704	0.6757	0.7561	0.8088	0.8490	0.8759	0.8961	0.9053	0.9113
		R	0.0666	0.1513	0.1816	0.2124	0.2494	0.4053	0.5222	0.6262	0.7149	0.7841	0.8297	0.8748	0.9071	0.9169
		P	0.9623	0.9721	0.9431	0.9660	0.9742	0.9628	0.9573	0.9541	0.9311	0.9257	0.9275	0.9185	0.9035	0.9058

Tables 5 and Table 6 show that the raw-feature-based model produced low F1-scores and recall across all datasets, while adding self-supervised representation features improved performance by complementing the normal structural information that raw features lack. In the extremely low-label regime (10 to 50 labels), the contextual-feature-based compositions recovered more rapidly, exceeding an F1-score of 0.6 on WADI with only 40 labels, whereas the raw-feature-based model remained below 0.13. At 500 labels, the contextual feature-based model recorded 0.9496, 0.9762, and 0.9088 on SWaT, WADI, and HAI, and the proposed model combining both feature types outperformed it from 50 labels onward, reaching 0.9519, 0.9765, and 0.9113. Thus, the contextual features had the greatest impact, while the self-supervised features played a supplementary role in reducing false alarms.

5 Conclusion and Future Works

We proposed an anomaly detection model based on time-series contextual representation learning for ICS environments. The model constructs contextual features from operational data, derives latent representations, reconstruction errors, and prediction errors through self-supervised learning on normal data, and combines them with a small number of normal and attack labels to calibrate the detection boundary. Evaluated on SWaT, WADI, and HAI with 10 to 500 attack labels, the model achieved F1-scores of 0.9519, 0.9765, and 0.9113 at 500 labels, showing equal or higher performance than existing supervised, unsupervised, and semi-supervised models under most label conditions, and LightGBM provided the most stable performance among the compared boundary learners, including XGBoost, Random Forest (RF), MLP, and ResNet. This confirms that learning the time-series context of normal data and calibrating the boundary with a few labels is effective in label-limited ICS environments. In future works, we aim to develop boundary calibration techniques robust to label noise for real industrial settings.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00241376, Development of security monitoring technology-based network behavior against encrypted cyber threats in maritime environment, 50%) and Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT)(RS-2026-25585348, Graduate School of Security Convergence(AI)(Gachon University), 50%).

References

1. Knowles, W., et al., *A survey of cyber security management in industrial* control systems. International journal of critical infrastructure protection, 2015. 9: p. 52–80.
2. Drias, Z., A. Serhrouchni, and O. Vogel. Analysis of cyber security for industrial control systems. in 2015 international conference on cyber security of smart cities, industrial control system and communications (ssic). 2015. IEEE.
3. Langner, R., Stuxnet: Dissecting a cyberwarfare weapon. IEEE security & privacy, 2011. 9(3): p. 49–51.
4. Umer, M.A., et al., Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. International Journal of Critical Infrastructure Protection, 2022. 38: p. 100516.
5. Mathur, A.P. and N.O. Tippenhauer. SWaT: A water treatment testbed for research and training on ICS security. in 2016 international workshop on cyber-physical systems for smart water networks (CySWater). 2016. IEEE.
6. Ahmed, C.M., V.R. Palleti, and A.P. Mathur. WADI: a water distribution testbed for research in the design of secure cyber physical systems. in Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks. 2017.
7. Shin, H.-K., et al. HAI 1.0:HIL-based augmented ICS security dataset. in 13Th USENIX workshop on cyber security experimentation and test (CSET 20). 2020.
8. Mokhtari, S., et al., A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics, 2021. 10(4): p. 407.
9. Zhong, Z., et al., Labels Matter More Than Models: Quantifying the Benefit of Supervised Time Series Anomaly Detection. arXiv preprint arXiv:2511.16145, 2025.
10. Fung, C., et al. Perspectives from a comprehensive evaluation of reconstruction-based anomaly detection in industrial control systems. in European Symposium on Research in Computer Security. 2022. Springer.
11. Inoue, J., et al. Anomaly detection for a water treatment system using unsupervised machine learning. in 2017 IEEE international conference on data mining workshops (ICDMW). 2017. IEEE.
12. Kravchik, M. and A. Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. in Proceedings of the 2018 workshop on cyber-physical systems security and privacy. 2018.
13. Ruff, L., et al., Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694, 2019.
14. Xu, H., et al., RoSAS: Deep semi-supervised anomaly detection with contamination-resilient continuous supervision. Information Processing & Management, 2023. 60(5): p. 103459.
15. Liu, Y., et al., Semi-supervised attack detection in industrial control systems with deviation networks and feature selection: Y. Liu et al. The Journal of Supercomputing, 2024. 80(10): p. 14600–14621.
16. Tuli, S., G. Casale, and N.R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284, 2022.

Who Gains More from AI? A Stackelberg Game Analysis of Cyber Attack–Defense Race

SeHee Jung[1 [0000−0003−3087−8159]] and Jonghu Lee[1 [0009−0007−2287−0321]]

NSR

Abstract. AI is reshaping the strategic choices of both attackers and defenders in the cybersecurity landscape. Attackers can leverage AI to automate the attack life-cycle, while defenders can strengthen threat detection, Cyber Threat Intelligence (CTI), and proactive response capabilities. Accordingly, many states and organizations have elevated Active Cyber Defense (ACD) as a core cybersecurity policy instrument. Yet systematic analysis of which side gains more from AI, and which ACD policy instruments matter most remains limited.

This paper models the AI-enabled attack–defense competition as a three-stage Stackelberg game, in which ACD policy serves as the defender's primary strategic instrument and analyzes how AI cost asymmetry, CTI capability, and ex-ante response investment affect equilibrium attack intensity and defender outcomes. We complement the theoretical analysis with an empirical illustration using 1,569 exploited vulnerabilities observed between 2021 and 2026.

The results show that AI's effect is conditional on the relative cost reductions each side achieves: under attacker-advantage conditions, greater AI intensifies attacks, while under defender-advantage conditions, it deters them. Post-AI, the attack–defense race has compressed substantially, with attack timelines shortening faster than defense timelines. CTI-based risk assessment better explains actual exploitation activity and defense prioritization than vulnerability severity scores (CVSS) alone, suggesting that CTI quality and timeliness are central to effective ACD. These findings imply that effective ACD policy should prioritize CTI-based risk assessment and proactive investment over broad defensive automation.

Keywords: AI Attack-Defense Race · Active Cyber Defense · Game Theory · Stackelberg Game · Cyber Threat Intelligence

1 Introduction

The dual-use nature of AI creates a fundamental tension in cybersecurity: the same capabilities that strengthen threat detection and automated defense can equally accelerate cyberattacks and lower the barrier to sophisticated exploitation [10]. Yet determining which side gains more remains an open and empirically underexplored question [13].

Active Cyber Defense (ACD) has emerged as a cornerstone of national cybersecurity strategies, emphasizing proactive threat identification over passive post-breach response. Yet which ACD instruments matter most in a rapidly evolving

AI environment remains insufficiently understood, and CTI is attracting attention as one promising lever for prioritizing genuine threats under constrained resources.

This paper addresses that question through two complementary analyses. First, we construct a three-stage Stackelberg game to characterize how AI cost asymmetry and ACD policy affect the strategic balance. Second, we conduct an empirical illustration using an integrated dataset from CISA KEV, NVD, Mandiant, and FIRST EPSS (1,569 CVEs, 2021–2026). Drawing on both analyses, we derive implications for which ACD policy instruments defenders should prioritize in an AI-enabled environment, and why strengthening CTI capability may be central to that effort.

2 Related Work

2.1 AI and Cybersecurity

AI enhances both offensive and defensive capabilities across the cybersecurity landscape [10, 19]. On the offensive side, AI augments every stage of the Cyber Kill Chain, enabling attack campaigns that previously required weeks or months to be completed in hours or days—including autonomous vulnerability discovery and exploit development without expert assistance [1, 20]. On the defensive side, AI contributes to prevention, detection, response, and recovery by processing large volumes of data to identify anomalies and accelerate incident response [21].

This parallel adoption resembles an arms race [18]. The prevailing view holds that the structural asymmetry of cybersecurity—attackers need only one success while defenders must address every threat—continues to favor attackers even with AI [5], though this assessment relies primarily on expert surveys [13].

2.2 Active Cyber Defense

Active Cyber Defense(ACD) has become a cornerstone of national cybersecurity strategies in Japan [6], the UK [11], Australia [4], Korea [17], and the US [22]. Definitions vary: NIST describes ACD as synchronized real-time capability to discover, detect, analyze, and mitigate threats [16]; DARPA emphasizes direct engagement with sophisticated adversaries in cyberspace [7]; and the UK NCSC frames it as large-scale protection against commodity attacks [15]. This paper treats ACD as the capability to *detect, analyze, and mitigate threats before an attack materializes*, distinct from offensive cyber operations.

This paper identifies three instruments through which ACD policy operates: CTI (g_{TI}), which governs the quality of threat anticipation; information sharing (g_{SH}), which determines how rapidly intelligence propagates across organizations; and operational capability (g_{OP}), which translates detection into deterrence. Together, these three instruments capture the know–share–act mechanism through which defenders reduce attack incentives before an incident occurs. These form the ACD policy vector $\boldsymbol{g} = (g_{TI}, g_{SH}, g_{OP})$ in Section 3; yet empirical evidence on their relative effectiveness remains limited.

2.3 Game Theory in Cybersecurity

Game theory [9] is well suited to cybersecurity, where attackers and defenders choose strategies while anticipating each other's responses. Attacker decisions depend on asset value, resource requirements, exploit secrecy, and detection risk; defender decisions involve asset value, user trust, regulatory compliance, and resource allocation. This strategic interdependence makes game-theoretic modeling a natural fit.

Prior work spans network security [24], Kill-Chain-based defense modeling [12], optimal investment strategies from an economic perspective [8], deception tactics such as honeypots and disinformation [23], and vulnerability-sharing decisions within cyber alliances [2]. The Stackelberg game—where the defender commits to a security posture first and the attacker responds after observing it—is the dominant formulation [3], faithfully mirroring environments in which defenders set policies before attackers choose entry points.

This paper addresses two gaps in the literature. First, no existing study explicitly models the asymmetric effect of AI on both sides within a game-theoretic structure. Second, while prior work touches on individual ACD elements, none integrates CTI and ex-ante response investment—the core instruments through which defenders operationalize ACD—into a unified analytical framework.

3 Model

3.1 Setup

We model the strategic interaction as a three-stage Stackelberg game under complete information. The defender moves first (Stage 1), the attacker responds (Stage 2), and post-attack outcomes are realized (Stage 3). Complete information is justified by the attacker's ability to infer defender capability through reconnaissance along the Cyber Kill Chain.

Stage 1. The defender commits to a defense strategy vector:

$$z = (z_p,\, z_d,\, z_{r,ex}) \in \mathbb{R}^3_+, \tag{1}$$

comprising prevention investment z_p, detection investment z_d, and ex-ante CTI investment $z_{r,ex}$.

Stage 2. The attacker observes z and selects attack effort and evasion effort:

$$x \geq 0, \qquad e \in [0,\, \bar{e}]. \tag{2}$$

Stage 3. The attack outcome is realized. Post-attack response capability $z_{r,post}$ adjusts to the realized situation: greater ex-ante CTI investment raises response readiness, while a more intense attack strains the available response capacity.

Key model variables are summarized in Table 1. The ACD policy vector $\boldsymbol{g} = (g_{TI}, g_{SH}, g_{OP})$ captures all three instruments identified in Section 2; the empirical illustration in Section 4 focuses on g_{TI} (CTI capability), for which proxy data are available.

Table 1. Model Variable Definitions

Variable	Definition	Domain
x	Attack effort	$x \geq 0$
e	Evasion effort	$e \in [0, \bar{e}]$
$z_p, z_d, z_{r,ex}$	Prevention / Detection / CTI investment	≥ 0
$z_{r,post}$	Post-attack response capability	≥ 0
a	AI capability level	$[0, 1]$
g_{TI}, g_{SH}, g_{OP}	CTI / Information-sharing / Operational capability	$[0, 1]$

3.2 Key Functions

Post-attack response capability increases with ex-ante CTI investment and decreases with attack intensity:

$$z_{r,post} = \frac{\eta_{zr} z_{r,ex} + \eta_{op} g_{OP}}{1 + \eta_x x}, \quad \eta_{zr}, \eta_{op}, \eta_x > 0. \tag{3}$$

Attack success probability $q(x, z_p, z_d; a, \boldsymbol{g})$ increases in x and decreases in z_p, z_d. **Detection probability** $p_{det}(z_d, e; g_{TI}, g_{SH}, g_{OP})$ increases in z_d and decreases in e, with CTI capability exerting a stronger marginal effect than information sharing:

$$\frac{\partial p_{det}}{\partial g_{TI}} > \frac{\partial p_{det}}{\partial g_{SH}} > 0, \qquad \frac{\partial^2 p_{det}}{\partial g_{SH}^2} < 0. \tag{4}$$

Attacker payoff:

$$U_A = q \cdot R(V) - C_A(x; a) - C_{ev}(e; a) - p_{det} \cdot F. \tag{5}$$

Defender payoff:

$$U_D = -q \cdot L(V, z_{r,ex}, z_{r,post}) - C_D(z; a, \boldsymbol{g}). \tag{6}$$

3.3 AI Cost Asymmetry

AI reduces costs for both sides:

$$\frac{\partial C_A}{\partial a} < 0, \qquad \frac{\partial C_D}{\partial a} < 0. \tag{7}$$

We distinguish two conditions.

A1 (Attacker Advantage):

$$\left| \frac{\partial C_A}{\partial a} \right| > \left| \frac{\partial C_D}{\partial a} \right|. \tag{8}$$

A2 (Defender Advantage):

$$\left| \frac{\partial C_A}{\partial a} \right| < \left| \frac{\partial C_D}{\partial a} \right|. \tag{9}$$

3.4 Equilibrium and Propositions

The Stackelberg equilibrium is derived by backward induction:

$$\big(z^*,\, x^*(z^*),\, e^*(z^*)\big). \tag{10}$$

Proposition 1 (AI Asymmetry and Attack–Defense Balance).

$$\frac{\partial x^*}{\partial a} \begin{cases} > 0 & \textit{under A1 (attacker advantage),} \\ < 0 & \textit{under A2 (defender advantage).} \end{cases} \tag{11}$$

If AI lowers attack costs more (A1), greater AI intensifies attacks; if defense costs fall more (A2), AI deters attack effort. The net effect is an empirical question determined by which side extracts greater efficiency gains from AI.

Proposition 2 (Deterrence vs. Evasion Substitution).

$$g < \bar{g} \;\Rightarrow\; x^* > 0 \;\; \textit{and} \;\; \frac{\partial x^*}{\partial g} < 0. \tag{12}$$

$$g \geq \bar{g} \;\Rightarrow\; x^* = 0. \tag{13}$$

ACD reduces attack effort but does not achieve full deterrence below threshold $\bar{g}$; attackers substitute toward evasion rather than abandoning attacks entirely.

Proposition 3 (Strategic Value of CTI Investment).

$$z_{r,ex} \uparrow \Rightarrow\; U_D^* \uparrow \;\; \textit{then} \;\downarrow\; (\textit{inverted-U, direct effect}) \;\; \textit{and} \;\; x^* \uparrow \;\; (\textit{indirect effect}). \tag{14}$$

CTI investment improves response readiness and initially reduces expected losses, but beyond an interior optimum the strategic externality—whereby higher CTI investment raises attacker incentives—dominates, yielding an inverted-U payoff for the defender. The monotone increase in x^* confirms the indirect effect.

4 Analysis

4.1 Numerical Equilibrium

At baseline ($g_{TI} = 0.54$, $g_{SH} = 0.49$, $V = 10$, $F = 3$, $a = 0.3$), the equilibrium yields $p_{success} = 0.833$ (attacker dominance) and concentrates defender investment on ex-ante CTI ($z^*_{r,ex} = 0.972$) over prevention ($z^*_p = 0.000$).

P1: AI Asymmetry and Attack–Defense Balance Under A1, x^* rises monotonically as a increases from 0.3 to 0.8; under A2 it remains flat rather than strictly declining, which is consistent with Proposition 1 interpreted as a non-increasing condition rather than strict monotone decrease (See Figure 1). A1 support ranges from 32% to 69% across sensitivity combinations. A robustness check across 16 ($a_{pre} \times a_{post}$) combinations returns 69% P1 support, confirming $a = 0.3$ and $a = 0.8$ are not knife-edge values.

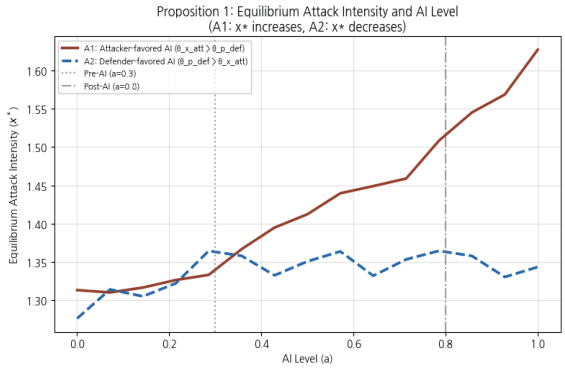


Fig. 1. P1: Equilibrium attack effort x^* under Conditions A1 and A2 as a increases from 0.3 to 0.8.

P2: Deterrence vs. Evasion Substitution ACD strengthening reduces e^* while x^* oscillates rather than declining monotonically, reflecting the simultaneous deterrence and evasion-substitution effects interacting at different ACD levels; the average directional tendency ($\partial x^*/\partial g < 0$) is preserved but strict monotonicity is not guaranteed (Figure 2): tighter ACD deters some attacks but raises evasion incentives for those that continue, so net effects partially cancel. The non-binding state ($g < \bar{g}$) holds across all 41 parameter combinations (100% robust).

P3: Strategic Value of CTI Investment Defender payoff follows an inverted-U in $z_{r,ex}$ while x^* rises monotonically (See Figure 3), confirming both effects of Proposition 3. The inverted-U is 100% robust; the dual effect holds in 60% of cases when η_{zr} exceeds a threshold, underscoring that investment quality matters more than quantity.

4.2 Empirical Illustration

The empirical component is an *illustration*, not a structural test: we assess whether real-world vulnerability patterns are directionally consistent with the model's predictions.

Data set We integrate CISA KEV, NVD, Mandiant, and FIRST EPSS covering 1,569 CVEs (2021–2026), with 2023 as the AI adoption breakpoint. Table 2 reports descriptive statistics. EPSS scores are collected at each CVE's KEV listing date and are subject to a collection-time bias toward older CVEs; they are therefore excluded from P1 pre/post comparisons and used only as a CTI proxy in P3.

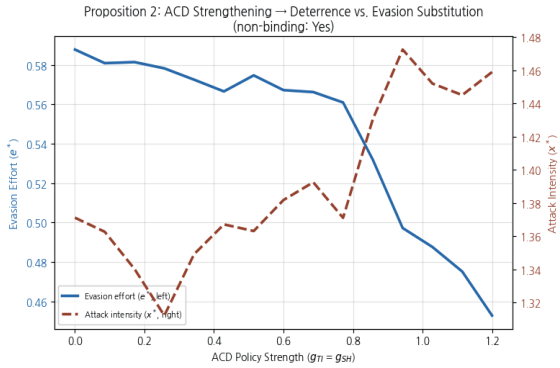


Fig. 2. P2: Attack effort x^* and evasion effort e^* as ACD policy g increases; non-binding state ($g < \bar{g}$) persists.

Table 2. Empirical Variable Definitions

Variable	Measurement	Role / Proxy for
$T_{A,realized}$ (days)	KEV DateAdded − NVD PublishedDate	Attack speed; proxy for x^* (P1)
$T_{D,due}$ (days)	CISA DueDate − NVD PublishedDate	Defense speed; risk-adjusted prioritization (P1)
`epss_score`	FIRST EPSS at KEV listing date	Exploit probability; CTI proxy (P3 only)
`exploit_state`	Mandiant 5-level score (0–4)	Actual exploitation activity; outcome var (P2/P3)
`risk_rating`	Mandiant Risk Rating (LOW=1 to CRITICAL=4)	CTI assessment quality; proxy for g_{TI} (P3)
`cvss_base`	NVD CVSS Base Score (0–10)	Vulnerability severity; control variable

P1: Attack–Defense Balance P1 predicts that AI compresses the attack–defense time race; we operationalize this as the relative shortening of attack versus defense timelines post-AI. We use $T_{A,realized}$ as the attack speed proxy and $T_{D,due}$ (CISA risk-adjusted patch deadline) as the defense proxy; shorter deadlines for higher-risk vulnerabilities (CRITICAL: 10 days vs. LOW: 47 days) make $T_{D,due}$ a prioritization indicator rather than a uniform cutoff. Restricting to CVEs exploited within one year ($N = 794$), the median $T_{A,realized}$ shortened 95.5% (112→5 days) post-AI while $T_{D,due}$ shortened 79.8% (126→26 days), with a 22-fold compression confirmed by Kaplan–Meier survival analysis (log-rank $p < 0.001$; Figure 4). The findings present mixed evidence. Median comparisons and Kaplan–Meier survival analysis support A1, while linear regression on TA,realized points toward A2, driven primarily by a small number of long-tail observations. Taken together, the weight of evidence leans toward A1, but A2 cannot be fully excluded with the current data.

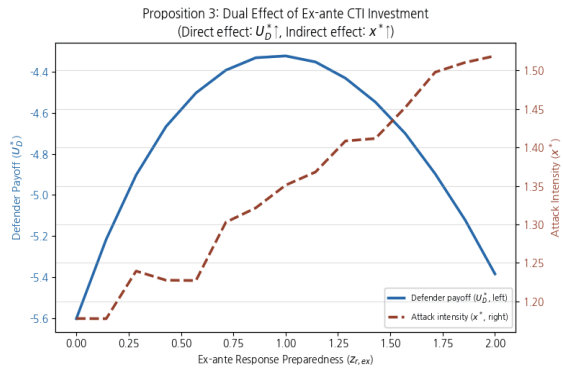


Fig. 3. P3: Defender payoff U_D^* (blue solid) and attacker effort x^* (brown dashed) as $z_{r,ex}$ increases; inverted-U direct effect and monotone indirect effect.

P2: Deterrence vs. Evasion Substitution Attack Complexity is unchanged post-AI ($\chi^2 = 0.062$, $p = 0.803$) and Exploitation State shows no significant shift ($\beta = -0.088$, $p = 0.098$), consistent with—though not confirming—the two effects canceling out; an alternative explanation is that the available proxies lack sufficient resolution to detect them separately. M-Trends Dwell Time corroborates this: fast detections ($\leq$1 week) rose from 30% to 40% while long-dwell intrusions (>6 months) fell from 12.3% to 8.0%, suggesting dual-effect polarization (See Figure 5).

P3: Strategic Value of CTI Investment Using Mandiant Risk Rating as a proxy for g_{TI} and Exploitation State as the outcome, CTI Risk Rating is positively and significantly associated with Exploitation State ($\beta = 0.238$, $p < 0.001$) and adds explanatory power beyond CVSS alone ($\Delta R^2 = 0.036$). Cross-validation via EPSS ($r = 0.321$ with Risk Rating; $r = 0.161$ with Exploitation State; both $p < 0.001$) further supports CTI assessment accuracy. As a supplementary analysis, $\beta_{epss \times AI} = +0.593$ ($p < 0.01$) suggests CTI's defensive value may strengthen post-AI, though this result is exploratory (Figure 6).

5 Discussion

5.1 Summary of Findings

The numerical and empirical analyses jointly support the three propositions, though with important qualifications on the strength of evidence.

Proposition 1 is supported by both numerical and empirical evidence: equilibrium attack effort rises under A1 conditions, and empirically, attack timelines compressed 22-fold while defense timelines shortened by less (95.5% vs.

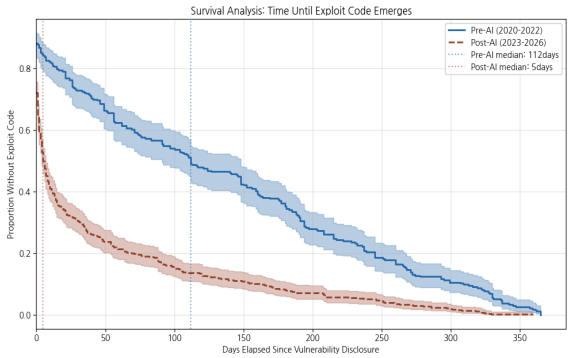


Fig. 4. P1: Kaplan–Meier survival curves for $T_{A,realized}$ (pre-AI median 112 days → post-AI median 5 days; log-rank $p < 0.001$).

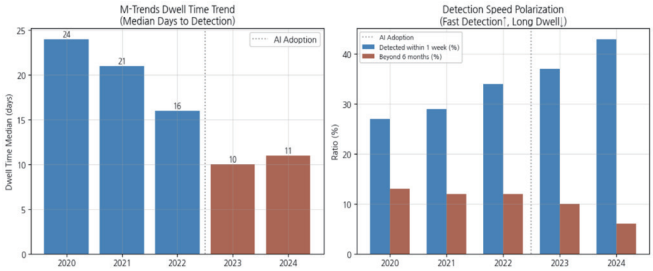


Fig. 5. P2: M-Trends Dwell Time pre/post AI. Left: annual median dwell time (days); right: share detected within 1 week vs. persisting beyond 6 months, illustrating fast-detection and long-dwell polarization consistent with $g < \bar{g}$.

79.8%). The A1/A2 switching across sensitivity analyses (32%–69%) reflects the condition-dependence Proposition 1 predicts, not model instability.

Proposition 2 finds the non-binding ACD state ($g < \bar{g}$) 100% robust numerically, and the empirical null result together with M-Trends polarization corroborate the dual-effect structure. What is needed is not a change in the type of ACD policy, but a higher level of commitment to the current approach, sufficient to cross the deterrence threshold $\bar{g}$.

Proposition 3 is confirmed by the inverted-U payoff structure (100% robust) and the empirical finding that CTI Risk Rating outperforms CVSS ($\Delta R^2 = 0.036$), jointly establishing that CTI investment quality is the binding constraint on defensive effectiveness.

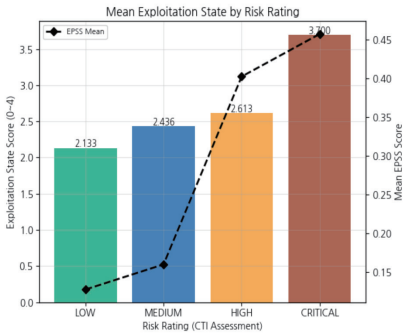


Fig. 6. P3: Exploitation State by Risk Rating and EPSS distribution.

5.2 Policy Implications

Three conclusions follow: (i) attackers currently extract greater AI efficiency gains, sustaining their traditional advantage; (ii) higher ACD investment is needed to cross the deterrence threshold; and (iii) CTI-based risk assessment and prioritization are central to effective ACD. We recommend that governments and organizations prioritize ACD policy with emphasis on AI-enhanced CTI, the domain where raising AI maturity yields the greatest defensive returns. AI application in proactive CTI remains underdeveloped relative to detection and response [14], implying significant upside potential as the technology matures.

5.3 Limitations

The analysis covers only KEV-listed vulnerabilities (confirmed exploits), excluding unsuccessful attacks, and does not capture organizational AI-maturity heterogeneity. All theoretical constructs (x^*, e^*, g_{TI}) are unobservable; empirical variables serve as proxies with necessarily imperfect mappings to the underlying model constructs. EPSS is subject to collection-time bias and restricted to P3. The 2023 AI breakpoint simplifies a gradual adoption process, and the empirical design cannot fully disentangle AI's effect from concurrent structural changes such as KEV catalog maturation, the introduction of CVSS v4.0, and evolving vendor patching practices. The continuous-a numerical analysis addresses theoretical sensitivity to parameter values, but not this empirical confounding. As the data are observational, causal isolation of AI's effect from concurrent trends is not possible; the game-theoretic model mitigates this by providing a theoretical mechanism for the observed changes.

6 Conclusion

This paper modeled attacker–defender interaction in an AI-enabled environment as a three-stage Stackelberg game and examined three propositions through nu-

merical equilibrium analysis and an empirical illustration. The attacker advantage persists in the post-AI period, as attackers currently extract greater efficiency gains from AI than defenders do. These results should not be interpreted as causal evidence that AI adoption has strengthened the attacker advantage. Rather, they provide an illustrative assessment of patterns consistent with the model, given the observational nature of the data, the indirect measurement of key theoretical constructs, and concurrent structural changes in the vulnerability landscape. However, ACD policy that incorporates AI-enhanced CTI can induce attackers to abandon attacks rather than merely evade detection, thereby shifting the competitive landscape toward defenders. The effectiveness of ACD increases with AI adoption maturity, and CTI is the domain where raising that maturity yields the greatest returns. We accordingly recommend that governments and organizations prioritize investment in AI-enhanced CTI as the highest-return instrument within an ACD framework. Specifically, future work should address how AI can be applied to CTI reliability assessment, adversary mapping, and exploitability-based prioritization—the operational bottlenecks that currently limit AI-enhanced CTI in practice.

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

References

1. Anthropic: Disrupting the first reported ai-orchestrated cyber espionage campaign. Tech. rep., Anthropic (November 2025), `https://www.anthropic.com/news/disrupting-AI-espionage`, accessed: 2026-04-30
2. Benkő, G., Biczók, G.: The cyber alliance game: how alliances influence cyberwarfare. In: Nordic Conference on Secure IT Systems. pp. 196–216. Springer (2024)
3. Bustamante-Faúndez, P., Labbé, M., Marianov, V., Ordóñez, F., et al.: Branch-and-price with novel cuts, and a new stackelberg security game. arXiv preprint arXiv:2405.09356 (2024)
4. Commonwealth of Australia: 2023–2030 australian cyber security strategy: A cyber secure nation. Strategy paper, Department of Home Affairs, Canberra, Australia (November 2023), `https://www.homeaffairs.gov.au/about-us/our-portfolios/cyber-security/strategy/2023-2030-australian-cyber-security-strategy`
5. Corsi, G., Kilian, K., Mallah, R.: Considerations influencing offense-defense dynamics from artificial intelligence. arXiv preprint arXiv:2412.04029 (2024)
6. Cybersecurity Strategic Headquarters: Cybersecurity strategy. Tech. rep., National center of Incident readiness and Strategy for Cybersecurity (NISC), Cabinet Secretariat, Government of Japan (December 2025), `https://www.cyber.go.jp/pdf/policy/kihon-s/cs_strategy2025_english.pdf`, provisional translation
7. Denning, D.E.: Framework and principles for active cyber defense. Computers & Security **40**, 108–113 (2014)
8. Ebel, A., Mitra, D.: Economics and optimal investment policies of attackers and defenders in cybersecurity. Journal of Cybersecurity **10**(1), tyae019 (2024)
9. Gibbons, R.S.: Game theory for applied economists. Princeton University Press (1992)

10. Hassanin, M., Moustafa, N.: A comprehensive overview of large language models (llms) for cyber defences: Opportunities and directions. arXiv preprint arXiv:2405.14487 (2024)
11. HM Government: National cyber strategy 2022: Pioneering a responsible and democratic cyber power. Policy paper, Cabinet Office, London, United Kingdom (January 2022), https://www.gov.uk/government/publications/national-cyber-strategy-2022
12. Kour, R., Karim, R., Dersin, P.: Modelling cybersecurity strategies with game theory and cyber kill chain. International journal of system assurance engineering and management pp. 1–12 (2025)
13. Lohn, A.J.: The impact of ai on the cyber offense-defense balance and the character of cyber conflict. arXiv preprint arXiv:2504.13371 (2025)
14. Mezzi, E., Massacci, F., Tuma, K.: Large language models are unreliable for cyber threat intelligence. In: International Conference on Availability, Reliability and Security. pp. 343–364. Springer (2025)
15. National Cyber Security Centre UK: Ncsc-ukglossary: Active cyber defense. National Cyber Security Centre UK (2026), https://www.ncsc.gov.uk/section/advice-guidance/glossary, accessed: April 30, 2026
16. National Institute of Standards and Technology: Nist glossary: Active cyber defense. NIST Computer Security Resource Center (2026), https://csrc.nist.gov/glossary/term/active_cyber_defense, accessed: April 30, 2026
17. National Security Office: National cybersecurity strategy. Tech. rep., Republic of Korea (February 2024), https://ncsc.go.kr/eng/cop/bbs/selectBoardArticle.do?bbsId=Notice_eng&nttId=143562, available at the National Cyber Security Center (NCSC)
18. Oesch, S., Hutchins, J., Austria, P., Chaulagain, A.: Agentic ai and the cyber arms race. Computer **58**(5), 82–85 (2025)
19. Palani, K., Kethar, J., Prasad, S., Torremocha, V.: Impact of ai and generative ai in transforming cybersecurity. Journal of Student Research **13**(2) (2024)
20. Potter, Y., Guo, W., Wang, Z., Shi, T., Li, H., Zhang, A., Kelley, P.G., Thomas, K., Song, D.: Frontier ai's impact on the cybersecurity landscape. arXiv preprint arXiv:2504.05408 (2025)
21. Swetha, T., Kumaran, U., Meena, V., Hameed, I.A.: Leveraging ai for enhanced cybersecurity: a comprehensive review. Discover Applied Sciences **7**(6), 584 (2025)
22. U.S. Department of Defense: Department of defense strategy for operating in cyberspace. Tech. rep., U.S. Department of Defense (July 2011), https://csrc.nist.gov/csrc/media/projects/ispab/documents/dod-strategy-for-operating-in-cyberspace.pdf, unclassified summary released in July 2011
23. Yang, Y.T., Zhu, Q.: When to deceive: A cross-layer stackelberg game framework for strategic timing of cyber deception. arXiv preprint arXiv:2505.21244 (2025)
24. Zhu, Q., Rass, S.: Game theory meets network security: A tutorial. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 2163–2165 (2018)

You Are Not My Teammate: Behavioral Fingerprint-based Detection of Suspicious Account Misuse

Dong Hwan Lee and Huy Kang Kim

School of Cybersecurity, Korea University, Republic of Korea
{20-10076, cenda}@korea.ac.kr

Abstract. Online games have been continuously affected by cyber threats such as game bots and gold farming. Game bots, which are automated programs that play on behalf of human users, significantly accelerate character progression and reduce the engagement of legitimate players, potentially leading to user churn. In addition, gold farming enables the monetization of in-game currency into real-world money, resulting in unfair profits. For these reasons, prior studies have primarily focused on detecting game bots and gold farming. However, in competitive Multiplayer Online Battle Arena (MOBA) games such as League of Legends, match outcomes and rankings are the primary objectives, where individual performance is more critical than in-game economic factors. Accordingly, account misuse such as account sharing and boosting has emerged as a major threat to fair competition. In this study, we propose a behavioral fingerprint-based detection method. Our approach analyzes and quantifies changes between a player's historical and recent in-game behaviors. Consequently, it enables the robust identification of suspicious account sharing and boosting, even in label-scarce environments. Experimental results show that behavioral fingerprints within the same account are distinguishable from those across different accounts, supporting rapid detection of suspicious account misuse even with limited labeled data.

Keywords: Behavioral fingerprint · Account misuse · Anomaly detection · Game security · Boosting · League of Legends

1 Introduction

Online games have evolved beyond entertainment into major social and economic platforms, giving rise to diverse cheating behaviors including game bots and gold farming [1]. However, in competitive Multiplayer Online Battle Arena (MOBA) games such as League of Legends, individual skill and rank are more critical than in-game economic assets, making account misuse such as account sharing and boosting major concerns [2]. In particular, when a player with abnormally high skill intervenes in a matchmaking environment intended for similarly skilled players, the balance of the game is disrupted. This inflicts repeated, unfair losses on legitimate players, directly degrading user engagement and accelerating churn

[3]. Thus, detecting and mitigating account misuse is critical in MOBA games from the perspective of game's security and fairness. To address these issues, Riot Games explicitly prohibits account misuse in its terms of service and states that violations may result in penalties such as service restrictions [4]. This indicates that the threats posed by boosting are recognized not as a regional concern, but as a universal challenge to competitive fairness in online gaming. Conventional login-based security mechanisms primarily focus on authentication of users and detection of abnormal login activities. However, after authentication, it is difficult to directly verify whether the actual player is the legitimate account owner. Furthermore, since boosting is performed by human players rather than game bots, it is difficult to detect using conventional approaches such as social profiling. To address these challenges, we propose an approach that detects suspicious account misuse based on players' gameplay fingerprints, focusing on the fact that a player's gameplay style and strategy shift when a different player controls the account. Fig. 1 illustrates the process of detecting suspicious activities based on changes in behavioral fingerprints caused by a player change. We apply a normalization technique that accounts for structural differences across positions, thereby reducing bias and effectively capturing detailed behavioral variations within the same position. Furthermore, by comparing past and recent behavioral fingerprints of a player, we can effectively identify suspicious accounts even in a label-scarce environment. The main contributions of this paper are as follows:

- We propose a behavioral fingerprint-based method that enables rapid triage of suspicious account misuse using only server-side match logs.
- We apply role-based normalization to reduce structural bias across positions and enable fair behavioral comparison.
- We show that intra-account behavioral fingerprints are distinguishable from inter-account fingerprints, supporting detection in label-scarce environments.

2 Related Work

2.1 Online Game Security and Cheating

With the growth of online games, various forms of cheating have continuously increased, leading to the development of diverse detection methods. Early studies systematically categorized cheating in online games and defined fairness as a core element of game security [5], followed by research that further classified cheating in MMORPG environments into game bots, account theft, and gold farming, along with corresponding detection methods [6]. Subsequent studies detected game bots using multimodal behavioral features [7] and self-similarity [8], and identified gold farming networks via trade graph analysis [9]. These studies primarily focus on detecting cheating behaviors driven by automation or economic profit. However, beyond cheating driven by automation or economic profits, account misuse such as account sharing and boosting has also been identified as a critical threat to game fairness, and prior studies have analyzed the

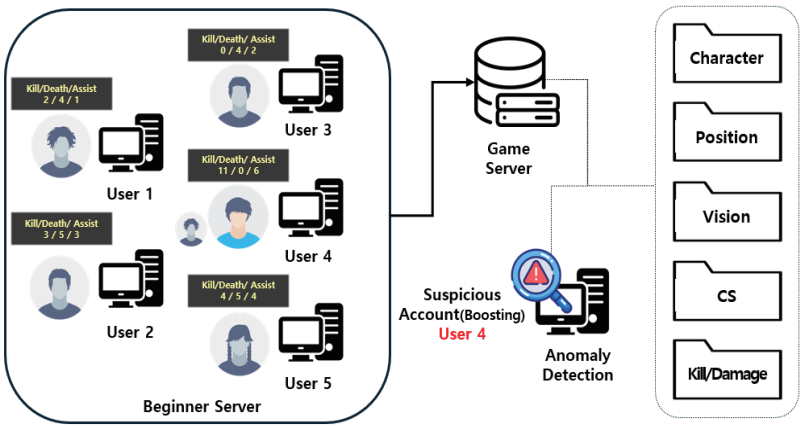


Fig. 1: Overview of the proposed approach for detecting suspicious account misuse based on behavioral fingerprint analysis.

concept of boosting and its negative impacts, including unfair matchmaking and the degradation of ranking system integrity [2, 10]. Nevertheless, studies that directly detect boosting using gameplay data remain limited. In MOBA games, particularly in League of Legends, prior research has explored matchmaking fairness and rank system design [10, 11], and player performance evaluation and match outcome prediction [12, 13]. However, despite this breadth of League of Legends research, no prior study has directly addressed the detection of account misuse such as account sharing and boosting using server-side match logs alone. This gap motivates the present work.

2.2 Behavioral and Biometric-based Approaches

Research aimed at identifying individuals based on players' behavioral patterns has also been continuously conducted. Early studies showed that player-specific characteristics could be captured through idle-time distributions during gameplay [14]. Subsequently, an approach was proposed that modeled the process of identity theft step by step and detected anomalous behavior by combining behavioral patterns with economic activities [15] or by analyzing server-side action sequences [16]. In FPS environments, behavioral fingerprints such as aim accuracy and movement patterns have been used to detect cheating [17], demonstrating that server-side behavioral analysis is feasible. User authentication using mouse and keystroke dynamics has also been explored, including account owner identification in League of Legends [18]. However, such approaches are constrained in real-world service deployment because they require client-side data collection. Accordingly, we complement prior work by detecting suspicious account misuse using only server-side match logs.

3 Background

3.1 Account Misuse

Account misuse refers to the use of a game account by an individual other than the legitimate owner, in violation of the platform's Terms of Service [4]. Two primary forms exist in competitive MOBA environments: account sharing, where another individual uses someone else's account, and boosting, where a highly skilled player plays on behalf of another user's account to artificially increase its rank. Both behaviors are distinct from account theft and cause a noticeable shift in the account's behavioral fingerprint regardless of motivation. Since the goal of this work is to detect accounts that show such a shift, our method does not need to distinguish between the two cases.

3.2 Position-based Game Structure and Early Game

In League of Legends, each player is assigned one of five positions — Top, Jungle, Mid, Bottom, and Support — per match, each with distinct roles and behavioral patterns. For example, Bottom and Jungle tend to focus on Creep Score (CS), whereas Support emphasizes vision control and team combat [12]. Due to these structural differences, position-based bias must be accounted for when analyzing behavioral fingerprints.

Similarly, the gameplay evolves over time from an early phase focused on individual play to a late phase dominated by team combat. The period up to approximately 10 minutes after the start of a match is defined as the early game, and it can have a significant impact on match outcomes [19]. During this phase, players primarily engage in individual play, allowing their personal play styles to be directly reflected. Due to these characteristics, early game data are suitable for analyzing players' distinctive behavioral features.

3.3 Behavioral Fingerprint

A behavioral fingerprint refers to a quantitative representation of a player's distinctive play style and strategy, based on the concept that individuals can be distinguished through their in-game behavior. In MOBA environments, this encompasses aspects such as champion selection preferences, farming behavior, and combat participation patterns, which inherently change when the actual player changes.

4 Method

4.1 Data Collection

We collected match data via the Riot Games API by randomly selecting seed accounts from ranked players on the Korean server and iteratively expanding the account set through match participant information. Since our proposed

approach analyzes behavioral patterns by comparing windows of consecutive matches within each account, sufficient per-account match records are essential to construct both the baseline and recent windows reliably. Because the number of available matches varied considerably across accounts, we selectively collected additional matches for accounts with insufficient records in later stages. We excluded accounts with fewer than 35 matches, yielding 100 accounts for analysis.

4.2 Feature Design

To design behavioral features, we extracted server-side match and timeline metrics from the Riot Games API and constructed derived features to capture behavioral changes within an account. Based on prior MOBA and League of Legends studies [10, 12, 13, 19], we extracted 12 derived features from server-side match logs and organized them into five categories: champion selection pattern, position distribution, farming behavior, vision score, and combat participation behavior. We computed farming behavior from the first 10 minutes of each match to capture individual early-game behavior. For features that vary significantly across positions, we applied role-based Z-score normalization.

Champion Selection Pattern Champion selection pattern captures a player's champion preference using Champion Pool Size and Champion Usage Entropy. Champion Pool Size measures the number of distinct champions used, while Champion Usage Entropy captures how evenly champion selections are distributed:

$$ChampionPoolSize = |\{Champion_k : k = 1, \ldots, N\}|, \quad H = -\sum_{i=1}^{C} p_i \log p_i \quad (1)$$

where $Champion_k$ denotes the champion played in the k-th match, and N is the total number of matches in the window. C denotes the number of distinct champions, p_i denotes the proportion of matches in which the i-th distinct champion was played, and H denotes the champion usage entropy.

Position Distribution Unlike champion selection, which spans a large and variable set of options, position is drawn from a small fixed set of five roles, allowing a direct proportion-based formulation. Position Distribution represents a player's position preference as a five-dimensional vector, where each dimension corresponds to the proportion of matches played in one of the five positions:

$$P_r = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(pos_i = r) \quad (2)$$

where N is the total number of matches, pos_i denotes the position selected by the player in the i-th match, $r \in \{\text{TOP, JUNGLE, MIDDLE, BOTTOM, SUPPORT}\}$, and $\mathbb{I}$ is the indicator function.

Farming Behavior We selected CS-based features to represent a player's farming ability. To more accurately capture individual behavior, we used CS per minute at 10 minutes. Since CS distributions differ substantially across positions, we applied role-based Z-score normalization:

$$CS_{10} = \frac{CS\ at\ 10\ min}{10}, \quad CS_{role_z} = \frac{CS_{10} - \mu_{role}}{\sigma_{role}} \tag{3}$$

To capture both performance and consistency across matches, we used $CS_{role_z_mean}$ and $CS_{role_z_std}$ as derived features.

Vision Score Vision Score captures vision control behavior, including ward placement and removal. To remove the effect of match duration and position-based differences, we first divided the raw Vision Score V by the match duration t (in minutes) and then normalized it within each position:

$$Vision_{role_z} = \frac{(V/t) - \mu_{role}}{\sigma_{role}} \tag{4}$$

Combat Participation Behavior Combat Participation Behavior was represented using Kill Participation (KP) and Damage Share (DS). KP measures a player's involvement in team kills, while DS measures the proportion of total team damage dealt by the player. We normalized both features within each position to reduce position-based bias:

$$KP = \frac{Kills + Assists}{Team\ Kills}, \quad DS = \frac{Damage\ Dealt}{Team\ Total\ Damage} \tag{5}$$

$$KP_{role_z} = \frac{KP - \mu_{role}}{\sigma_{role}}, \quad DS_{role_z} = \frac{DS - \mu_{role}}{\sigma_{role}} \tag{6}$$

4.3 Behavioral Fingerprint Modeling

To quantitatively model player behavior, we constructed behavioral fingerprints for each account based on match data ordered chronologically. We generated windows consisting of a fixed number of matches, each represented as a single 12-dimensional vector. For example, we summarized kill participation and damage share as their mean across matches in the window. We defined a baseline window to represent a stable gameplay style and a recent window to capture changes in behavioral fingerprints.

The baseline window used a relatively larger window size to reflect long-term behavioral patterns, whereas the recent window used a smaller window size to sensitively capture short-term shifts in gameplay style. We set candidate window sizes to 5, 10, 20, 25, and 30 matches, and the selection criteria are described in Section 5.3.

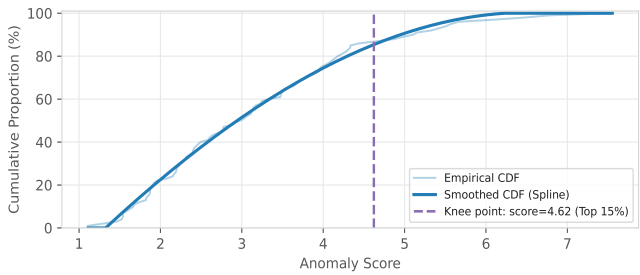


Fig. 2: Empirical CDF of anomaly scores. The Kneedle threshold is detected at score 4.62.

4.4 Anomaly Detection

To identify suspicious account misuse, we defined the change in behavioral fingerprints between the baseline and recent windows as an anomaly score [20], computed as the Euclidean distance between their feature vectors:

$$d = \|v_{baseline} - v_{recent}\|_2 \tag{7}$$

where $v_{baseline}$ and v_{recent} denote the feature vectors of each window. Prior to computation, we applied StandardScaler to ensure that no single feature disproportionately influences the distance. To determine the threshold in a data-driven manner, we applied the Kneedle algorithm [21] to the empirical CDF of anomaly scores, which identifies the knee point marking the boundary between the concentrated normal region and the sparse upper tail directly from the data without requiring an arbitrarily chosen percentile. We fitted a cubic smoothing spline to the CDF with the initial value set to zero to satisfy the boundary condition. We set the sensitivity parameter S to 1 to avoid premature knee detection, confirming consistency for $S = 2$ and $S = 3$. The resulting threshold at score 4.62 corresponded to the top 15%, as shown in Fig. 2. We therefore classified accounts with anomaly scores above this threshold as suspicious [22].

5 Experimental Design and Analysis

5.1 Experimental Setup

We collected data from January to March 2026 using the Riot Games API, targeting ranked solo queue matches on the League of Legends Korean server. We selected the Korean server because it provides an active and highly competitive ranked environment, and a prior study has also reported account sharing sanctions in the Korean competitive gaming context [2]. We excluded accounts with fewer than 35 matches to ensure sufficient data for simultaneously constructing candidate baseline windows of up to 30 matches and a recent window of 5

Table 1: Intra-account and Inter-account Distance Comparison

Distance Type	Window Pair	Mean	Median
Intra-account distance (maximum per anchor window)	5 vs 35	3.293	3.025
	10 vs 35	2.285	2.197
	20 vs 35	1.282	1.243
	25 vs 35	0.875	0.796
	30 vs 35	0.509	0.444
Inter-account distance	5 vs 5	5.179	5.105
	10 vs 10	4.747	4.791
	20 vs 20	4.496	4.521
	25 vs 25	4.492	4.513
	30 vs 30	4.489	4.488
	35 vs 35	4.513	4.500

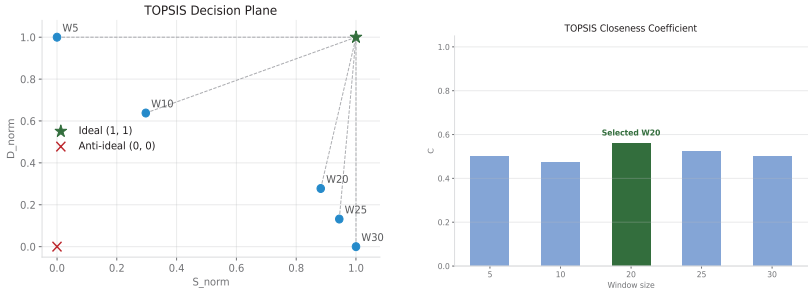
matches, yielding 100 accounts for analysis. To examine whether the proposed method responds to behavioral changes, we conducted a controlled behavioral-change validation using 10 accounts. In each case, the original account owner intentionally altered their gameplay style and strategy across a minimum of 5 consecutive ranked matches while playing the account personally. For example, we instructed participants to adopt a more passive combat style than usual while focusing more aggressively on farming. This validation does not constitute actual account sharing or boosting, but serves to examine whether the proposed behavioral fingerprint can detect controlled behavioral changes within an account.

5.2 Validity Verification

We conducted validity verification to confirm the core assumption underlying the proposed detection approach — namely, that distances between behavioral fingerprints within the same account are significantly smaller than those between different accounts. For each account, we set the most recent 35 matches as the reference interval and constructed nested windows of sizes 5, 10, 20, 25, 30, and 35 matches, all anchored at the most recent match within the reference interval. We represented each window as a 12-dimensional behavioral fingerprint vector comprising features such as champion selection diversity and position distribution. We compared intra-account distances and inter-account distances, following the same standardization procedure described in Section 4.4. Table 1 summarizes the representative intra- and inter-account distances. Even the largest intra-account distance (3.293 for 5 vs. 35) remained substantially below the smallest inter-account distance (4.489 for 30 vs. 30), confirming a clear separation between the two distributions. These findings demonstrate that behavioral fingerprints stably capture account-level patterns and are applicable to detecting suspicious account misuse.

5.3 Baseline Window Size Selection

To determine the optimal window sizes, we measured two criteria for each candidate window. Stability (S_{norm}) measures how consistently a behavioral finger-



(a) Decision Plane (b) Closeness coefficient C_w per window

Fig. 3: TOPSIS-based baseline window size selection. (a) Each candidate window is plotted in the normalized criterion space. (b) The 20-match window achieves the highest closeness coefficient and is selected as the baseline window.

print is maintained when transitioning to an adjacent larger window, measured as the Euclidean distance between adjacent window pairs, with inverted Min-Max normalization applied so that higher scores correspond to greater stability. Sensitivity (D_{norm}) measures how well a window captures the difference between short-term and long-term behavioral patterns, measured as the Euclidean distance between each window and the maximum-size window of 35 matches, normalized via standard Min-Max normalization. To objectively select the optimal baseline window size, we employed TOPSIS [23] with normalized stability and sensitivity as evaluation criteria. Letting $\mathbf{v}_w = (S_{\text{norm},w}, D_{\text{norm},w})$ denote the criterion vector of window w, we defined the ideal solution $\mathbf{v}^+ = (1, 1)$ and anti-ideal solution $\mathbf{v}^- = (0, 0)$. We computed the distance to each reference point and the closeness coefficient C_w as:

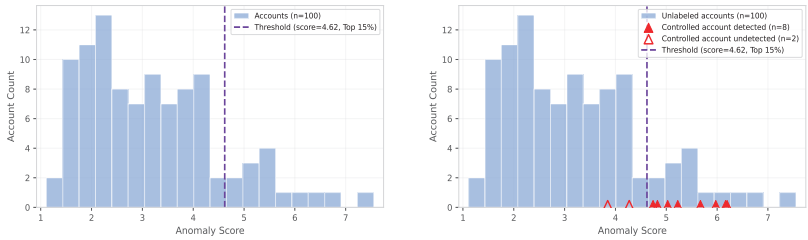
$$d_w^{\pm} = \|\mathbf{v}_w - \mathbf{v}^{\pm}\|_2, \qquad C_w = \frac{d_w^-}{d_w^+ + d_w^-} \tag{8}$$

As shown in Fig. 3, the 20-match window achieved the highest closeness coefficient ($C = 0.558$), so we selected it as the baseline window. Since we considered fewer than 5 matches likely to compromise statistical stability, we established 5 matches as the minimum for the recent window.

6 Results and Discussion

6.1 Suspicious Account Detection Results

We computed anomaly scores based on a 20-match baseline window and a 5-match recent window, with the baseline constructed from the 20 matches immediately preceding the recent window. We present the score distribution across 100 accounts in Fig. 4(a), ranging from 1.11 to 7.55 with a mean of 3.19 and a



(a) Anomaly scores (100 accounts) (b) Anomaly scores (Controlled accounts)

Fig. 4: Distribution of anomaly scores with threshold at score 4.62 (dashed line). (a) Most accounts concentrate in the lower region with few in the upper tail. (b) Among 10 controlled accounts (red triangles), 8 are detected and 2 are not.

median of 3.01. Applying the threshold of 4.62, we classified 15 accounts (the top 15%) as suspicious. To validate detection capability, we introduced 10 controlled behavioral-change accounts into the pipeline. As shown in Fig. 4(b), 8 of the 10 controlled accounts (80%) exceeded the threshold. The two undetected accounts recorded scores of 4.27 and 3.85, which may reflect a relatively moderate degree of behavioral change compared to more pronounced real-world cases. Our feature-level analysis of the detected accounts revealed that $CS_{role_z_std}$ and position distribution consistently exhibited the largest variation, emerging as primary indicators of a gameplay style shift. These results indicate that our approach can effectively identify suspicious accounts in a label-scarce environment, demonstrating its potential as a rapid triage tool for further analysis.

6.2 Discussion

While our proposed approach demonstrated the ability to effectively identify suspicious account misuse, several limitations warrant consideration. First, we conducted experiments on 100 accounts, which may not be sufficient to validate generalizability, and future work should construct larger-scale datasets. We instantiated our framework using League of Legends features, and testing it on other games remains future work as well. Second, since we explicitly instructed participants to alter specific behaviors, the observed changes may partly reflect these instructions, and this controlled setting may not fully capture the degree of gameplay style shifts observed in actual misuse cases. Finally, we defined the baseline window as the 20 matches immediately preceding the recent window. If account misuse persists over an extended period, behavioral patterns from the misuse period may be incorporated into the baseline window, diluting the detectable change and limiting the method to short-term or sporadic misuse rather than long-term, persistent cases. Future research may consider dynamically configuring the baseline window to mitigate this issue. While false positives cannot be fully excluded, we designed the proposed approach to serve as a rapid triage

tool, and we treat accounts exceeding the threshold as suspicious candidates for further analysis, not as confirmed misuse cases.

7 Conclusion

We proposed a behavioral fingerprint-based detection method for identifying suspicious account misuse such as account sharing and boosting in MOBA game environments. Unlike prior studies focused on game bot detection or player identification, we addressed the problem of detecting behavioral changes within the same account using a sliding-window framework and an unsupervised approach to ensure applicability in label-scarce environments. Our experimental results confirmed that intra-account distances were significantly smaller than inter-account distances, and the data-driven threshold derived from the Kneedle algorithm identified 15 suspicious accounts in the upper tail of the score distribution. Our validation using 10 controlled accounts further demonstrated an 80% detection rate. These results suggest that our proposed approach can serve as a rapid triage tool, enabling early detection of suspicious accounts. Among the features, we identified $CS_{role_z_std}$ and position preference as primary indicators of play style shift. However, the detection results on unlabeled accounts remain at the level of suspicion, as verified labels for real-world misuse cases were unavailable. Future work will validate detection performance using larger datasets with verified labels and extend the framework to handle long-term misuse scenarios.

Acknowledgements. This work was supported by Korea Internet & Security Agency (KISA) grant funded by the Korea government (PIPC) (No. 2780000036, Masters and Doctoral Program for Advanced Talent Development in Personal Data Protection and Responsible Use).

References

1. Kim, J.E., Tsvetkova, M.: Cheating in online gaming spreads through observation and victimization. Network Science **9**(4), 425–442 (2021)
2. Conroy, E., Kowal, M., Toth, A.J., Campbell, M.J.: Boosting: Rank and skill deception in esports. Entertainment Computing **36**, 100393 (2021)
3. Kang, H., Suh, C., Kim, H.K.: Match experiences affect interest: Impacts of matchmaking and performance on churn in a competitive game. Heliyon **10**(3), e24891 (2024)
4. Riot Games: Terms of Service. `https://www.riotgames.com/en/terms-of-service`, last accessed 2026/06/27
5. Yan, J., Randell, B.: An investigation of cheating in online games. IEEE Security & Privacy **7**(3), 37–44 (2009)
6. 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)
7. Kang, A.R., Jeong, S.H., Mohaisen, A., Kim, H.K.: Multimodal game bot detection using user behavioral characteristics. SpringerPlus **5**(1), 523 (2016)

8. Lee, E., Woo, J., Kim, H., Mohaisen, A., Kim, H.K.: You are a game bot!: Uncovering game bots in MMORPGs via self-similarity in the wild. In: Proceedings of the Network and Distributed System Security Symposium (NDSS), pp. 1–15 (2016)
9. Kwon, H., Mohaisen, A., Woo, J., Kim, Y., Lee, E., Kim, H.K.: Crime scene reconstruction: Online gold farming network analysis. IEEE Transactions on Information Forensics and Security **12**(3), 544–556 (2016)
10. Fan, G., Zhang, C., Wang, K., Li, Y., Chen, J., Xu, Z.: CUPID: Improving battle fairness and position satisfaction in online MOBA games with a re-matchmaking system. Proceedings of the ACM on Human-Computer Interaction **8**(CSCW2), 1–39 (2024)
11. Claypool, M., Decelle, J., Hall, G., O'Donnell, L.: Surrender at 20? Matchmaking in League of Legends. In: Proceedings of the IEEE Games Entertainment Media Conference (GEM), pp. 1–4. IEEE (2015)
12. De Bois, M., Parmentier, F., Puget, R., Tanti, M., Peltier, J.: PandaSkill-player performance and skill rating in esports: Application to League of Legends. IEEE Transactions on Games **17**(4), 989–1002 (2025)
13. Hojaji, F., McIlroy, R.E., Dupuy, A., Pedroni, G., Toth, A.J., Campbell, M.J.: Deep learning techniques for identifying KPIs in League of Legends: Win prediction, map navigation, and vision control. Computers in Human Behavior Reports **19**, 100718 (2025)
14. Chen, K.T., Hong, L.W.: User identification based on game-play activity patterns. In: Proceedings of the ACM SIGCOMM Workshop on Network and System Support for Games (NetGames), pp. 7–12 (2007)
15. Woo, J., Choi, H.J., Kim, H.K.: An automatic and proactive identity theft detection model in MMORPGs. Applied Mathematics & Information Sciences **6**(1S), 291S–302S (2012)
16. Kim, H., Yang, S., Kim, H.K.: Crime scene re-investigation: A postmortem analysis of game account stealers' behaviors. In: 2017 15th Annual Workshop on Network and Systems Support for Games (NetGames), pp. 1–6. IEEE Computer Society (2017)
17. Alayed, H., Frangoudes, F., Neuman, C.: Behavioral-based cheating detection in online first person shooters using machine learning techniques. In: Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG), pp. 1–8 (2013)
18. Beserra, I.D.S., Camara, L., da Costa-Abreu, M.: Using keystroke and mouse dynamics for user identification in the online collaborative game League of Legends. In: Proceedings of the International Conference on Imaging for Crime Detection and Prevention (ICDP), pp. 1–6 (2016)
19. Gaina, R., Nordmoen, C.: League of Legends: A study of early game impact. School of Electronic Engineering and Computer Science, Queen Mary University of London, London (2018)
20. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(3), 1–58 (2009)
21. Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a "Kneedle" in a haystack: Detecting knee points in system behavior. In: Proceedings of the 31st International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 166–171 (2011)
22. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review **22**(2), 85–126 (2004)
23. Tzeng, G.-H., Huang, J.-J.: Multiple Attribute Decision Making: Methods and Applications. CRC Press (2011)

A Blockchain-based Security Policy Management Framework for Multi-Organizational VPP/SCADA Environments

Chanwoong Ji[1][0009−0000−9008−660X], Soojin Lee[2][0000−0003−1690−8577], and Seung-Hyun Seo[3][0009−0001−1323−2746]

[1] Dept. of Computer and Information Security, Hanyang University, Seoul 04763, Korea
ichanwoong@hanyang.ac.kr
[2] Center for Creative Convergence Education, Hanyang University ERICA, Ansan 15588, Korea
tssn195@hanyang.ac.kr
[3] The Division of Electrical Engineering, Hanyang University ERICA, Ansan 15588, Korea
seosh77@hanyang.ac.kr

Abstract. In multi-organizational virtual power plant (VPP) and SCADA environments, each organization operates its security appliances independently, making it difficult for a single organization to detect advanced persistent threats spanning multiple organizations. This paper proposes a blockchain-based security policy management framework based on Hyperledger Fabric and a private IPFS. An asymmetric CP-ABE scheme preserves the confidentiality of organizational logs while enabling policy sharing among participating organizations. An optimistic deployment scheme reduces initial policy enforcement latency by applying signature-verified policies before the ledger commit, while Fabric consensus proceeds in parallel. Final policy activation is performed only after the corresponding ledger metadata is committed and verified. The policy lifecycle is immutably recorded on-chain to support auditability. A prototype evaluation on a Raspberry Pi testbed shows that the proposed scheme enforces policies earlier than the consensus-first approach, with the latency gain corresponding to the ledger commit latency.

Keywords: VPP/SCADA · Blockchain · Security policy management.

1 Introduction

SCADA-based power operation environments are becoming increasingly complex with the growing adoption of renewable and distributed energy resources. Virtual Power Plants (VPPs) in particular require multiple organizations, including Distribution System Operators (DSOs), VPP clouds, and local Energy Management Systems (EMSs), to jointly manage these resources. Each organization operates its own security solutions, such as firewalls, intrusion detection

systems, and honeypots, but manages them independently. As a result, threats detected by one organization are not shared with others, which limits coordinated defense across organizational boundaries.

The lack of security information sharing among organizations makes it difficult to respond effectively to multi-stage cyber attacks. An attacker may first compromise a vulnerable system in the supply chain and then move laterally across interconnected organizations before deploying a stealthy backdoor such as BPFDoor in a critical control system. Because only partial attack traces remain within each organization, no single organization can reconstruct the entire attack chain or establish a coordinated response. Current environments also lack mechanisms to verify the authenticity of shared policies and to systematically manage rollback histories, which limits auditability and post-incident analysis.

Blockchain has been adopted for securely sharing distributed security information and transparently managing policy change histories, since it ensures data integrity and provides a traceable record of changes. Several blockchain-based approaches have therefore been proposed to enhance power system security [9].

However, existing studies have several limitations. Approaches that store data entirely on-chain [5,7] are not suitable for handling large-scale security logs and do not support integrated security policy management across multiple organizations. FACOS [8], which combines an on-chain/off-chain architecture with a Trusted Execution Environment (TEE), improves storage efficiency but remains unsuitable for SCADA environments due to the latency introduced by Byzantine consensus. Moreover, it does not support the complete security policy lifecycle, including policy creation, sharing, deployment, modification, and rollback.

To address these limitations, this paper proposes a multi-organizational security policy management framework based on a hybrid architecture integrating Hyperledger Fabric [2] with a private InterPlanetary File System (IPFS) [3]. The proposed framework focuses on security policy management rather than policy generation while providing inter-organizational confidentiality and policy auditability. The main contributions of this paper are as follows:

- **Asymmetric Information Sharing:** Ciphertext-Policy Attribute-Based Encryption (CP-ABE) restricts each security log to the originating organization and the policy generator, while the resulting policies remain accessible to all participants, enabling coordinated responses without exposing sensitive operational information.
- **Optimistic Policy Deployment:** A policy is stored in IPFS and applied immediately with a predefined TTL, while consensus proceeds in parallel. This provisional enforcement enables rapid response without waiting for the ledger commit, and the policy becomes active only after the transaction is committed through Fabric consensus.
- **Auditable Policy Lifecycle Management:** State transitions among `pending`, `active`, and `rolled_back` are recorded as immutable on-chain transactions, each with its timestamp, reason, and the organizations involved, enabling end-to-end traceability of policy changes.

2 Related Work

This section reviews three lines of research that motivate the design of the proposed framework: blockchain-based security for power and industrial control systems, on-chain/off-chain hybrid architectures, and multi-organizational confidential sharing with fine-grained access control.

2.1 Blockchain for Smart Grid / SCADA

Blockchain has been widely studied to improve the security and resilience of smart grid and SCADA environments. Liang *et al.* [7] store smart meter measurements on-chain to detect false data injection attacks through distributed voting, and Kirkman *et al.* [5] protect SCADA historians against ransomware, showing that the blockchain must be separated from the control path to meet real-time requirements. These studies, however, focus on protecting operational data or system infrastructure rather than on sharing and managing security policies across organizations.

2.2 Hybrid Architectures and Secure Sharing

To improve scalability, recent frameworks adopt hybrid on/off-chain architectures that keep only hashes or metadata on-chain and store large data off-chain, commonly on IPFS, and many of them employ Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to protect the off-chain data with fine-grained access control.

Table 1 compares representative frameworks with the proposed approach. In FACOS [8], data becomes available only after consensus, so deployment latency depends on Byzantine consensus. TrustShare [1] targets confidential threat intelligence sharing rather than policy deployment or lifecycle management. ConsortiumSec [11] treats policies as static objects and supports only policy updates without explicit lifecycle states. Existing frameworks therefore provide secure storage and fine-grained access control but support neither real-time policy deployment nor policy lifecycle management, which the proposed framework addresses through CP-ABE-based asymmetric information sharing, Optimistic Policy Deployment, and auditable lifecycle management.

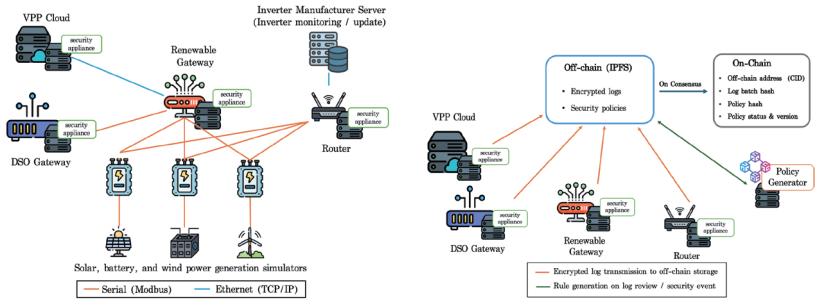
3 System Architecture and Threat Model

This section describes the participants and blockchain structure of the proposed framework, and defines the trust and threat assumptions it makes. It then presents the design goals for secure cross-organizational sharing of logs and policies in a multi-organizational VPP/SCADA infrastructure where each organization operates its own security appliances independently.

Table 1: Comparison of multi-organizational secure sharing frameworks.

	Blockchain Platform	Off-chain Storage	Access Control	Real-time Pre-deployment	Policy Lifecycle
FACOS [8]	Fabric	BFT	ABE	×	×
TrustShare [1]	Fabric	IPFS	CP-ABE	×	×
ConsortiumSec [11]	Fabric	IPFS	Distributed ABE	×	△
Proposed framework	Fabric	Private IPFS	CP-ABE	✓	✓

× not supported △ partially supported ✓ supported



(a) Multi-organizational VPP/SCADA environment

(b) On/off-chain data flow of the proposed framework

Fig. 1: Overview of the proposed multi-organizational security policy management framework

3.1 System Architecture

The target environment is a multi-tier, multi-organizational virtual power plant (VPP) and SCADA infrastructure. As shown in Figure 1a, it involves several operating entities such as the distribution system operator (DSO), a VPP cloud (C-VPP), and distributed energy resources (DER), each independently operating heterogeneous security appliances such as firewalls and endpoint detection and response (EDR) systems. To support secure policy management across these organizations, we introduce a *policy generator* that cross-analyzes multi-organizational logs to derive security policies and then distributes and manages them across all organizations. We assume the policy generator to be a trusted entity that does not misuse the collected information, which justifies its access to multi-organizational logs for policy derivation. Attribute keys are issued by A key generation center (KGC) which is assumed to be fully trusted issues attribute keys. This paper focuses on the distribution of generated policies rather than on policy generation itself.

The proposed framework adopts a hybrid on-chain/off-chain architecture that combines a permissioned Hyperledger Fabric blockchain [2] with a private IPFS, as shown in Figure 1b. The blockchain is organized as a consortium in which

the policy generator and each organization form their own membership service provider and operate an endorsing peer on a single shared channel. Policy transactions are endorsed under `OutOf(3, {DSO, CVPP, RenewGW, Router})`, which excludes the policy generator so that the organizations enforcing a policy are also the ones approving it. Endorsed transactions are then ordered by a Raft-based ordering service that is itself operated by the participating organizations, and committed to the replicated ledger of every peer.Each organization also runs an agent attached to its security appliances, which collects, encrypts, and transmits logs and applies deployed policies. The blockchain stores only metadata such as content identifiers (CIDs), hashes, and policy status, while the private IPFS stores the encrypted logs and the policies themselves.

3.2 Threat Model

We consider an Advanced Persistent Threat (APT) that unfolds across multiple organizations. The attacker first compromises a vulnerable supply-chain system to establish a foothold, then expands through lateral movement across interconnected organizations, and eventually deploys a stealthy kernel-level backdoor such as BPFDoor on a critical control system. Since only fragmented traces remain within each organization, no single organization can observe the complete attack chain.

We assume that the attacker may compromise nodes within participating organizations to forge security logs or inject malicious policies, while the policy generator remains trusted as stated above. The primary security objectives are therefore the integrity of deployed policies and the confidentiality of the logs each organization generates.

3.3 Design Goals

To address the problems above under the stated trust and threat assumptions, the proposed framework is designed to meet the following five goals.

(G1) Log confidentiality. Each organization's operational logs must be accessible only to the originating organization and the policy generator.

(G2) Consistent multi-organizational enforcement. A policy derived from one organization's logs must be consistently propagated to and applied across all relevant organizations.

(G3) Protection against tampering and forgery. Deployed policies must be protected from tampering through signature and hash verification, and from forgery through multi-organizational endorsement.

(G4) Immediate enforcement before ledger commit. A policy must be applicable without waiting for the ledger commit, and the commit must proceed in parallel with enforcement so that real-time response is not impaired. Enforcement should complete within the sub-second range.

(G5) Lifecycle management and auditability. The full history of a policy, from creation to confirmation and rollback, must be retained as an immutable on-chain record for post-hoc auditing.

4 Proposed Framework

This section describes the overall workflow of the proposed multi-organizational security policy management framework. The framework consists of four operational phases: Log Registration, Log Access Control, Optimistic Policy Deployment, and Policy Lifecycle Management.

We use the following notation. Let MK and PP be the master key and public parameters of CP-ABE [4], and let PG be the policy generator. Each organization Org_i holds an attribute key SK_i, and PG holds a privileged key $\mathsf{SK}_{\mathsf{PG}}$. In addition, $H(\cdot)$ denotes a cryptographic hash function, and σ_{PG} denotes the generator's signature on policy metadata.

4.1 Log Registration

Each participating organization periodically collects security logs generated by its security appliances. To reduce storage and communication overhead, logs generated within a predefined time window are grouped into a log batch L_i. A log batch L_i is encrypted under the access policy $\mathbb{A}_i = (\mathsf{PG} \vee \mathsf{Org}_i)$ to produce the ciphertext $\mathsf{CT}_i = \mathsf{Enc}(\mathsf{PP}, \mathbb{A}_i, L_i)$. Under this access policy, only the originating organization holding the attribute key SK_i and the Policy Generator holding the privileged key $\mathsf{SK}_{\mathsf{PG}}$ can decrypt the ciphertext.

The ciphertext is uploaded to the private IPFS, while only its metadata is recorded on the Hyperledger Fabric blockchain. The metadata includes the content identifier (CID), the batch hash $H(\mathsf{CT}_i)$, the organization identifier, and the corresponding collection time window. The batch hash enables integrity verification of the entire log batch without requiring metadata for individual log entries.

4.2 Log Access Control

When security log analysis is required, an authorized entity submits a log access request for the desired log batch. The blockchain first retrieves the corresponding log metadata, including the content identifier(CID), and returns it to the requester. Using the retrieved CID, the encrypted log batch is fetched from the private IPFS.

The requester then attempts to decrypt the ciphertext. By the access policy $\mathbb{A}_i$ defined above, only the originating organization and the policy generator can succeed, while any other requester fails to satisfy the policy.

4.3 Optimistic Policy Deployment

After analyzing the collected security logs, the policy generator generates a security policy, signs it with its digital signature σ_{PG}, and uploads the signed policy to the private IPFS. The IPFS returns a CID, which uniquely identifies the stored policy. The policy generator then submits the `CreatePolicy` transaction to register the policy metadata on the blockchain. The metadata includes the policy

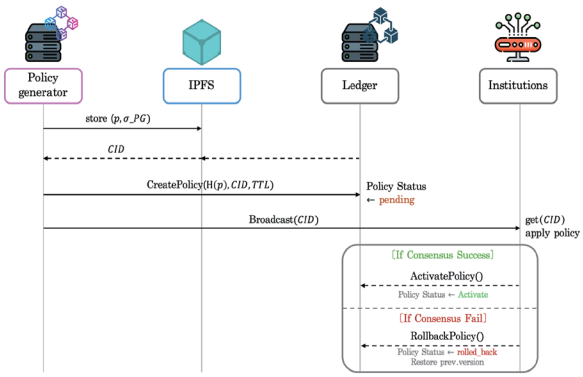


Fig. 2: Proposed Optimistic Policy Deployment and Management

hash $H(P)$, the CID, the policy version, the deployment status, the time-to-live (TTL), and the Policy Generator's digital signature. The newly registered policy is initially assigned the `pending` state.

Figure 2 illustrates the proposed *Optimistic Policy Deployment*. The policy generator immediately disseminates the CID and the signed metadata to all participating organizations. Using the received CID, each organization retrieves the policy from the private IPFS and verifies it using the policy generator's signed metadata, including the policy hash. If the verification succeeds, the policy is enforced provisionally at each organization before the `CreatePolicy` transaction is committed. Meanwhile, blockchain consensus proceeds in parallel through endorsement, ordering, validation, and commit. After the `CreatePolicy` transaction is committed, each organization compares the committed ledger metadata with the previously verified signed metadata. If they are consistent, the `ActivatePolicy` transaction changes the policy state to `active`. Otherwise, or if the TTL expires, the `RollbackPolicy` transaction changes the state to `rolled_back` and restores the previously verified policy version. This approach reduces the initial policy enforcement latency by applying a verified policy before ledger commit, while final activation remains dependent on Fabric consensus and ledger validation.

4.4 Policy Lifecycle Management

The proposed framework manages the lifecycle of each security policy by recording every state transition on the blockchain. As shown in Fig. 2, each policy progresses through the `pending`, `active`, and `rolled_back` states. Every transition is recorded together with its timestamp, triggering event, and participating organizations. The complete lifecycle of a policy can be retrieved through a single `GetPolicy` query, and the replicated ledger across blockchain peers provides immutable and tamper-resistant audit records.

Rollback serves as a safeguard for policies that have not yet been confirmed, and each rollback event is likewise recorded for subsequent auditing. Policies in the `active` state are regarded as verified and are not subject to rollback. Instead, any modification creates a new policy version that undergoes the same deployment process. By distinguishing rollback for unconfirmed policies from updates to confirmed ones, the framework preserves policy integrity while maintaining an auditable policy history.

5 Security and Design Analysis

This section evaluates the security properties of the proposed framework under the trust assumptions and threat model defined in Sections 3.1 and 3.2.

5.1 Confidentiality of Operational Logs

Each log is encrypted using CP-ABE under the access policy `(PolicyGenerator OR {Originating Institution})` and stored off-chain, so only the policy generator and the originating institution can decrypt it. Without the required attribute keys, an adversary cannot recover plaintext from observed ciphertexts even if a participating institution is compromised or the private IPFS is exposed, which ensures log confidentiality against compromised participants and off-chain storage exposure (G1).

5.2 Integrity and Authenticity of Policies

Tampering during distribution. The policy is stored off-chain, while its hash is bound to the policy generator's signed metadata and is later recorded on-chain. Before applying a policy, each institution verifies the retrieved policy against the hash in the signed metadata. Since the metadata is signed by the policy generator, an adversary who modifies the off-chain policy cannot produce a matching hash without forging that signature, so any modification is detected before the ledger commit. Once the `CreatePolicy` transaction is committed, each institution cross-checks the applied policy against the on-chain record, so that a policy inconsistent with the ledger is never confirmed to the `active` state (G3).

Forgery during policy generation. A policy can be confirmed to the `active` state only if it satisfies the endorsement rule `OutOf(3, {DSO, CVPP, RenewGW, Router})`, which excludes the policy generator so that the operating institutions check the policies it produces. Consequently, no single institution can unilaterally confirm a policy, and an adversary must compromise at least three of the four institutions to have a forged policy activated. We note that this quorum is an application-level approval rule enforced by Fabric endorsement rather than a Byzantine agreement protocol, since the ordering service in our deployment is crash fault-tolerant (G3).

5.3 Consistency and Availability under Optimistic Deployment

The proposed *Optimistic Policy Deployment* enables early policy enforcement before ledger commit while ensuring eventual consistency. A policy is applied once its signed metadata is verified, and is confirmed as `active` only after the `CreatePolicy` transaction is committed and the ledger metadata is verified. Because the policy generator disseminates the same CID and signed metadata to every institution, all institutions apply an identical policy version, which yields consistent enforcement across organizations (G2). Enforcement does not wait for the ledger commit and completes in approximately 85 ms in our prototype (G4). To prevent unconfirmed policies from remaining applied indefinitely, a TTL-based rollback mechanism revokes the applied policy and restores the most recent verified version if the ledger metadata is inconsistent, the transaction fails, or the TTL expires. If the initial policy is rolled back, the default policy (version 0) is recovered, so the system never enters a policy-less state. Consistency is therefore restored once the ledger commit completes, and the window of possible divergence is bounded by the TTL.

5.4 Auditability

All policy lifecycle events, including creation, confirmation, and rollback, are immutably recorded on-chain with their timestamps, reasons, and participating institutions, and are replicated across all peers so that no single institution can modify them. A single `GetPolicy` query retrieves the complete lifecycle of a policy, enabling post-incident tracing of when a policy was applied or revoked and which institutions endorsed the decision. Unlike approaches that maintain only the current state, the framework preserves the entire policy history, thereby providing auditability and accountability (G5).

5.5 Limitations

The current design has three limitations. CP-ABE key management is simplified, in that the prototype assumes a static attribute set and does not address key revocation or attribute updates. The policy generator and the KGC are single points of trust and failure, so their compromise may allow a malicious policy to be enforced for at most one TTL interval, or expose the logs of all institutions. Rollback also restores the policy state without undoing the effects already produced during enforcement, so the TTL bounds the exposure window rather than eliminating it. In addition, the framework guarantees that the same policy is distributed to every institution, but it does not collect explicit acknowledgements confirming that each institution has actually enforced it. A threshold KGC, multi-authority ABE, and acknowledgement-based enforcement confirmation are left for future work.

Table 2: End-to-end pipeline stage breakdown by log size on RPi 5 (unit: ms)

Stage	1 KB	4 KB	16 KB
CP-ABE encrypt	39.65	39.87	40.14
IPFS upload	12.91	13.22	13.26
On-chain register	1865.2	1877.5	1931.3
Total	1917.8	1930.6	1984.7

6 Evaluation

In this section, we evaluate the proposed framework on a prototype testbed in two respects. We measure the end-to-end latency for a policy to traverse the full pipeline and be registered on-chain, and we compare the policy application latency of optimistic deployment against the consensus-first approach.

6.1 Experimental Setup

The experiments are conducted on four Raspberry Pi 5 nodes (8 GB, ARM64, Ubuntu 24.04) representing the four participating organizations (DSO, C-VPP, RenewGW, Router), while the policy generator and the KGC run as separate services on an Apple M4 Pro (48 GB). The blockchain layer is built on Hyperledger Fabric v2.5.10. Each of the four organizations and the policy generator operates both an endorsing peer and an orderer node, forming a five-node Raft ordering cluster on a single channel, and the block size is left at the Fabric default. Off-chain storage uses a private IPFS (Kubo 0.32.1) [6] isolated by a swarm key. Access control is implemented with CP-ABE (BSW07) using the `rabe` 0.4 [10] library, where each log body is encrypted with AES and only the symmetric key is sealed under CP-ABE in a hybrid scheme. We use SHA-256 for hashing and ECDSA (P-256) for signatures. All reported values are the average over 100 runs.

6.2 End-to-end Pipeline Latency

We measured the end-to-end latency from log generation to on-chain registration, decomposed into encryption, off-chain storage, and blockchain registration, while varying the log size from 1 KB to 16 KB.

As shown in Table 2, increasing the log size by a factor of 16 has only a marginal effect on the overall latency. CP-ABE encryption stays near 40 ms and IPFS upload near 13 ms regardless of size, because CP-ABE encrypts only the symmetric key, IPFS stores small log batches, and the blockchain records only metadata. In contrast, on-chain registration consistently dominates, accounting for approximately 97 % of the total.

This registration delay is not caused by cryptographic operations or data storage, but by the structural characteristic of Hyperledger Fabric, where a transaction waits for the `BatchTimeout` before being included in a block. Therefore,

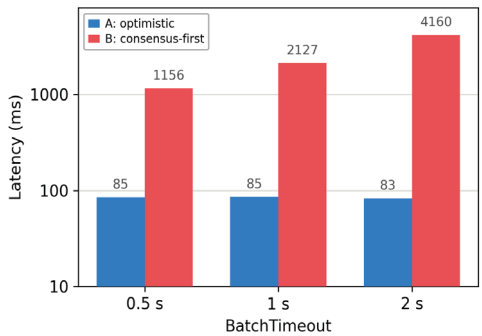


Fig. 3: (A)Optimistic vs. (B)consensus-first deployment latency on a logarithmic scale as the BatchTimeout increases.

the primary bottleneck of the pipeline lies in the blockchain consensus process. This observation motivates the proposed *Optimistic Policy Deployment*, which moves the ledger commit latency out of the initial enforcement path, thereby reducing the time until a policy takes effect while preserving policy integrity through subsequent ledger verification.

6.3 Policy Deployment Latency: Optimistic vs. Consensus-first

We evaluated the deployment efficiency of the proposed scheme by deploying the same policy under the same network environment and measuring two enforcement points. Optimistic deployment (A) is the time at which the policy is applied right after being stored in IPFS, whereas consensus-first deployment (B) is the time at which it is applied after the ledger commit completes. Their difference $B - A$ is therefore the ledger commit latency that the proposed approach removes from the initial enforcement path.

Figure 3 shows that the enforcement time of optimistic deployment (A) remains nearly constant at approximately 85 ms regardless of the `BatchTimeout`, since the policy is applied without waiting for the ledger commit. In contrast, consensus-first deployment (B) grows proportionally with the `BatchTimeout`, so $B - A$ widens as the commit latency increases, reaching up to 4,078 ms in the longest setting. These results show that the proposed scheme reduces the time until a policy takes effect by decoupling enforcement from the ledger commit, while final activation still relies on Fabric consensus.

7 Conclusion

This paper proposed a security policy management framework for multi-organizational VPP/SCADA environments, built on a hybrid architecture combining Hyperledger Fabric with a private IPFS. CP-ABE protects the confidentiality of orga-

nizational security logs while the generated policies are shared among all participants, and off-chain pre-storage enables policy enforcement before the ledger commit. The deployment, confirmation, and rollback history of each policy is recorded immutably on-chain to ensure auditability. Future work will extend the framework to incorporate the policy generation process.

Acknowledgments. This work was supported by the Industrial Technology Alchemist Project (RS-2025-02653102) funded by the Ministry of Trade, Industry and Resource (MOTIR, Korea).

References

1. Ali, H., Buchanan, W.J., Ahmad, J., Abubakar, M., Khan, M.S., Wadhaj, I.: TrustShare: Secure and trusted blockchain framework for threat intelligence sharing. Future Internet **17**(7), 289 (2025)
2. Androulaki, E., Barger, A., Bortnikov, V., et al.: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference (2018)
3. Benet, J.: IPFS – content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561 (2014)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy (SP). pp. 321–334 (2007)
5. Kirkman, S.S., Fulton, S., Hemmes, J., Garcia, C., Wilson, J.C.: A blockchain architecture to increase the resilience of industrial control systems from the effects of a ransomware attack. ACM Transactions on Cyber-Physical Systems **8**(1), 1–23 (2024)
6. Kubo: IPFS implementation in Go. https://github.com/ipfs/kubo, version 0.32.1
7. Liang, G., Weller, S.R., Luo, F., Zhao, J., Dong, Z.Y.: Distributed blockchain-based data protection framework for modern power systems against cyber attacks. IEEE Transactions on Smart Grid **10**(3), 3162–3173 (2019)
8. Liu, C., Hou, C., Jiang, T., Ning, J., Qiao, H., Wu, Y.: FACOS: Enabling privacy protection through fine-grained access control with on-chain and off-chain system. IEEE Transactions on Information Forensics and Security **19**, 7060–7073 (2024)
9. Mollah, M.B., Zhao, J., Niyato, D., Lam, K.Y., Zhang, X., Ghias, A.M.Y.M., Koh, L.H., Yang, L.: Blockchain for future smart grid: A comprehensive survey. IEEE Internet of Things Journal **8**(1), 18–43 (2021)
10. rabe: Rust attribute-based encryption library. https://github.com/Fraunhofer-AISEC/rabe, version 0.4
11. Sha, K., Yue, K.B., Wei, W., Wu, Y., Koduru, M., Vuchuru, P.: ConsortiumSec: Blockchain-based distributed security framework for consortium applications. ACM Distributed Ledger Technologies: Research and Practice **4**(4), 1–23 (2025)

Analyzing the Effect of Native-Language Transformation on Password Guessing

Hamin Kim[1], Minjeong Cho[2], Yeog Kim[3], and Kiwook Sohn[1] Changhoon Lee[1]*

[1] Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Republic of Korea
{hamin99,chlee,kiwook}@seoultech.ac.kr

[2] Training and Education Center for Active Cyber Security Experts, Department of Computer Science and Engineering, Seoul National University of Science and Technology, Republic of Korea
chomj@seoultech.ac.kr

[3] Research Institute of Electrical and Information Technology, Seoul National University of Science and Technology, Seoul, Republic of Korea
yeogkim@gmail.com

Abstract. Password-guessing models learn from leaked passwords as stored, yet a stored English-looking string may not be the user's intended form. For Chinese, Japanese, and Korean users, pinyin, romaji, or QWERTY keystroke traces can survive into the stored string. To our knowledge, whether reconstructing them before training changes what a generator recovers has not been tested across model families. We define a deterministic, language-aware reconstruction and apply it as representation-level preprocessing, not as a rule inside one model's grammar. Holding each generator family fixed, we evaluate that reconstruction on PassGPT, PassGAN, PassRVAE, and PCFG at matched budgets of 10^7 and 10^8 guesses, over splits that are line-aligned for Japanese and near-aligned for Korean. The reconstruction is our primary contribution. At 10^8 guesses it raises Korean PassGPT recovery from 12.577% to 25.901% (+13.324 pp) and Japanese PCFG from 15.035% to 21.535% (+6.500 pp) under a chosen hiragana-trigram terminalization, yet costs PassGAN 5.610 pp on Korean and 3.864 pp on Japanese, and leaves Japanese PassGPT unchanged within evaluation-sampling error (+0.019 pp). Reconstruction is therefore a per-model, per-language modeling choice rather than a default preprocessing step, and where it helps a language-blind model can under-guess such passwords within a fixed budget.

Keywords: password guessing · PCFG · deep learning · generative models · data preprocessing

1 Introduction

Passwords remain the dominant low-cost authentication mechanism online. Because users reuse words, names, dates, keyboard patterns, and language-specific

* Corresponding author.

habits, leaked corpora are also training data for offline guessing models. Classic and neural generators alike exploit the concentration of password distributions: a small fraction of likely candidates often recovers a meaningful fraction of a held-out breach, so the order in which a model allocates probability mass is operationally important.

Most studies treat a password as the exact string stored by the service, which is natural for authentication but not always for modeling user intent. A Korean user may think of a Hangul word and type it through a QWERTY keyboard, a Japanese user may begin with romaji before IME conversion, and a Chinese user may rely on pinyin or character conversion. The stored string is then English-like while carrying a non-English trace, and the modeling question is whether the generator should learn that surface or a reconstructed language-aware representation.

The answer is not obvious. A transformation may shorten a sequence, expose repeated linguistic units, and reduce sparsity, but it may also destroy surface regularities, expand the alphabet, reduce duplicate sampling diversity, or create an evaluation mismatch. We therefore treat transformation as a model-conditional strategy rather than a universal improvement.

We run the same paired comparison across four generator families with different inductive biases: PassGPT (autoregressive), PassGAN (adversarial), PassRVAE (latent-variable), and PCFG (explicit grammar). The aim is not to rank them, but to show when one transformation helps, fails, or changes meaning across those biases.

Our primary contribution is the transformation itself: a deterministic reconstruction mapping Korean keyboard traces, Japanese romaji, and Chinese pinyin back into their native representation, applied as representation-level preprocessing rather than as a rule inside one model's grammar. Second, we measure its worth across the four families at matched budgets of 10^7 and 10^8 guesses, in-distribution (Section 7): at 10^8 Korean PassGPT gains 13.324 pp and Japanese PCFG 6.500 pp, while the same reconstruction costs PassGAN 5.610 pp on Korean and 3.864 pp on Japanese. These come from one training run per condition, so we report observed magnitudes rather than calibrated effects, and the PassGAN loss is a first-class finding rather than a caveat. Third, reconstruction alone is not the method: how the reconstructed run is decomposed into grammar terminals is a separate, decisive axis whose best chunk length is language-specific (Table 5). Fourth, we report every cell separately with a bound on its evaluation-sampling error (Section 7), rather than an average that would have hidden both the Korean gain and the PassGAN loss. Why particular cells gain remains a working hypothesis we do not measure here.

2 Related Work

Probabilistic context-free grammars (PCFGs) learn templates over letters, digits and symbols and enumerate candidates in decreasing probability order [1]; a semantically enhanced variant adds 43 types of semantic information over seven-

teen leaked databases in four languages [6]. Both operate on the stored surface string, so the structure they exploit must already be visible there.

Neural generators replace templates with learned distributions: adversarial models such as PassGAN [2], latent-variable models such as PassRVAE [8] and PassDNVAE [15], and recurrent or convolutional next-character models [10]. Transformer and LLM approaches treat generation as autoregressive language modeling, with PassGPT [3] and LLM-based guessers [13, 14] recovering many unseen passwords, and unified benchmarks now compare these generators directly [12].

A separate line observes that password choice is shaped by the user's native language: Chinese web passwords show heavy pinyin and locale-specific use [5], and language-specific guessers build Chinese syllable structure into the model [11]. Closest to our setting, Yan and Ku add Asian-language input-method and pronunciation grammars to a modern PCFG and report a 2 to 14 percent gain in cracking success [9].

We introduce neither a new generator nor a new strength metric; we isolate the representation variable, and because an offline attacker is bounded by how many candidates they can try, we evaluate finite-budget recovery only and state a budget with every claim. Unlike approaches that induce representations from data [4, 7], our transform is explicit and deterministic. Unlike grammars and locale studies that model native structure already visible on the surface [5, 6], and unlike Yan and Ku, who add input-method rules to one PCFG while leaving the surface unchanged [9], we reconstruct the trace into its native representation as a preprocessing step and evaluate that same reconstruction across PassGPT [3], PassGAN [2], PassRVAE [8] and PCFG [1, 6] under matched budgets. To our knowledge no prior study reconstructs Korean, Chinese and Japanese input traces before training and evaluates the effect across four generator families in one controlled comparison.

3 Problem Setting and Threat Model

3.1 Stored surface and intended representation

The original condition is the stored, authentication-facing string, here English-like or ASCII-like; the transformed condition is the language-aware representation produced by a deterministic conversion rule. The transformation is not assumed to be semantically perfect: it is an intervention that changes which units the generator sees. The security target remains the stored password, so a transformed generator is useful only if its candidates can be mapped back. For Korean that inversion is comparatively direct; for Japanese and Chinese it can be ambiguous, so the transformed condition is a controlled modeling experiment rather than a deployment recipe.

3.2 Attacker model and budgets

We consider an offline adversary with an auxiliary training set from the same population who submits a finite candidate list against a held-out test set; success

is the fraction recovered within the budget. We use 10^7 and 10^8 guesses, large enough to reveal model behavior and small enough to evaluate four families locally.

3.3 Corpus sources and balanced data reporting

The source collections are listed at the corpus level in Table 1.

After cleaning, language selection, transformation and an 80/20 train/test split, the files are summarized in Table 2, symmetrically across languages. Two facts qualify the Chinese condition. Its two sides come from one pool of 6,352,060 strings but were split independently, so unlike Korean and Japanese they are not line-aligned; and the Chinese original pool has no duplicates whereas the transformed test file has 126,190 duplicate lines among 1,132,788 distinct forms, 14,106 of which occur more than once. With 1,270,411 and 1,258,978 lines the two files admit no line-for-line correspondence, so no pre-image count is available, and the Chinese deltas are computed over different held-out sets and denominators. We therefore do not read them as a paired comparison.

The Korean and Japanese transformed test files contain no repeated lines, while their originals contain 118 and 39. A deterministic per-line map can only create repeats, never remove them, so the two sides are not a line-wise image of one another and a collision cannot surface in the transformed files at all: the zero one would measure there is a property of how the corpora were assembled, not evidence that the reconstruction is injective. Re-applying the reconstruction to the Korean original test set gives the direct count instead: 101 collapse events, all of size two, against 1,247,980 inputs.

We verified test-in-train membership on the exact files used. Overlap is zero for both Chinese splits and for Korean and Japanese transformed; the Korean and Japanese original splits hold 270 and 356 such entries, 0.02% and 0.03% of those test sets. Falling only in the original condition, the residual overlap can favor only the original baseline.

The Korean PassGPT rows carry one further qualification whose direction and size are both known. The two pipelines differ in one cleaning step: the transformed one discarded bcrypt and MySQL41 hash records and the original did not, so 8,096 hash strings remain in the original test set and none in the transformed one; scanning the full 10^8 candidate list confirms the generator recovered none of them. In addition 34,453 transformed test records have no counterpart in the original set, against 42,799 the other way. The bias therefore has a determinate sign: unrecoverable records enlarge the original denominator, depressing the original baseline and flattering the transformed condition. Section 7 bounds the hash-record and denominator components at under 0.09 pp each; the non-correspondence itself is not corrected, so that part of the bias is signed but not quantified. We call the comparison near-paired rather than strictly paired.

Table 1. Source collections used to construct the multilingual password corpus. Counts are the documented sizes of the source releases, which are credential records rather than distinct passwords. Table 2 reports the Chinese, Japanese, and Korean subsets actually used after preprocessing.

Source	Records
Collection #1	2.7 billion
Collection #2–#5	2.2 billion
CompilationOfManyBreaches	3.2 billion
A 12-billion-record credential compilation	13 billion
Other Korean and Chinese password datasets	about 7.8 million

Table 2. Language-level file sizes after preprocessing and train/test splitting. Japanese is split line-for-line across representations; Korean is near-paired and the two Chinese sides were split independently from a pool of identical total size (Section 3.3). Test columns are physical line counts, and Table 4 divides by the *distinct* size, smaller wherever a file repeats: 1,239,415 Korean original, 1,100,995 Japanese original, 1,132,788 Chinese transformed. PassGPT is the exception: its Korean original run used a separate file of 1,247,980 entries (Section 7).

	Original (lines)		Transformed (lines)	
Language	Train	Test	Train	Test
Korean	4,958,563	1,239,533	4,958,563	1,239,533
Chinese	5,081,649	1,270,411	5,093,082	1,258,978
Japanese	4,404,131	1,101,034	4,404,131	1,101,034

4 Methodology

4.1 Pipeline

Figure 1 shows the pipeline. We introduce no new generator; we hold the generator families fixed and vary the representation, asking whether the same information becomes more or less usable under each model's inductive bias.

4.2 Transformation by language

Figure 2 illustrates the three language-specific mappings. Korean reorganizes English keyboard traces into Hangul syllable structure, so a long Latin keystroke sequence becomes a shorter and more structured sequence of Korean units. Japanese maps romaji-like traces into hiragana units, which can compress repeated syllabic patterns but can also remove surface regularities an ASCII model learned well. Chinese is the hardest case: pinyin and character representations create a large unit space, and our module operates only on the longest alphabetic run, dropping the digits and symbols that the Korean and Japanese modules keep. Chinese passwords differing only outside that run therefore reconstruct to the same form, which is the mechanism behind the collisions of Section 3.3 and a property of our implementation rather than of Chinese itself.

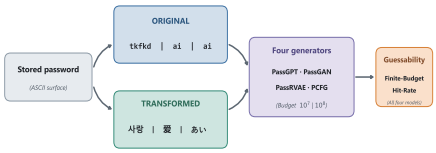


Fig. 1. Representation-controlled evaluation pipeline.

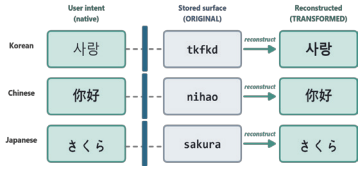


Fig. 2. Examples of the transform: stored surface (original) mapped to a native representation (transformed).

Transformation rules. The reconstruction is deterministic, not a probabilistic input-method decoding, so the transformed corpus is fully determined by the original. Korean maps dubeolsik key symbols to jamo and composes syllables at U+AC00 $+ 588c + 28v + f$ for initial c, medial v, final f, accepting at least two Hangul syllables and a Hangul ratio ≥ 0.6. Japanese lowercases, rewrites m as n, drops apostrophes and hyphens, then applies longest-match romaji-to-hiragana with gemination and syllabic-n rules, accepting a hiragana ratio ≥ 0.6. Chinese normalizes the diaeresis vowel to v, expands zero-initial syllables, segments greedily against a table of legal initial-final pairs, and maps the syllable sequence to characters along a single best path. A string no module accepts is left unchanged, so the corpus mixes reconstructed and untouched entries, heavily weighted to the latter: the reconstruction alters 19.05% of Korean and 16.02% of Japanese held-out records, leaving 80.95% and 83.98% byte-identical. The intervention is thus far narrower than a corpus-level framing suggests, which also qualifies the Korean result: with four fifths of the records byte-identical, the gain cannot be attributed to the altered subset alone without a script-subset breakdown we do not report.

4.3 Model families

The four families differ in how a representation change reaches them. PassGPT predicts each token from context, so reconstruction helps if it shortens the sequence and hurts if the vocabulary becomes sparse. PassGAN does not optimize likelihood over the held-out strings, so its output depends on sampling diversity, which a changed character distribution can reduce. PassRVAE samples from a latent space, where holding transformed characters in the vocabulary is useless unless those regions are sampled often enough to reach a finite guess list. PCFG learns explicit terminals, so it is sensitive to how a run is decomposed; we compare the whole run against characters, bigrams and trigrams.

4.4 Experimental setup

Table 3 lists the configuration. We give the PassGPT tokenizer per language and condition because they are not uniform: the Korean and Japanese transformed models use character vocabularies built on the reconstructed script, while the

Table 3. Experimental setup. Tokenizer rows give the type and the number of vocabulary entries actually present on disk.

Item	Original	Transformed
PassGPT tokenizer, Korean	byte-level BPE, 99 character, 4,923	
PassGPT tokenizer, Chinese	byte-level BPE, 99 byte-level BPE, 99	
PassGPT tokenizer, Japanese	byte-level BPE, 99 character, 165	
PassGPT architecture	GPT-2, 8 layers, 12 heads, 768 hidden, 1024 positions	
Input truncation	16 characters	
Effective batch size	512 (128×4 gradient accumulation)	
PCFG implementation	`pcfg_cracker` (Weir-style probabilistic grammar)	
Guess budgets	10^7 and 10^8 unique candidates	

Chinese transformed model reuses the original byte-level vocabulary. The tokenizer therefore changes with the condition, so the PassGPT deltas measure the preprocessing pipeline including its tokenizer rather than the representation in isolation; that non-uniformity is also why we report no likelihood result (Section 7).

4.5 Metrics

Guessability is the finite-budget hit-rate. With T the deduplicated held-out set and G(B) the first B unique candidates, $\textit{hit-rate}(B) = |G(B) \cap T|/|T|$ at B = 10^7 and 10^8; both sides are deduplicated, so repeats waste neither budget nor denominator. For PassGAN we also report the unique ratio of the generated list. In the transformed condition G(B) and T are both expressed in the transformed representation, so the hit-rate measures recovery within that representation rather than after mapping candidates back to the stored surface.

Hit-rate is therefore the only outcome we report; the reason we report no likelihood is given in Section 7.

5 Results

5.1 Overall finite-budget hit-rate

Table 4 reports the central finite-budget result. The same transformation has very different effects depending on the model and language. Figures 3 and 4 visualize the per-cell cracking curves and the transformation delta.

The largest positive cell is Korean PassGPT (+7.049 pp at 10^7, +13.324 pp at 10^8); a second is Japanese PCFG under the chosen hiragana-trigram terminalization (+4.269 pp and +6.500 pp).

The counterexamples matter equally. PassGAN is worse than original for Korean and Japanese and near-tied for Chinese, and its lists also lose diversity: the unique ratio at 10^8 falls from 0.851 to 0.618, 0.778 to 0.620 and 0.841 to 0.738 for Chinese, Korean and Japanese. PassRVAE is weak or budget-dependent, with sub-point 10^7 changes and a reversal only in the Korean 10^8 cell.

Table 4. Budgeted hit-rate (%) by model, language, and representation. Delta is transformed minus original in percentage points. †Chinese deltas are not a paired comparison (Section 3.3).

		10^7 guesses			10^8 guesses		
Model	Lang.	Orig.	Trans.	Δ	Orig.	Trans.	Δ
PassGPT [3]	KO	1.954	9.003	+7.049	12.577	25.901	+13.324
	CH†	6.154	6.677	+0.523	19.982	20.789	+0.807
	JA	5.385	5.432	+0.047	15.449	15.468	+0.019
PassGAN [2]	KO	4.989	3.220	−1.769	14.516	8.906	−5.610
	CH†	3.422	4.180	+0.758	12.272	12.368	+0.096
	JA	3.087	1.599	−1.488	9.257	5.393	−3.864
PassRVAE [8]	KO	8.588	9.418	+0.830	24.661	22.235	−2.426
	CH†	6.081	6.752	+0.671	19.462	19.570	+0.108
	JA	5.087	5.085	−0.002	15.184	14.004	−1.180
PCFG [1]	KO	24.788	24.678	−0.110	36.814	36.939	+0.125
	CH†	10.105	9.272	−0.833	16.629	14.786	−1.843
	JA	9.008	13.277	+4.269	15.035	21.535	+6.500

Table 5. PCFG terminalization ablation at 10^8 guesses. Values are script-subset hit rates (%); the best setting per language is in bold.

	Terminalization			
Language	Whole run	Char	Bigram	Trigram
Chinese	0.000	28.211	**46.721**	3.823
Japanese	0.000	41.459	48.236	**51.473**
Korean	**59.045**	42.610	54.550	58.020

Japanese PassGPT is the sharpest null. At 10^8 the two conditions recover 170,098 and 170,309 of a comparably sized held-out set, a difference of 0.019 pp whose 95% evaluation-sampling intervals overlap almost completely: on this evidence reconstruction produces no measurable gain here. With the PassGAN losses it makes the same point from the opposite side, since the reconstruction that doubles Korean PassGPT recovery is worthless in one cell and harmful in others, which is why we treat it as a per-cell modeling choice rather than a preprocessing default.

5.2 PCFG terminalization

How the reconstructed run is decomposed is decisive, and the best unit is language-specific (Table 5): trigrams for Japanese, bigrams for Chinese, which collapses to 3.823 under trigrams, and the whole Hangul run for Korean. A whole run depends on exact repetition, which is why script-subset recovery is 0.000 for Chinese and Japanese but 59.045 for Korean, whose Hangul runs do recur.

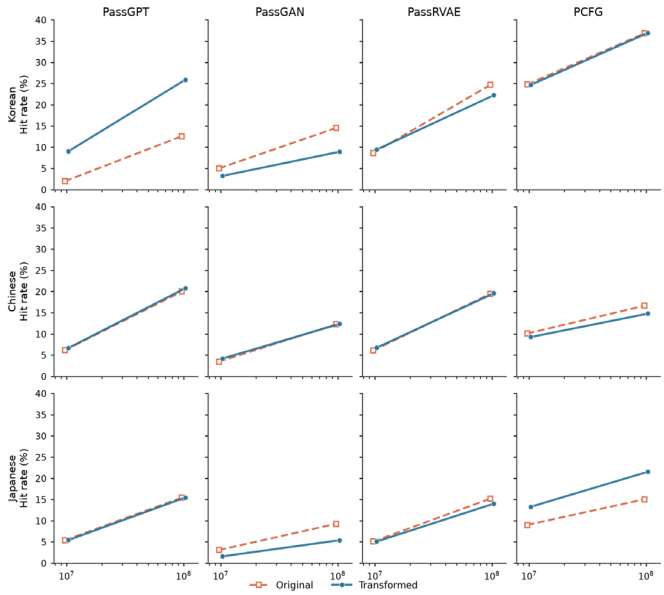


Fig. 3. Language- and model-separated cracking curves at 10^7 and 10^8 guesses.

6 Analysis

The dominant pattern is interaction: the effect reads as model by language by representation, not as one global coefficient. Korean PassGPT is the clearest gain, because reconstruction turns a long Latin keystroke trace into Hangul syllables and removes arbitrary surface transitions. Japanese is model-dependent: PCFG improves strongly once hiragana trigram chunks become reusable terminals, whereas PassGPT does not move at all, since PCFG ranks reusable fragments while PassGPT must assign next-token probability across a whole sequence. Chinese is near-null for the three generative families and declines 1.843 pp for PCFG at 10^8 (Section 7). The negative PassGAN and weak PassRVAE cells rule out a representation-only story: a cleaner linguistic form does not help a generator that cannot place, or sample, enough mass on the held-out identities.

Countermeasures. Our results suggest design implications rather than evaluated defences. Where native structure hides in an English-looking surface a language-blind model under-guesses such passwords, most clearly for Korean, so a strength estimate built on one could overstate their resistance; a defender auditing a multilingual population should score candidates in both representations, and blocklists and strength meters can be extended with reconstructed forms, cheap to derive from the rules above. The effect is model-conditional and degraded PassGAN in four of six cells, so any deployment needs language detection and per-model selection, and multilingual benchmarks should report both conditions.

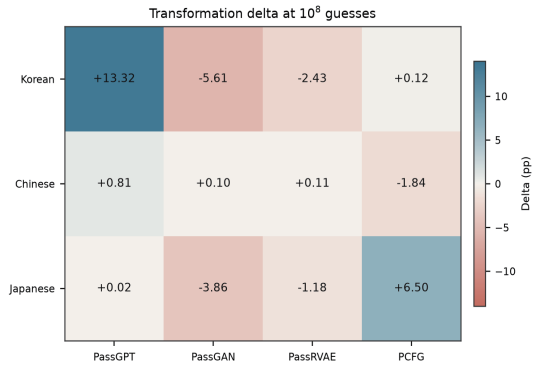


Fig. 4. Transformation delta at 10^8 guesses. Teal indicates improvement; terracotta indicates degradation.

All gains here are finite-budget, so salting and rate limiting remain the controls that decide whether a budget advantage matters. None of these defences is evaluated here.

7 Limitations

Uncertainty has two layers and only one is measured. Evaluation sampling is small: treating each held-out set as a sample, the 95% Wilson interval on every cell of Table 4 has a half-width of at most 0.08 pp, so Korean PassGPT (+13.324 pp), Japanese PCFG (+6.500 pp) and the PassGAN losses lie far outside that band while the Japanese PassGPT difference of 0.019 pp does not. Training-seed variance is not measured at all, since each cell rests on one training run; run-to-run variability cannot be estimated from our data and the true uncertainty of every cell exceeds the interval just quoted by an unknown amount. We therefore read the sub-one-point cells as magnitude-level indications rather than precise effects.

A further limitation concerns the PCFG rows. For each language the transformed PCFG row in Table 4 uses the terminalization with the highest total hit-rate on this same split: bigram chunks for Chinese, trigram chunks for Japanese, and the whole Hangul run for Korean. Those rows are therefore best-configuration results rather than one pre-registered grammar, so the Japanese PCFG gain should be read as an upper estimate. For Korean the selected setting is the whole Hangul run, which is that language's best total hit-rate of the four we tried; we keep it for that reason, while noting that a whole-run terminal fires only when the same run recurs between training and test and so generalizes less than a chunked terminal. Two corrections bear on the Korean PassGPT delta of +13.324 pp at 10^8, and both are small. Removing the residual hash records of Section 3.3 from the original test set moves it to +13.242 pp. Separately, the two conditions were scored against 1,247,980 and 1,239,533 entries, and rescaling the original

onto the smaller denominator moves it to +13.239 pp. Neither correction reaches 0.09 pp, under one percent of the effect it qualifies.

We report no modelability result. A token-level likelihood comparison across the two conditions would not be informative here, because the per-condition PassGPT models were trained with different tokenizers, over a 94-symbol alphabet in the original condition and a 160-symbol alphabet in the transformed one. Negative log-likelihood per token is a per-symbol code length, so under those two tokenizers it measures partly the size of the alphabet rather than the fit of the representation: the gap between the two uniform-alphabet ceilings alone is larger than any effect we could have attributed to the transformation. Establishing modelability would require retraining both conditions under one shared tokenizer, which we leave to future work.

External validity is limited. The splits are in-distribution, so cross-corpus evaluation would be a stronger test of whether the reconstruction captures general population behavior; keyboard layout, IME convention, regional spelling and service context may all shift the result. We also compare the four families only to one another, not to an external baseline such as an ordered Markov enumerator, so absolute-performance context is missing.

8 Ethical Considerations

All password data was already public before this study: documented breach compilations, including the standard 000webhost benchmark, together with publicly circulating combination lists. We deliberately do not reproduce the distribution-channel filenames of the latter, since those strings point to where stolen credentials are traded. Only the strings are used, all runs are local, we redistribute nothing, and the transformed representations derive only from already-public strings, so the analysis does not extend the exposure caused by the original leaks. The techniques could in principle aid an attacker, but the same analysis helps defenders account for hidden input-method structure a language-blind model may miss; because no new credentials are recovered, we judge the defensive value to outweigh the marginal risk.

9 Conclusion

Our primary contribution is a deterministic native-language reconstruction, applied as representation-level preprocessing and evaluated across four generator families. Its value is conditional: PassGAN is usually harmed and Japanese PassGPT gains nothing measurable, so transformation is not a universal preprocessing step. The largest positive effect is Korean PassGPT, whose finite-budget hit-rate rises by 13.324 pp at 10^8; Japanese PCFG gains under a chosen hiragana terminalization, while PassRVAE moves under a point at 10^7 and reverses only for Korean at 10^8.

The central lesson is methodological. A representation can be more linguistic without being more useful to every generator: the reconstruction that doubles

Korean PassGPT recovery costs PassGAN several points and leaves Japanese PassGPT where it was. The correct claim is not that native transformation is superior, but that its effect is conditional on model and language. As a working hypothesis, not a measured result, we read this as a probability-mass alignment effect; measuring that mechanism is left to future work.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00407295).

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

References

1. Weir, M., Aggarwal, S., de Medeiros, B., and Glodek, B. Password cracking using probabilistic context-free grammars. IEEE S&P, 2009.
2. Hitaj, B., Gasti, P., Ateniese, G., and Perez-Cruz, F. PassGAN: A deep learning approach for password guessing. ACNS, 2019.
3. Rando, J., Perez-Cruz, F., and Hitaj, B. PassGPT: Password modeling and (guided) generation with large language models. ESORICS, 2023.
4. Pasquini, D., Gangwal, A., Ateniese, G., Bernaschi, M., and Conti, M. Improving password guessing via representation learning. IEEE S&P, 2021.
5. Li, Z., Han, W., and Xu, W. A large-scale empirical analysis of Chinese web passwords. USENIX Security, 2014.
6. Wang, Y., Qiu, W., Tang, P., Tian, H., and Li, S. SE#PCFG: Semantically enhanced PCFG for password analysis and cracking. IEEE Trans. Dependable Secure Comput., 22(4), 2025.
7. Pasquini, D., Ateniese, G., and Troncoso, C. Universal neural-cracking-machines: Self-configurable password models from auxiliary data. IEEE S&P, 2024.
8. Xiao, Y. PassRVAE: Improved trawling attacks via recurrent variational autoencoder. CNSCT, 2024.
9. Yan, S.-H., and Ku, C. C.-Y. Using language-specific input methods and pronunciation rules to improve the guesses of passwords. J. Inf. Secur. Appl., 77, 103588, 2023.
10. Xia, Z., Yi, P., Liu, Y., Jiang, B., Wang, W., and Zhu, T. GENPass: A multi-source deep learning model for password guessing. IEEE Trans. Multimedia, 22(5), 2020.
11. Zhang, Y., Xian, H., and Yu, A. CSNN: Password guessing method based on Chinese syllables and neural network. Peer-to-Peer Netw. Appl., 13, 2020.
12. Corrias, W., De Gaspari, F., Hitaj, D., and Mancini, L. V. MAYA: A unified benchmark for generative password guessing. arXiv:2504.16651, 2025.
13. Zou, Y., An, M., and Wang, D. Password guessing using large language models. USENIX Security, 2025.
14. Choi, D.-H., Kim, Y., Son, K.-W., and Lee, C.-H. SeekPass: A DeepSeek-based password dictionary generator. J. Korean Inst. Inf. Secur. Cryptol., 36(1), 201–215, 2026.
15. Cha, Y., et al. PassDNVAE: A lightweight password dictionary generation model using a DenseNet-based VAE. IEEE Internet Things J., 2026.

Optimized Implementation of Warp-Cooperative GPU HCTR2-ARIA Wide-Block Encryption

Siwoo Eum[1]*, Minho Song[1], Seung-Won Lee[1], Hagyeong Kim[1], and Hwajeong Seo[1]

Hansung University, Seoul, South Korea
{shuraatum, smino0906, dkajdfhd1, kimhaha4420, hwajeong84}@gmail.com

Abstract. HCTR2 is a *wide-block* encryption mode that encrypts one fixed-size message as a single unit, so that flipping a single plaintext bit re-randomizes the whole ciphertext. Its main use is disk encryption, where the message is a disk sector. We instantiate it with ARIA, the Korean national block-cipher standard, and implement it on an NVIDIA RTX 4080 GPU. With many independent messages, assigning one thread per message keeps the device occupied. At low queue depth, however, most of the GPU sits idle, and encrypting one 16 KiB message is several times *slower* than a single CPU core. The reason is that one HCTR2 message is internally sequential, so it cannot simply be split across threads. We present a *fine-grained* kernel in which one 32-thread warp cooperates on a single message. The kernel exploits the linearity of POLYVAL. It splits the long hash computation into contiguous per-thread pieces, has each thread compute a partial result, and combines the partials with warp shuffles to obtain exactly the same value as the sequential computation. This brings the latency of one message down to *parity* with a single CPU core (ARIA-256: 222 μs on the GPU vs. 209 μs on the CPU). Moreover, the number of threads per message and of messages per block acts as a knob that trades latency for throughput, spanning 222 μs/30 GB/s to 1774 μs/49 GB/s. With as few as four messages in flight the GPU already beats the CPU. We validate correctness for all three key sizes.

Keywords: HCTR2 · ARIA · Wide-block encryption · Tweakable cipher · GPU · CUDA · Bank conflict · POLYVAL · Storage encryption.

1 Introduction

Disk and file encryption must be *length-preserving*. A storage device keeps data in fixed-size sectors and has no room to make the ciphertext larger for a nonce or an authentication tag. Each such fixed-size unit, which we call a *message*, should also behave as one wide block, meaning that changing any single plaintext bit re-randomizes the whole ciphertext. The de-facto standard XTS-AES [3] does not do this. It is *narrow-block*, encrypting each 16-byte piece on its own, so an

* Corresponding authors: Siwoo Eum (shuraatum@gmail.com) and Hwajeong Seo (hwajeong84@gmail.com).

attacker can reorder the pieces inside a message or confine a one-bit change to a single piece. *Wide-block* modes, also known as tweakable enciphering and recently as *accordion* [12] modes, instead realize a strong pseudorandom permutation over the whole message. HCTR2 [2] is a modern, efficient example. It has a Hash–Encrypt–Counter–Hash structure that uses one block-cipher call, a counter-mode keystream, and two passes of the POLYVAL [4] hash. It improves on the earlier HCTR [16], has a security proof, and ships in the Linux kernel.

We build HCTR2 on **ARIA** [6,7], the 128-bit block cipher that is a Korean national standard (KS X 1213, ISO/IEC 18033-3). ARIA is a substitution–permutation network with 12/14/16 rounds for 128/192/256-bit keys. This pairing gives a standards-compliant wide-block mode for settings that require a national cipher, yet no fast implementation of it has been reported.

Encrypting many messages is easy to parallelize because the messages are independent, which suits a GPU's high *throughput*. The hard case is low *latency* on only a few messages, such as a single random disk read or a queue with just a few messages in flight. The common GPU mapping, *one thread per message*, is ideal for the first case but the worst for the second. With only a few messages, a few threads out of thousands of GPU cores do all the work, and a single 16 KiB message ends up several times slower than one CPU core. Making it worse, one HCTR2 message is internally sequential. Two POLYVAL hash passes run in order, and the counter keystream between them depends on a value S produced by the first pass, so a message cannot simply be split across threads. Splitting it makes the threads communicate. This paper targets exactly this low-queue-depth *latency gap*.

At the implementation level, both halves of HCTR2-ARIA read *tables at positions that depend on the secret data*. ARIA does data-dependent S-box lookups, and a fast POLYVAL multiplies using a table indexed by message bytes. GPU memory handles this pattern badly. Constant memory is efficient for warp-uniform broadcasts but serializes divergent addresses. A single shared-memory copy of a table causes *bank conflicts* when threads in a warp read different entries that land in the same memory bank. Table placement is therefore a first-order design decision.

Related work. GPU symmetric cryptography has focused almost entirely on AES, with shared-memory T-table designs [10,5] and bitslicing [13], and later work covering more ciphers, including Korean ones [1,8]. It is a known principle that higher occupancy is not always faster [15]. Shared-memory bank conflicts have also been addressed for AES. Tezcan [14] kept T-tables in shared memory while removing conflicts, reaching 878.6 Gbps for AES-128 on an RTX 2070 Super, and showed that conflict removal is the key lever for GPU T-table kernels. We extend this bank-conflict-free technique to two different non-AES data-dependent tables (the ARIA S-box and the POLYVAL multiply table) as the foundation of our kernel. Our focus, however, is not throughput but closing the single-message *latency* gap by having a warp cooperate on one message. Ours is also, to our knowledge, the first GPU realization of HCTR2, and the first over a non-AES cipher.

Contributions.

- **A warp-cooperative fine-grained kernel.** A group of TPS threads cooperates on one message. Its core is a *parallel hash reduction* that uses a linearity property of POLYVAL. The length-b hash sequence is split into contiguous per-thread pieces whose partial sums d_t are recombined as $D = \bigoplus_t d_t\,(h^{\mathrm{CHUNK}})^{\mathsf{TPS}-1-t}$ with in-warp shuffles, which reproduces the sequential result exactly. The counter keystream is split the same way, so each thread hashes the block it just produced without re-reading it.
- **A latency–throughput design space with a fair CPU comparison.** Threads per message (TPS) and messages per block (SPB) trade latency for throughput (Section 5). Fine-graining lowers the latency of one 16 KiB message from several times slower than a well-tuned CPU core to *parity* with it (ARIA-256: 222 vs. 209 μs), and the GPU overtakes the CPU with just four messages in flight. The baseline is a single-core reference compiled with `-O3 -march=native`, chosen conservatively so it does not favor the GPU.
- **Bank-conflict-free shared tables (the foundation).** We replicate *both* the ARIA S-box table and the POLYVAL multiply table so that thread ℓ always reads bank ℓ, which drives shared-table conflicts to zero. We also use a half-size POLYVAL table built from two 4-bit sub-tables using the linearity of multiplication. Together these keep the kernel's data-dependent lookups cheap.

2 Background

ARIA. ARIA [6,7] is a 128-bit block cipher with 128/192/256-bit keys and $N_r = 12/14/16$ rounds. Each round XORs in a round key, applies a substitution layer, and then a diffusion (bit-mixing) layer, with no diffusion in the last round. Two substitution types alternate from round to round, built from two S-boxes S_1, S_2 and their inverses. The diffusion layer is a fixed 16×16 binary matrix A. The standard software trick packs the four S-box outputs for one input byte into a single 32-bit word, giving one combined table of 256 entries. Each thread then picks the correct S-box byte with a byte permutation, and the matrix A is applied with word XORs and byte shuffles inside the word rather than more table lookups. So the only data-dependent memory access in the whole cipher is a single 1 KB table.

HCTR2. A wide-block mode behaves like a tweakable random shuffle [9]. The ciphertext has the same length as the plaintext, yet is indistinguishable from a random permutation even when the attacker can choose ciphertexts. HCTR2 [2] achieves this with one block-cipher call E, a counter keystream, and two passes of a $\mathrm{GF}(2^{128})$ hash. From the key it derives a POLYVAL hash key h (from $\mathsf{E}(0)$) and a block $L = \mathsf{E}(\texttt{0x01} \,\|\, 0^{120})$. It splits a message into its first block M and remainder N under a tweak T (a per-message extra input, such as the message's location). Writing $\mathcal{H}_h(\cdot)$ for the POLYVAL hash over a shared tweak/length

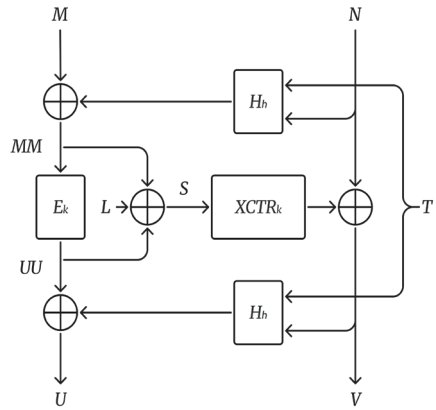


Fig. 1. HCTR2 encryption of one message.

prefix together with its argument, it computes

$$MM = M \oplus \mathcal{H}_h(N,T), \quad UU = \mathsf{E}(MM), \quad S = MM \oplus UU \oplus L, \tag{1}$$

$$V = N \oplus \mathrm{XCTR}_S, \quad U = UU \oplus \mathcal{H}_h(V,T), \quad C = U \,\|\, V, \tag{2}$$

where XCTR_S is the counter keystream whose block i equals $\mathsf{E}(S \oplus i)$. Two properties drive how we map this onto the GPU. First, the tweak and a length block form a *common prefix* of both hashes, so we can hash it once and then split the hash state. Second, for a message of b blocks the work is one ECB call (one block-cipher call), $b-1$ counter-mode calls, and $2(b-1)+2$ field multiplications. For our 16 KiB message this is 1 ECB, 1023 counter-mode calls, and about 2048 multiplications (Fig. 1).

POLYVAL and $\mathrm{GF}(2^{128})$. POLYVAL [4] is the hash used in AES-GCM-SIV, a close relative of GHASH [11]. With key h, the hash of blocks $X_1, \ldots, X_m$ is $\sum_j X_j\, h^{m-j+1}$, computed one block at a time as $\mathrm{acc} \leftarrow (\mathrm{acc} \oplus X_j) \cdot h$. Because h is fixed, the multiply by h can be turned into a per-key table. An 8-bit table $M_8[v] = v \cdot h$ replaces each multiply with about 16 lookups and XORs instead of about 128 bit-serial steps. How to place and replicate that table without bank conflicts is the subject of Section 4.

GPU memory and bank conflicts. A GPU runs threads in groups of 32, called *warps*, that execute together. *Shared memory* (fast on-chip memory) is split into 32 *banks*. The 32-bit word at byte address a lives in bank $\lfloor a/4 \rfloor \bmod 32$, and a warp can serve one word per bank at a time. If two threads want different words

in the same bank, the access is serialized. This is a *bank conflict*, and it is exactly what a data-dependent table lookup tends to cause. *Constant memory* serves a warp-uniform broadcast in a single access, but serializes k different addresses into k separate ones. Global-memory accesses *coalesce* when consecutive threads touch consecutive addresses, combining into one transaction. Finally, high *occupancy* (many warps resident at once) hides waiting time but is not always faster. A kernel limited by computation or by shared-memory speed can reach its peak at only modest occupancy [15].

3 GPU Design

Warp-cooperative fine-grained mapping. HCTR2 exposes abundant parallelism *across* messages. Each message is a self-contained unit with its own tweak, so with many messages *one thread per message* fills the GPU and gives high throughput. At low queue depth, however, only a few threads run, most cores sit idle, and the latency of one message is high. Reducing that latency requires parallelism *within* a message, but an HCTR2 message has two *sequential* POLYVAL hash chains and a counter keystream tied to a value S between them, so it cannot simply be split.

We therefore have a *group* of TPS threads (a full warp, or a power-of-two fraction of one so that TPS divides 32) cooperate on one message, in three phases. *(A) Parallel hash 1.* Because POLYVAL is linear, the length-b hash sequence (the tweak block followed by the N blocks) is split into contiguous pieces of $\mathsf{CHUNK} = b/\mathsf{TPS}$ blocks, one per thread. Thread t hashes its piece into a partial sum d_t, where only the group's first thread starts from the shared prefix state. A *warp shuffle* then recombines the partials via (3), giving *exactly* the sequential result. *(B) Single ECB and S broadcast.* The group leader forms MM, UU, S and shuffles S to the other threads. *(C) Parallel XCTR + fused hash 2.* The counter blocks $\mathsf{E}(S \oplus i)$ are independent, so with the same split each thread makes and writes its own V_i and folds V_i into a second partial sum while it is still in registers, with no re-read. The second partials are recombined as in (A) to produce U. The only communication between threads is these cheap shuffles, and nothing is staged in shared memory.

This mapping has two knobs. A larger TPS splits the counter and hash work more finely, which lowers the latency of one message. A smaller TPS gives each group more work and raises throughput (Section 5). We also pack SPB messages (groups) into one thread block so they share the block's shared tables, which are staged once. Round keys, L, h, and the hashed prefix sit in constant memory, where these accesses are warp-uniform. The three key sizes differ only in N_r, which we make a compile-time parameter so the round loop unrolls fully.

ARIA device core. The device core follows Section 2. A 256-entry combined table packs the four S-box outputs per byte. One round indexes this table and uses the hardware byte-permute instruction (`__byte_perm`) to select and place the substitution byte for the current round type, then applies the diffusion A

entirely in registers. The final round substitutes and XORs the last round key to produce ciphertext. The only data-dependent memory access is the 1 KB table lookup, which is exactly what makes aggressive replication affordable.

POLYVAL backends. We implement three interchangeable multiply backends behind one block-update routine, each selected at compile time so it can be measured on its own in Section 5. They are BITSERIAL, the table-free $\mathrm{GF}(2^{128})$ multiply (about 128 steps), TABLE, an 8-bit table in constant memory, and TABLE_SH, the replicated conflict-free shared table of Section 4, which is the optimized default. The POLYVAL byte-reversal is folded into the load and store steps.

Parallel reduction and the algorithm. The recombination in phase (A) is the identity

$$D = \bigoplus_{t=0}^{\mathsf{TPS}-1} d_t \cdot \left(h^{\mathsf{CHUNK}}\right)^{\mathsf{TPS}-1-t}, \qquad \mathsf{CHUNK} = b/\mathsf{TPS}, \tag{3}$$

which equals the *sequential* POLYVAL accumulator over the whole sequence. Here d_t is thread t's partial sum over its contiguous piece (only thread 0 starts from the shared prefix), and h^{CHUNK} shifts the degree by one piece. The leader does the recombination by multiplying by h^{CHUNK} and XOR-ing in the d_t it reads by shuffle ($\mathsf{TPS}-1$ steps). The identity holds whenever TPS divides b. Our 16 KiB message has $b = 1024$, which is divisible by every $\mathsf{TPS} \in \{2, 4, 8, 16, 32\}$. Algorithm 1 states the procedure from one thread's view, and each data block is read once and written once.

4 Bank-Conflict-Free Tables and the Design Space

For the fine-grained kernel to be cheap, its data-dependent table lookups must be fast. We first summarize the foundation, bank-conflict-free shared tables, which is a technique also used by throughput-oriented mappings. We then treat this kernel's own design space: threads per message and messages per block.

The problem. In both tables, each thread of a group indexes with a different, data-dependent byte (an ARIA state byte, or a POLYVAL message byte). Two natural placements fail. *Constant memory* broadcasts one address per access, so different indices are serialized through the constant cache. A *single shared copy* of a 32-bit-per-entry table puts entry i in bank i mod 32, so any threads whose indices are equal modulo 32 conflict and must replay. With random bytes this happens on nearly every access, which serializes the lookup.

Bank replication. The fix is to give every thread its own private copy of each table column, so that any access lands in *that thread's* bank. We declare the ARIA table `__shared__ u32 tS[256][RF]` with the *replication factor* RF (how

Algorithm 1 HCTR2-ARIA encryption of one message (thread t of a TPS-thread group)

Require: message blocks $M \parallel N_1 \parallel \cdots \parallel N_{b-1}$, tweak T, shared tables, constants h, L, prefix state acc_0, piece size $\mathsf{CHUNK} = b/\mathsf{TPS}$, sequence $X_0{=}T$ and $X_j{=}N_j$ for $j{\geq}1$

Phase A: parallel hash 1
$a \leftarrow \mathrm{acc}_0$ if $t{=}0$, else 0
for $j = t \cdot \mathsf{CHUNK}$ **to** $(t{+}1) \cdot \mathsf{CHUNK} - 1$ **do**
  $a \leftarrow \textsc{PolyUpdate}(a, X_j)$
end for
$dg_1 \leftarrow \textsc{Emit}(\textsc{Recombine}(a))$ ▷ by (3), via shuffle
Phase B: one block-cipher call (leader only)
if $t = 0$ **then**
  $MM \leftarrow M \oplus dg_1$
  $UU \leftarrow \textsc{Aria}_{N_r}(MM)$
  $S \leftarrow MM \oplus UU \oplus L$
end if
broadcast S from the leader by shuffle
Phase C: parallel counter mode and hash 2
$a \leftarrow \mathrm{acc}_0$ if $t{=}0$, else 0
for $j = t \cdot \mathsf{CHUNK}$ **to** $(t{+}1) \cdot \mathsf{CHUNK} - 1$ **do**
  if $j = 0$ **then**
    $X'_j \leftarrow T$ ▷ tweak, only hashed
  else
    $ks \leftarrow \textsc{Aria}_{N_r}(S \oplus j)$
    $V_j \leftarrow N_j \oplus ks$, then **store** V_j and set $X'_j \leftarrow V_j$
  end if
  $a \leftarrow \textsc{PolyUpdate}(a, X'_j)$
end for
$dg_2 \leftarrow \textsc{Emit}(\textsc{Recombine}(a))$
if $t = 0$ **then**
  store $U \leftarrow UU \oplus dg_2$
end if

many copies we keep) as the innermost dimension, and thread ℓ reads column $c = \ell \bmod \mathsf{RF}$. Element `tS[i][c]` sits at word $i \cdot \mathsf{RF} + c$, hence in bank

$$\big(i \cdot \mathsf{RF} + c\big) \bmod 32. \tag{4}$$

Setting $\mathsf{RF} = 32$ makes (4) equal $(32i + c) \bmod 32 = c = \ell$, so *every thread ℓ reads bank ℓ for every index i*. A warp then touches 32 different banks in one conflict-free access (Fig. 2). The price is 32× storage, so the 1 KB table becomes 32 KB. The POLYVAL table, declared `[nibble][word][RF]` with RF innermost, becomes conflict-free in the same way, since element $[n][w][c]$ is in bank $((4n + w) \cdot \mathsf{RF} + c) \bmod 32 = \ell$. Staging the tables (copying each row into all RF columns) is a one-time cost paid before the encrypt loop.

A half-size POLYVAL table from linearity. Replicating the 8-bit table M_8 (256× 16 bytes) by 32 would need 128 KB. We avoid this using the fact that multiplying

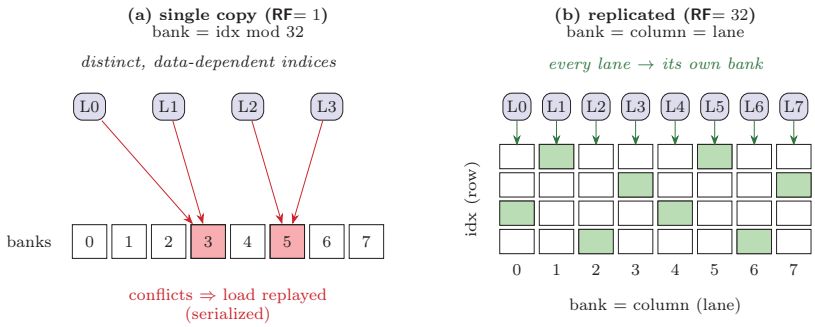


Fig. 2. Bank replication eliminates shared-memory conflicts (8 banks shown; the design uses 32).

by the fixed key h is linear (multiplying a sum equals summing the products). Since a byte splits as $v = (\mathrm{hi} \ll 4) \oplus \mathrm{lo}$, we can write

$$v \cdot h = M_4^a[\mathrm{hi}] \oplus M_4^b[\mathrm{lo}], \quad M_4^a[k] = (k \ll 4) \cdot h, \; M_4^b[k] = k \cdot h, \tag{5}$$

using two 16-entry tables, one for each 4-bit half of the byte. Replicated by $\mathsf{RF} = 32$ they take $2 \times 16 \times 4 \times 32 \times 4 = 16\,\mathrm{KB}$, which is half of a replicated 8-bit table and a quarter of the naive 128 KB. Together with the 32 KB ARIA table this comes to **48 KB of shared memory per block**, which is what sets occupancy.

Design space: threads per message TPS *and messages per block* SPB. Once the data-dependent lookups are cheap, performance is set by the two knobs of Section 3. First, *threads per message* TPS trades latency for throughput. A larger TPS splits a message's counter and hash work more finely and cuts the latency of one message, but each thread's piece gets shorter and the relative cost of the recombination (3) grows, which lowers per-group efficiency. A smaller TPS gives each group more work and raises throughput at the cost of latency. So TPS sweeps a frontier between latency and throughput.

Second, *messages per block* SPB sets the staging cost of the 48 KB shared tables and the occupancy. The 48 KB is a fixed per-block cost paid once, so packing SPB messages (groups) into a block spreads it over SPB messages and raises the number of resident warps. On the Ada SM two 48 KB blocks (96 KB) fit within the 100 KB capacity (with the shared-memory carveout, the split between L1 cache and shared memory, set to its maximum), so 2 blocks live on each SM, and raising SPB increases the warps those 2 blocks carry. At SPB = 1 (one thread block per message) the 48 KB is re-staged for every message, which is wasteful. Grouping a few messages per block removes that waste.

Every global load and store is one 128-bit `uint4` access through the read-only path (`__ldg`). Section 5 sweeps this design space (TPS, SPB) in both latency and throughput and compares against a well-tuned single CPU core.

Table 1. Latency of one 16 KiB message (μs, lower is better).

Approach	ARIA-128	ARIA-192	ARIA-256
One thread per message (naive)	—	—	∼2310
CPU one core (`-O3 -march=native`)	185	203	209
GPU fine-grained (TPS=32)	**207**	**217**	**222**
Ratio (CPU/GPU time)	0.90×	0.94×	0.94×

5 Evaluation

Setup. We measure on an NVIDIA RTX 4080 (AD103, Ada, `sm_89`, 76 SMs, ≈716.8 GB/s DRAM peak), CUDA 12.0, driver 535.309.01, gcc 13.3, Ubuntu 24.04, with GPU flags `-arch=sm_89 -O3`. Shared tables use $\mathsf{RF} = 32$ (conflict-free) with the maximum shared-memory carveout. We report two metrics. *Latency* is the time to encrypt a single 16 KiB message, in μs per message. For the GPU we run the kernel many times in a row and divide the device time by the number of runs (this is within 1% of the wall-clock time including launch overhead), and we take the minimum over 300 repetitions. *Throughput* is plaintext bytes over device time for 65536 messages (1 GiB), averaged over 20 iterations. The *CPU baseline*, chosen to be *fair*, is a single core running the reference HCTR2-ARIA compiled with `-O3 -march=native -funroll-loops` (T-table ARIA and table-assisted POLYVAL), timed by wall clock. It is not AES-NI-class, because ARIA has no x86 hardware instruction, and a hand-vectorized CPU (CLMUL POLYVAL, AVX2 ARIA) would be faster. In other words this baseline is conservative and does not favor the GPU.

Correctness. All checks pass. We run device-core ARIA known-answer tests for all three key sizes, check the GF(2^{128}) multiply (bit-serial and table) over 2^{20} inputs, and confirm that the HCTR2 output is byte-identical to an independent reference. The fine-grained kernel is checked for *every* $\mathsf{TPS} \in \{2, 4, 8, 16, 32\}$, several values of SPB, and all three key sizes, including a message count that is not a multiple of SPB. The CPU baseline output also matches the reference.

Single-message latency: parity with a CPU core. Table 1 is the headline. The common *one thread per message* mapping takes about 2310 μs for a single 16 KiB message at ARIA-256, which is 11× slower than a CPU core (209 μs), because the GPU sits nearly idle. The warp-cooperative fine-grained kernel (TPS=32) cuts this to 222 μs, reaching *parity* with one CPU core (it uses 0.90–0.94× the CPU time across the three key sizes). It thus erases the single-message latency gap, from an order of magnitude down to parity.

Design space: TPS trades latency for throughput. Table 2 and Fig. 3 sweep TPS at a fixed 256-thread block ($\mathsf{SPB} = 256/\mathsf{TPS}$) for ARIA-256. Latency at queue depth 1 is measured on one message, and throughput on 65536 messages. Increasing TPS steadily drops the latency of one message from 1774 to 222 μs.

Table 2. The TPS design space (ARIA-256).

TPS	SPB	latency (μs)	throughput (GB/s)
2	128	1774	**49.0**
4	64	879	50.5
8	32	469	50.6
16	16	283	43.3
32	8	**222**	29.7

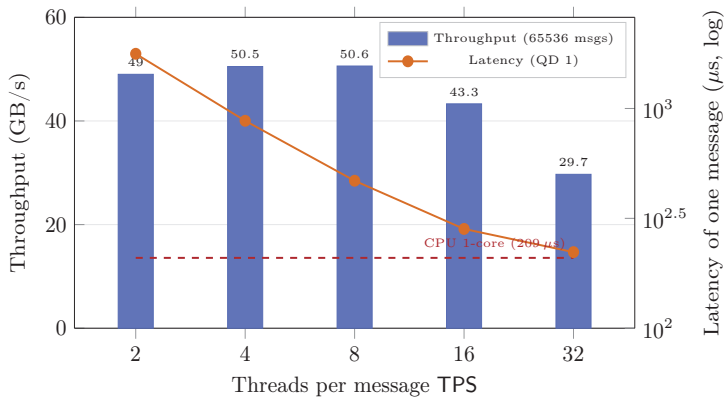


Fig. 3. The latency–throughput tradeoff as threads per message (TPS) varies (ARIA-256).

Throughput is highest for TPS$\leq$8 (about 50 GB/s, and at TPS=2 it reaches 49 GB/s, close to a throughput-oriented mapping) and falls as TPS grows, because of lower per-group efficiency and recombination cost. So TPS is a knob along a frontier between latency and throughput. TPS=32 is best for latency (222 μs), TPS$\leq$8 is best for throughput (about 50 GB/s), and TPS=16 is the balanced middle (283 μs, 43 GB/s).

Messages per block: sharing the table-loading cost. Table 3 increases SPB at TPS=32. With SPB=1 (one thread block per message) the 48 KB shared tables are re-loaded for every message, which caps throughput at 10.8 GB/s with only 2 resident warps. Grouping messages spreads the loading cost and raises occupancy, reaching 30.0 GB/s at SPB=8 (a 2.8$\times$ gain, 2 to 16 warps per SM). Beyond that, at SPB=16, the gain saturates.

Latency vs concurrency: the crossover. So far we fixed a single message (queue depth 1). Fig. 4 varies the number of messages processed at once and measures the time per message (TPS=32, SPB=8). The single CPU core's per-message time is flat at 209 μs, because it processes messages one after another. The GPU's

Table 3. Throughput vs messages per block SPB (ARIA-256, TPS=32).

SPB	block threads	warps/SM	GB/s
1	32	2	10.8
2	64	4	21.2
4	128	8	28.2
8	256	16	**30.0**
16	512	16	28.7

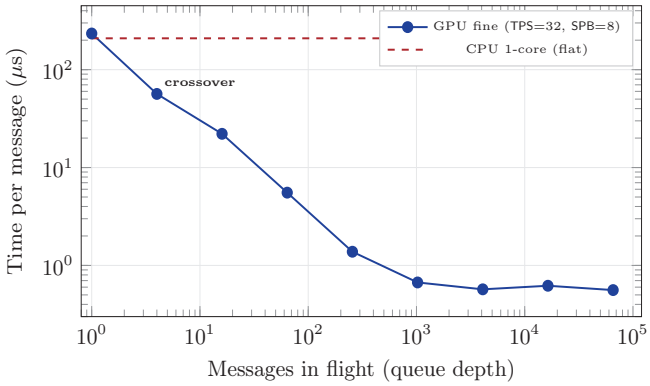


Fig. 4. Time per message as more messages run at once (ARIA-256, log–log). The GPU overtakes the CPU between 1 and 4 messages in flight.

per-message time falls steeply as the device fills up. At queue depth 1 the GPU is 234 μs, close to the CPU, but with just *four* messages in flight it is 56 μs, which is 3.7× faster than the CPU. It reaches 5.5 μs at 64 messages (38×) and saturates near 0.56 μs per message (about 30 GB/s). The crossover lies between queue depths 1 and 4. The value of fine-graining is exactly in this low-queue-depth regime. It matches or beats the CPU on a single message, and slides straight into GPU-class throughput as soon as a little concurrency appears.

Discussion. Fine-graining brings the latency of one message from 11× slower than a naive mapping to parity with one CPU core (222 μs), and it overtakes the CPU with as few as four messages in flight. TPS and SPB trade latency for throughput, so the same kernel spans 222 μs at 30 GB/s (latency-oriented) to about 880 μs at 50 GB/s (throughput-oriented). At queue depth 1 the GPU is at parity rather than a clear *win* over the CPU, and the main gain appears once concurrency is available. A faster CPU version (CLMUL POLYVAL, AVX2 ARIA) would put the queue-depth-1 CPU further ahead, which only reinforces the crossover-focused conclusion. We evaluated encryption on a single GPU. The design supports all three key sizes, and decryption is symmetric.

6 Conclusion

We presented a GPU implementation of HCTR2 wide-block encryption over the Korean standard ARIA, to our knowledge the first, focused on the *latency of one message at low queue depth*. The common one-thread-per-message mapping is ideal for bulk throughput, but with only a few messages it leaves the GPU idle and runs several times slower than a single CPU core. We close that gap with a fine-grained kernel in which a group of threads *cooperate* on one message. Its core is a parallel hash reduction (3) that uses the linearity of POLYVAL. Each thread hashes a piece into a partial sum, and a warp shuffle combines the partials to reproduce the sequential result exactly, with the counter keystream split the same way so each thread hashes the block it just produced. To make these data-dependent lookups cheap, we replicate *both* the ARIA S-box and POLYVAL multiply tables so that every thread reads its own bank ((4) guarantees exactly zero conflicts), and we shrink the replicated POLYVAL table with a 4-bit split.

The result is that fine-graining brings the latency of one 16 KiB message from 11× slower than a naive mapping to *parity* with a well-tuned CPU core (ARIA-256: 222 vs. 209 μs). Threads per message (TPS) and messages per block (SPB) then trade latency for throughput, so one kernel spans a range from latency-oriented (222 μs) to throughput-oriented (about 50 GB/s), and the GPU overtakes the CPU with as few as four messages in flight. At queue depth 1 the result is parity rather than a clear win against the CPU baseline; the larger gain comes from the crossover once concurrency appears. This approach should carry over to other wide-block modes and to national or lightweight ciphers built around a similar POLYVAL-style sequential hash. A faster CPU baseline (CLMUL/AVX), an overlap pipeline, and multi-GPU scaling are natural next steps.

7 Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2025-02306395, Development and Demonstration of PQC-Based Joint Certificate PKI Technology, 40%) and this work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25394739, Development of Security Enhancement Technology for Industrial Control Systems Based on S/HBOM Supply Chain Protection, 30%) and this work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25519543, Development of PQC Migration Technology for Quantum-Safe IC Chip, 30%).

References

1. An, S., Seo, S.C.: Highly efficient implementation of block ciphers on graphic processing units for massively large data. Applied Sciences **10**(11), 3711 (2020). https://doi.org/10.3390/app10113711
2. Crowley, P., Huckleberry, N., Biggers, E.: Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Paper 2021/1441 (2021), https://eprint.iacr.org/2021/1441
3. Dworkin, M.: Recommendation for block cipher modes of operation: The XTS-AES mode for confidentiality on storage devices. NIST Special Publication 800-38E (2010). https://doi.org/10.6028/NIST.SP.800-38E
4. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Nonce misuse-resistant authenticated encryption. RFC 8452, Internet Engineering Task Force (IETF) (2019). https://doi.org/10.17487/RFC8452
5. Iwai, K., Nishikawa, N., Kurokawa, T.: Acceleration of AES encryption on CUDA GPU. International Journal of Networking and Computing **2**(1), 131–145 (2012). https://doi.org/10.15803/ijnc.2.1_131
6. Kwon, D., Kim, J., Park, S., Sung, S.H., Sohn, Y., Song, J.H., Yeom, Y., Yoon, E.J., Lee, S., Lee, J., Chee, S., Han, D., Hong, J.: New block cipher: ARIA. In: Information Security and Cryptology – ICISC 2003. Lecture Notes in Computer Science, vol. 2971, pp. 432–445. Springer (2004). https://doi.org/10.1007/978-3-540-24691-6_32
7. Lee, J.H., Lee, J., Kim, J., Kwon, D., Kim, C.: A description of the ARIA encryption algorithm. RFC 5794, Internet Engineering Task Force (IETF) (2010). https://doi.org/10.17487/RFC5794
8. Lee, W.K., Goi, B.M., Phan, R.C.W.: Terabit encryption in a second: Performance evaluation of block ciphers in GPU with Kepler, Maxwell, and Pascal architectures. Concurrency and Computation: Practice and Experience **31**(11), e5048 (2019). https://doi.org/10.1002/cpe.5048
9. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Advances in Cryptology – CRYPTO 2002. Lecture Notes in Computer Science, vol. 2442, pp. 31–46. Springer (2002). https://doi.org/10.1007/3-540-45708-9_3
10. Manavski, S.A.: CUDA compatible GPU as an efficient hardware accelerator for AES cryptography. In: IEEE International Conference on Signal Processing and Communications (ICSPC). pp. 65–68. IEEE (2007). https://doi.org/10.1109/ICSPC.2007.4728256
11. McGrew, D.A., Viega, J.: The security and performance of the Galois/Counter Mode (GCM) of operation. In: Progress in Cryptology – INDOCRYPT 2004. Lecture Notes in Computer Science, vol. 3348, pp. 343–355. Springer (2004). https://doi.org/10.1007/978-3-540-30556-9_27
12. National Institute of Standards and Technology: Proposal of requirements for an accordion mode: Discussion draft. NIST, Workshop on Block Cipher Modes of Operation (2024), https://csrc.nist.gov/projects/block-cipher-techniques
13. Nishikawa, N., Amano, H., Iwai, K.: Implementation of bitsliced AES encryption on CUDA-enabled GPU. In: Network and System Security (NSS 2017). Lecture Notes in Computer Science, vol. 10394, pp. 273–287. Springer (2017). https://doi.org/10.1007/978-3-319-64701-2_20
14. Tezcan, C.: Optimization of Advanced Encryption Standard on graphics processing units. IEEE Access **9**, 67315–67326 (2021). https://doi.org/10.1109/ACCESS.2021.3077551

15. Volkov, V.: Better performance at lower occupancy. In: GPU Technology Conference (GTC) (2010)
16. Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: Information Security and Cryptology (CISC 2005). Lecture Notes in Computer Science, vol. 3822, pp. 175–188. Springer (2005). https://doi.org/10.1007/11599548_15

Accelerating the AIMer Post-Quantum Signature with AVX-512: A Field–Keccak Speedup Analysis

Seung-Won Lee[1[0009−0009−9583−1387]], Min-Ho Song[1[0009−0007−6277−0069]], Si-Woo Eum[1[0000−0002−9583−5427]], Ha-Gyeong Kim[1[0009−0008−5840−5517]], Min-Seo Kim[1[0009−0008−1979−5778]], and Hwa-Jeong Seo[1[0000−0003−0069−9061]]

Department of Information Computer Engineering,
Hansung University, Seoul, 02876 Korea
dkajdfhd1@gmail.com, smino0906@gmail.com, shuraatum@gmail.com, kimhaha4420@gmail.com, msqoo0212@gmail.com, hwajeong84@gmail.com

Abstract. AIMer is a post-quantum digital signature scheme with a conservative design. Its security relies only on the symmetric-key one-way function AIM2 and an MPC-in-the-Head (MPCitH) zero-knowledge proof. AIMer was selected as a final algorithm in the Korean Post-Quantum Cryptography (KpqC) competition. However, its final reference implementation, released in January 2026, is written in portable C. It does not include processor-specific optimizations. As a result, it does not exploit AVX-512, a 512-bit vector instruction set available on modern processors. This paper presents an AVX-512 optimized implementation for all six AIMer variants. We parallelize binary-field multiplication with VPCLMULQDQ. The implementation processes four MPC parties per 512-bit register. We also handle conditional XOR accumulation in the linear layer with a single VPTERNLOGQ instruction. In addition, we accelerate the Keccak permutation using AVX-512VL. Our implementation is 1.60–1.84× faster than the AVX2 implementation for signing. We also analyze where the speedup over AVX2 comes from. The AVX2 baseline already vectorizes binary-field arithmetic. In the 128-bit and 192-bit variants, Keccak accounts for approximately 86% of the additional improvement obtained with AVX-512. In the 256-bit variants, the contribution of field arithmetic increases to approximately 42%. These ratios depend on the selected baseline and do not represent an absolute execution-time breakdown relative to a scalar reference implementation. They show that Keccak remains an important optimization target when moving from AVX2 to AVX-512.

Keywords: Post-Quantum Cryptography · AIMer · MPC-in-the-Head · AVX-512 · VPCLMULQDQ · Binary-Field Arithmetic · Keccak · Benchmarking Methodology

1 Introduction

As the threat posed by quantum computers becomes increasingly realistic, the migration from conventional public-key cryptography to post-quantum cryptography (PQC) is rapidly progressing. For this transition

to be deployed in real-world systems, not only security but also implementation efficiency is required.

AIMer is a digital signature scheme selected as a final algorithm in the Korean Post-Quantum Cryptography (KpqC) competition. Many PQC signature schemes rely on mathematical structures such as lattices or multivariate quadratic equations. In contrast, AIMer derives its security solely from a symmetric-key one-way function and an MPC-in-the-Head (MPCitH) zero-knowledge proof [17]. The primary challenge of such a symmetric-primitive-based design is its computational cost. However, the computation is inherently parallel. In particular, the signer repeatedly performs *the same binary-field arithmetic* for a large number of virtual MPC parties. The entire process is tied together by *a large number of Keccak permutations*. Both components are ideal targets for wide vector instructions (SIMD). This observation motivates the work presented in this paper.

Despite this opportunity, the AIMer implementation released in January 2026 provides only a portable C reference implementation [10]. KpqClean, which collects implementations of KpqC candidate algorithms, includes AVX2-based optimizations. However, it does not yet exploit AVX-512, a 512-bit SIMD instruction set available on modern processors. This instruction set includes VPCLMULQDQ for binary polynomial multiplication and VPTERNLOGQ for three-input bitwise logic. In this work, we integrate both the reference and AVX2 implementations into the liboqs (Open Quantum Safe) framework. Liboqs is an open-source library that provides a common interface for a wide range of post-quantum cryptographic algorithms. We use these implementations as baselines and extend them with AVX-512 optimizations.

To address the issues outlined above, this paper makes the following contributions:

1. We present the first comprehensive AVX-512 implementation covering all six AIMer variants. The implementation includes a VPCLMULQDQ-based four-party field layer, a VPTERNLOGQ-based linear layer, and an AVX-512VL-based Keccak implementation. Compared with the AVX2 implementation, it achieves a speedup of 1.60–1.84×.
2. We analyze the additional speedup obtained when moving from AVX2 to AVX-512. The AVX2 baseline already vectorizes binary-field arithmetic. In the 128-bit and 192-bit variants, Keccak accounts for approximately 86% of this additional improvement. These ratios depend on the selected baseline.
3. We analyze the variation in AVX-512 efficiency across AIMer variants from the perspectives of field width and the number of MPC parties. Elements in $\mathrm{GF}(2^{128})$ and $\mathrm{GF}(2^{256})$ can be packed efficiently into 512-bit registers, resulting in higher gains. In contrast, $\mathrm{GF}(2^{192})$ incurs losses due to zero padding. Furthermore, the small variants benefit more from four-party packing because the packing operation is repeated more frequently as the number of parties increases.

The remainder of this paper is organized as follows. Section 2 reviews the background of the AIMer signature scheme, the AIM2 one-way function, and the MPCitH proof system. Section 3 surveys prior work on MPCitH

signatures and SIMD-based cryptographic optimization. Section 4 describes our AVX-512 implementation, including VPCLMULQDQ-based four-party field arithmetic, a VPTERNLOGQ-based linear layer, and AVX-512VL-accelerated Keccak. Section 5 quantitatively evaluates performance relative to AVX2 and analyzes the observed improvement by decomposing it into field-arithmetic and Keccak contributions. Finally, Section 6 concludes the paper and discusses limitations and future research directions.

2 Background

AIMer is a post-quantum digital signature scheme. Its security relies on a symmetric-key one-way function. This one-way function is called AIM2 and is defined over the binary field $\mathrm{GF}(2^\lambda)$. A one-way function is easy to evaluate, but it is difficult to recover its input from its output. In this sense, it is similar to a hash function.
The secret key is a randomly chosen input value. The public key is the output obtained by applying AIM2 to this input. A signature proves that the signer knows the input value that generated the public key. At the same time, it does not reveal the secret input itself. This type of proof, which proves only the fact of knowledge without revealing the secret, is called a **zero-knowledge proof**. AIMer generates signatures using a zero-knowledge proof [13].
AIMer constructs this proof using the **MPC-in-the-Head (MPCitH)** technique [23]. It simulates a setting in which N virtual parties hold shares of the secret key and jointly compute AIM2. The verifier then checks this simulated computation. A single execution still leaves room for cheating. Therefore, the same process is repeated several dozen times, denoted by τ in Table 1.
Two types of operations dominate the signing cost. First, the N parties repeatedly perform *the same binary-field arithmetic*. Second, *Keccak hashing* [6,7] is used extensively throughout the process for commitments and challenge generation. This paper accelerates these two components using VPCLMULQDQ and AVX-512VL (Vector Length) [1], respectively. The parameters of each variant are shown in Table 1.

Table 1. Parameters of AIMer variants. λ: security bits/field size, N: number of MPC parties, τ: number of parallel repetitions.

Variant	Security level	Field $\mathrm{GF}(2^\lambda)$	N	τ	Signature (bytes)
AIMER-128f	NIST 1	$\mathrm{GF}(2^{128})$	16	33	5,888
AIMER-128s	NIST 1	$\mathrm{GF}(2^{128})$	256	17	4,160
AIMER-192f	NIST 3	$\mathrm{GF}(2^{192})$	16	49	13,056
AIMER-192s	NIST 3	$\mathrm{GF}(2^{192})$	256	25	9,120
AIMER-256f	NIST 5	$\mathrm{GF}(2^{256})$	16	65	25,120
AIMER-256s	NIST 5	$\mathrm{GF}(2^{256})$	256	33	17,056

3 Related Work

Prior work related to this paper can be divided into two lines. The first line is *MPCitH-based signatures*, to which AIMer belongs. This line shows how signatures have been made smaller and faster by choosing suitable symmetric primitives and checking techniques [18]. The second line is SIMD-based cryptographic optimization, which this paper applies to AIMer. This line focuses on accelerating cryptographic operations using vector instructions. We review these two lines in order.

3.1 MPCitH-Based Signatures

MPC-in-the-Head (MPCitH) is a zero-knowledge proof paradigm proposed by Ishai et al. [14]. In this paradigm, the proof size is proportional to the number of *multiplications*, or nonlinear gates, in the circuit. Subsequent studies have reduced this cost in several directions.
First, Picnic adopted LowMC as a one-way function. LowMC is a block cipher with a small number of AND gates, which reduces the number of multiplications itself [8]. Second, later work introduced techniques that trade communication cost and signature size by using a preprocessing model and adjusting the number of parties N. These techniques reduced signatures to the range of several tens of kilobytes [16]. Third, Banquet used the standard block cipher AES instead of LowMC. Instead of verifying inverse-S-box multiplications one by one, it grouped them into polynomial identities and checked them at once using an amortized check. This reduced the cost of multiplication verification [4]. BN++ further simplified this checking procedure [15]. This last approach was adopted by AIMer.
Furthermore, the FAEST digital signature scheme uses VOLE-in-the-Head [3] instead of MPCitH. It efficiently generates correlated randomness and reduces AES-based signatures to about 5 KB. FAEST advanced to the second round of the NIST additional signature competition [2]. In this line of work, AIMer is a case that introduces a *dedicated* one-way function, AIM. AIM obtains short signatures by minimizing the number of multiplications in the circuit with a low-multiplication S-box. After AIM was published, it was subjected to algebraic analysis [21]. AIM2, a strengthened version reflecting this analysis, was subsequently adopted in AIMer, which was selected as a final algorithm in the KpqC competition [10]. Another line that obtains security only from symmetric primitives is the hash-based signature family SPHINCS+/SLH-DSA [5,22].

3.2 SIMD-Based Cryptographic Optimization

SIMD acceleration has been actively studied for compute-intensive workloads such as sorting [9]. In cryptographic software, Drucker et al. used the fact that 512-bit VPCLMULQDQ performs four independent 64×64 carry-less multiplications in a single instruction. With this property, they substantially improved the throughput of binary-field multiplication and

GHASH compared with PCLMULQDQ [11]. The same authors also proposed VAES. VAES vectorizes AES rounds by packing four blocks into one register. This accelerates parallel modes such as AES-GCM [12].
SIMD optimization has also been studied for lattice-based PQC. For example, AVX-512 vectorization of Dilithium has been reported [25]. For Korean standard cryptographic algorithms, AVX-512 acceleration has been reported for the LSH hash function and LEA-128-GCM. LEA-GCM combines 16-block parallel CTR mode with four-block VPCLMULQDQ-based GHASH. It achieved about a 3.3× improvement over the KISA SSE reference implementation. LSH achieved an average speedup of 2.28× using dual-message processing based on 512-bit register interleaving [20].
Prior implementation studies have also examined AIM and AIM2 directly. Lee et al. accelerated the original AIM symmetric primitive by optimizing the Mer operation and simplifying its linear layer [18]. Song et al. designed quantum circuits for all AIM2 variants and analyzed their resource requirements [24]. These studies respectively address software optimization of the underlying AIM primitive and quantum resource analysis, but neither presents an AVX-512 implementation covering all six AIMer variants. A recent SIMD study in a related proof-system setting accelerated FAEST on ARM using NEON AES instructions and parallel VOLE generation [19]. In contrast, our work provides an end-to-end AVX-512 implementation of all six AIMer variants, jointly optimizing binary-field arithmetic, the linear layer, and Keccak.

4 Implementation

As discussed in Section 2, the cost of AIMer is concentrated in binary-field arithmetic repeated across all parties and in a large number of Keccak calls. This section reduces these costs using AVX-512. The basic idea consists of two parts. The first is to place the data of multiple parties side by side in a 512-bit register and process them simultaneously with a single instruction. The second is to merge several bit operations into a single instruction.
The existing AVX2 implementation, which we use as the baseline, stores the element of only *one party* in a 128-bit register. It then processes one party at a time using PCLMULQDQ. This limitation comes from the AVX2 PCLMULQDQ instruction, whose VEX-encoded form operates on 128-bit XMM operands. In this section, we explain how this limitation is overcome with 512-bit registers. We describe binary-field multiplication in Section 4.1, the linear layer in Section 4.2, and Keccak in Section 4.3.

4.1 Binary-Field Arithmetic: Four-Party Parallel Multiplication per 512-Bit Register

An element of $\mathrm{GF}\left(2^{128}\right)$ is 128 bits long. Thus, it fits exactly into one 128-bit *lane*. A 512-bit ZMM register can store *four* such lanes. Therefore, elements from four different parties can be loaded side by side into a single register. The key instruction, VPCLMULQDQ, multiplies these

four lanes independently without cross-lane interference. As a result, four $64 \times 64 \rightarrow 128$ carry-less multiplications, one for each party, are completed with a single instruction. This contrasts with AVX2, which requires four instructions to perform the same work.
When two 128-bit elements are multiplied, the result can grow up to 256 bits. However, an element of the field $\mathrm{GF}(2^{128})$ must always be 128 bits long. Therefore, the upper part of the product is reduced back to 128 bits according to the irreducible polynomial. This modular reduction is also applied identically to the four lanes, so the four parties are processed together.
Squaring and multiply-accumulate operations are also grouped in units of four parties. For $\mathrm{GF}(2^{128})$, squaring is completed with two carry-less multiplications because the intermediate cross terms vanish. In sections where several products must be added consecutively, modular reduction is not performed after every multiplication. Instead, the products are accumulated first, and reduction is performed only once at the end. Modular reduction is an expensive operation. By delaying it and applying it once, its cost can be amortized over multiple products.
Elements of $\mathrm{GF}(2^{192})$ and $\mathrm{GF}(2^{256})$ occupy 1.5 and 2 128-bit lanes, respectively. The 256-bit case uses exactly two lanes. Thus, four parties fit into two ZMM registers without unused space. In contrast, the 192-bit case requires 1.5 lanes. The remaining 0.5 lane is filled with zero padding. Therefore, part of the lane capacity is wasted in the 192-bit variants.

4.2 Linear Layer: Combining Two Operations into One Instruction with VPTERNLOGQ

The linear layer multiplies an input vector by a binary matrix. The result is equal to the XOR of all rows whose corresponding input bits are 1. Therefore, the implementation scans the input bits one by one. If a bit is 1, the corresponding row is XORed into the accumulator. If a bit is 0, the row is skipped.
The issue is the conditional branch implied by "if the bit is 1." Branching is inefficient in SIMD code. Therefore, we replace the branch with *masking*. The branch-free procedure is as follows. First, one input bit is expanded into a value filled with the same bit. If the bit is 1, the expanded value consists entirely of 1s. If the bit is 0, it consists entirely of 0s. When this mask is ANDed with the row, the row remains unchanged if the bit is 1. If the bit is 0, the result becomes all zero. This value is then XORed into the accumulator.
The key point is that the computation "AND the mask with the row and then XOR the result into the accumulator" is a single *three-input bitwise logic* operation. In AVX2, this requires *two instructions*, AND and XOR. In contrast, **VPTERNLOGQ** in AVX-512 evaluates an arbitrary three-input bitwise logic function in *one instruction*.

4.3 Keccak: AVX-512VL Acceleration

We accelerate the Keccak permutation using the same two ideas described above. In this case, we use the **AVX-512VL** extension. AVX-

Algorithm 1 Linear-layer accumulation: mask generation and VPTERNLOGQ

Require: Party element $a = (h \,\|\, l)$ (upper and lower 64 bits), matrix rows $\mathrm{row}_0, \ldots, \mathrm{row}_{127}$, accumulator k
(h, l, k, row_j are 512-bit registers whose 64-bit lanes contain four parties)

1: **for** $j \leftarrow 63$ **downto** 0 **do** ▷ Upper 64 bits
2: mask $\leftarrow$ Asr(h, 63) ▷ Arithmetic right shift by 63: replicate the most significant bit to the full 64-bit word ($1 \rightarrow$ `1...1`, $0 \rightarrow$ `0...0`)
3: $k \leftarrow$ Vpternlogq(k, mask, row_{64+j}, `0x78`) ▷ $k \oplus (\mathrm{mask} \wedge \mathrm{row})$, AND+XOR in one instruction
4: $h \leftarrow$ Shl(h, 1) ▷ Move the next bit to the most significant position
5: **end for**
6: **for** $j \leftarrow 63$ **downto** 0 **do** ▷ Lower 64 bits, repeated identically with l and row_j
7: mask $\leftarrow$ Asr(l, 63); $k \leftarrow$ Vpternlogq(k, mask, row_j, `0x78`); $l \leftarrow$ Shl(l, 1)
8: **end for**
9: **return** k

512VL allows AVX-512 instructions to be used with 256-bit *YMM* registers.

Packing Multiple Hashes into One Register AIMer invokes the same Keccak permutation many times. Therefore, as shown in Fig. 1, we interleave four Keccak states. The same 64-bit lane from each state is placed side by side in one 256-bit YMM register, giving $4 \times 64 = 256$ bits per lane group. Then, a single instruction processes the corresponding lanes of four states at the same time. As a result, four hash computations proceed in parallel.

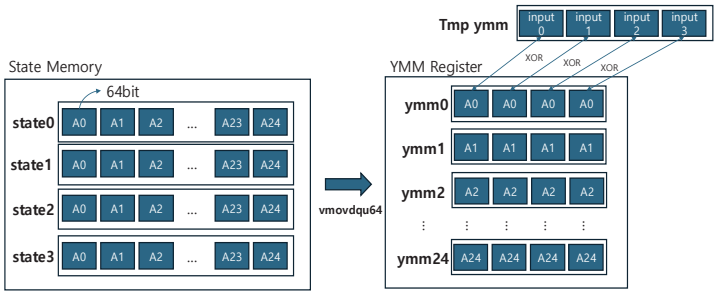


Fig. 1. The same lanes from four Keccak states are collected into one 256-bit YMM register. The blue A0 lanes from the four states form ymm0.

Merging Multiple Operations into One Instruction Each round of the Keccak permutation consists of five steps. Among them, the dominant operations are *bit rotations* and *logical operations*. The rotation step rotates each 64-bit lane by a fixed amount. AVX2 implements this

with three instructions: left shift, right shift, and OR. In contrast, `vprolq` performs the same operation with a single instruction.
The three XOR operations in the diffusion step and the nonlinear operation $a \oplus (\neg b \wedge c)$ are both *bitwise logic operations over three inputs*. Therefore, each of them can be handled by one VPTERNLOGQ instruction.

5 Performance

This section quantitatively evaluates the proposed AVX-512 implementation. We first describe the measurement environment. We then present the speedup over AVX2 for all six AIMer variants. Finally, we decompose the speedup and analyze whether it comes from field arithmetic or Keccak.

5.1 Measurement Environment

The measurements were conducted on an 11th-generation Intel Core i7-1165G7 processor with Ubuntu 24.04 and gcc 13.3.0. The code was compiled with `-O3 -fomit-frame-pointer`. The CPU governor was fixed to `performance`. For reproducibility, **Turbo Boost was disabled**. The benchmark was pinned to a single core using `taskset`. We report the median over 50 measurements for each operation.
The measurement environment is summarized in Table 2.

Table 2. Measurement environment

CPU	11th-generation Intel Core i7-1165G7 (Tiger Lake, base 2.8 GHz)
ISA	AVX-512/VPCLMULQDQ support
OS	Ubuntu 24.04
Compiler	gcc 13.3.0
CFLAGS	`-O3 -fomit-frame-pointer`
CPU governor	Fixed to `performance`
Turbo Boost	**Disabled** for reproducibility
Core pinning	Pinned to a single core using `taskset`
Reporting	Median over 50 measurements per operation

To enable independent reproduction and verification, we publicly release the complete source code, KAT vectors, benchmark scripts, and measurement data in our public GitHub repository.

5.2 Performance Evaluation: Reference, AVX2, and AVX-512

Table 3 shows the median cycle counts and speed ratios of the AVX2 and AVX-512 implementations for signing and verification. For signing,

Table 3. Performance of the reference, AVX2, and AVX-512 implementations. Values are median cycles over 50 measurements with Turbo Boost disabled. Ref./Ours and AVX2/Ours denote speedup ratios relative to our AVX-512 implementation. (*=ours)

Variant	Reference	AVX2	***AVX-512**	Ref./Ours	AVX2/Ours
			Signing		
128f	8,983,574	1,346,647	774,954	11.59	1.74
128s	73,300,846	10,024,946	5,434,747	13.49	1.84
192f	17,564,992	3,391,874	2,117,497	8.30	1.60
192s	137,642,299	25,234,828	15,185,253	9.06	1.66
256f	44,468,943	6,685,139	3,872,719	11.48	1.73
256s	347,547,541	47,950,861	26,097,011	13.32	1.84
			Verification		
128f	8,306,999	1,322,628	784,276	10.59	1.69
128s	71,723,444	9,888,369	5,303,667	13.52	1.86
192f	16,200,801	3,392,433	2,151,221	7.53	1.58
192s	136,979,070	25,051,151	15,023,986	9.12	1.67
256f	40,857,053	6,665,384	3,881,205	10.53	1.72
256s	346,444,256	47,656,786	25,852,180	13.40	1.84

AVX-512 is 1.60–1.84× faster than AVX2. For verification, it is 1.58–1.86× faster.

The speed ratio differs across variants. This can be explained by two factors. First, at the same security level, the small variants (N=256) show larger improvements than the fast variants (N=16). As the number of parties increases, four-party packing is repeated more often, and its benefit accumulates. Second, the 192-bit variants show the smallest improvements. As discussed in Section 4.1, an element of $\mathrm{GF}(2^{192})$ occupies 1.5 lanes, so part of the lane capacity is wasted due to zero padding. In contrast, 128-bit and 256-bit elements fit into lanes without unused space.

A more important question is whether this improvement comes from field arithmetic or Keccak. To separate the two effects, we fixed the Keccak backend to AVX-512VL and changed only the field implementation between AVX2 and AVX-512. The resulting speed ratios are shown in Table 4. The contribution of each component to the overall improvement is shown in Fig. 2.

For the 128-bit and 192-bit variants, the field-only speedup is 1.08–1.11×. Under the already-vectorized AVX2 baseline, Keccak accounts for approximately 86% of the incremental improvement. This baseline-specific ratio is not an absolute execution-time breakdown relative to the scalar reference implementation. In the 256-bit variants, exact two-lane packing increases the field-only speedup to 1.30–1.36× and the field contribution to about 42%.

6 Conclusion

This paper presented an AVX-512 optimized implementation for all six variants of AIMer, a signature scheme selected as a final algorithm in

Table 4. Speed ratios by variant. "Overall" denotes the AVX-512 speedup over AVX2. "Field only" denotes the speed ratio when the Keccak backend is fixed to AVX-512VL and only the field implementation is changed. The difference between the two corresponds to the Keccak contribution.

Variant	Overall	Field only
128f	1.74	1.11
128s	1.84	1.11
192f	1.60	1.08
192s	1.66	1.09
256f	1.73	1.30
256s	1.84	1.36

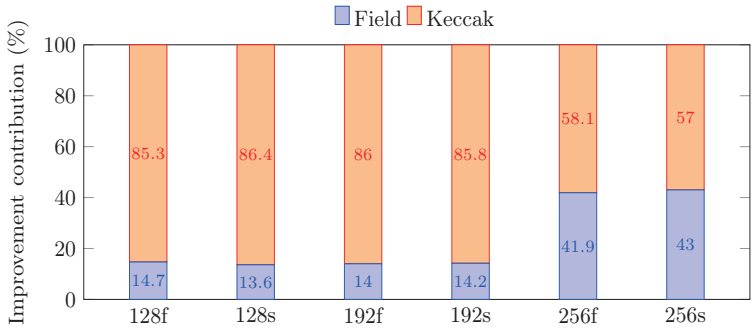


Fig. 2. Contribution of field arithmetic and Keccak to the overall speedup. Keccak accounts for about 86% of the improvement in the 128-bit and 192-bit variants, whereas the field contribution increases to about 42% only in the 256-bit variants.

the KpqC competition. We parallelized binary-field multiplication across four parties per 512-bit register using VPCLMULQDQ. We also handled conditional XOR accumulation in the linear layer using VPTERNLOGQ. In addition, we accelerated Keccak using AVX-512VL. As a result, our implementation achieved a 1.60–1.84× speedup over AVX2 for signing.
We further decomposed this improvement into the contributions of field arithmetic and Keccak. The results show that, in the 128-bit and 192-bit variants, about 86% of the improvement comes from Keccak. Only in the 256-bit variants does the field contribution increase to about 42%. These contribution ratios are specific to the selected AVX2 baseline. For the evaluated transition from AVX2 to AVX-512, Keccak accounts for most of the remaining improvement after field arithmetic has already been vectorized.
Our measurements are limited to a single Tiger Lake CPU with Turbo Boost disabled and one pinned core; other AVX-512 microarchitectures, frequency throttling, and multicore behavior remain unexamined.

Future work will evaluate these factors across other AVX-512 microarchitectures and extend the decomposition to ARM SVE and NEON. Our AVX2-to-AVX-512 results suggest that symmetric primitives remain important optimization targets once field arithmetic has already been vectorized.

7 Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2025-02306395, Development and Demonstration of PQC-Based Joint Certificate PKI Technology, 40%) and this work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25394739, Development of Security Enhancement Technology for Industrial Control Systems Based on S/H-BOM Supply Chain Protection, 30%) and this work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25519543, Development of PQC Migration Technology for Quantum-Safe IC Chip, 30%).

References

1. Anderson, C.S., Zhang, J., Cornea, M.: Enhanced vector math support on the intel® avx-512 architecture. In: 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH). pp. 120–124. IEEE (2018)
2. Baum, C., Beullens, W., Braun, L., de Saint Guilhem, C.D., Klooß, M., Majenz, C., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., et al.: Faest v2: Algorithm specifications (2025)
3. Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In: Annual international cryptology conference. pp. 581–615. Springer (2023)
4. Baum, C., de Saint Guilhem, C.D., Kales, D., Orsini, E., Scholl, P., Zaverucha, G.: Banquet: short and fast signatures from aes. In: IACR International Conference on Public-Key Cryptography. pp. 266–297. Springer (2021)
5. Bernstein, D.J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., Schwabe, P.: The sphincs+ signature framework. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 2129–2146 (2019)
6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Keccak sponge function family main document. Submission to NIST (Round 2) **3**(30), 320–337 (2009)
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Keccak. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 313–314. Springer (2013)

8. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Proceedings of the 2017 acm sigsac conference on computer and communications security. pp. 1825–1842 (2017)
9. Chhugani, J., Nguyen, A.D., Lee, V.W., Macy, W., Hagog, M., Chen, Y.K., Baransi, A., Kumar, S., Dubey, P.: Efficient implementation of sorting on multi-core simd cpu architecture. Proceedings of the VLDB Endowment **1**(2), 1313–1324 (2008)
10. Cho, J., Ha, J., Kim, S., Kwon, J., Lee, B., Lee, J., Lee, J., Lee, S., Moon, D., Son, M., et al.: The aimer signature scheme
11. Drucker, N., Gueron, S., Krasnov, V.: Fast multiplication of binary polynomials with the forthcoming vectorized vpclmulqdq instruction. In: 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH). pp. 115–119. IEEE (2018)
12. Drucker, N., Gueron, S., Krasnov, V.: Making aes great again: the forthcoming vectorized aes instruction. In: 16th International Conference on Information Technology-New Generations (ITNG 2019). pp. 37–41. Springer (2019)
13. Fiege, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing. pp. 210–217 (1987)
14. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. pp. 21–30 (2007)
15. Kales, D., Zaverucha, G.: Efficient lifting for shorter zero-knowledge proofs and post-quantum signatures. Cryptology ePrint Archive (2022)
16. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 525–537 (2018)
17. Kim, S., Ha, J., Son, M., Lee, B., Moon, D., Lee, J., Lee, S., Kwon, J., Cho, J., Yoon, H., et al.: The aimer signature scheme. Tech. rep., Technical report. NIST (2023)
18. Lee, M., Jang, K., Kwon, H., Sim, M., Song, G., Seo, H.: High-speed implementation of aim symmetric primitives within aimer digital signature. Cryptology ePrint Archive (2023)
19. Lee, S.W., Kim, H.G., Song, M.H., Eum, S.W., Seo, H.J.: Accelerating faest signatures on arm: Neon simd aes and parallel vole optimization. Applied Sciences **16**(8), 3782 (2026)
20. Lee, S.W., Song, M.H., Kim, H.G., Kim, U.J., Eum, S.W., Seo, H.J.: Advanced vector extensions 512 acceleration of lsh and lea-gcm. Cryptology ePrint Archive (2026)
21. Liu, F., Mahzoun, M., Øygarden, M., Meier, W.: Algebraic attacks on rain and aim using equivalent representations. IACR Transactions on Symmetric Cryptology **2023**(4), 166–186 (2023)
22. Saarinen, M.J.O.: Accelerating slh-dsa by two orders of magnitude with a single hash unit. In: Annual International Cryptology Conference. pp. 276–304. Springer (2024)

23. Delpech de Saint Guilhem, C., Orsini, E., Tanguy, T.: Limbo: efficient zero-knowledge mpcith-based arguments. In: Proceedings of the 2021 acm sigsac conference on computer and communications security. pp. 3022–3036 (2021)
24. Song, G., Jang, K., Yoon, S., Lee, M., Seo, H.: Quantum circuit implementation and resource analysis of aim2. In: International Conference on Information Security and Cryptology. pp. 220–233. Springer (2025)
25. Xu, R., He, D., Luo, M., Peng, C., Zeng, X.: Optimizing dilithium implementation with avx2/-512. ACM Transactions on Embedded Computing Systems **23**(6), 1–30 (2024)

Blind Trace-Only Segmentation of Cipher Implementations Without Algorithm Metadata

Hyunjun Kim[1[0000−0001−6757−6109]], Hwajeong Seo[2[0000−0003−0069−9061]], and Anupam Chattopadhyay[1[0000−0002−8818−6983]‡]

[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Division of IT Convergence Engineering, Hansung University, Seoul, South Korea
khj930704@gmail.com, hwajeong84@gmail.com, anupam@ntu.edu.sg

Abstract. Blind side-channel analysis (BSCA) requires an upstream mechanism for locating repeated execution and candidate leakage regions before key inference can be applied to an unlabeled trace. We formulate this task as trace-only discovery of implementation-level repetition structure and propose a two-stage method using only the per-sample mean and standard deviation, without plaintext, ciphertext, algorithm labels, source code, assembly, execution markers, or known points of interest. Stage 1 combines rank-normalized self-similarity across four summary-statistic channels to rank repetition scales, estimate their phase, and extract an anchor-supported stable core. Stage 2 stacks the stable-core windows into a representative repetition and partitions its relative time axis into high- and low-score segments.
We evaluate 16 block-cipher implementations on STM32F303 and XMEGA. Twelve targets form a continuous stable core, the median waveform-consistency score is 0.989, and twelve targets achieve $B \geq 0.97$. Post-hoc source and assembly analysis maps the Rank-1 trace repetitions to the compiled execution hierarchy: five targets show direct repeated-body count correspondence, two include adjacent boundary or whitening structure, six expose grouped macro bodies or finer-grained internal motifs, and three remain structurally ambiguous. In an AES/XMEGA case study, the trace-only high-score segments occupy 49.1% of the representative repetition while covering 100% of the top 0.5% and top 0.1% S-box CPA samples. These results show that the proposed method converts unlabeled traces into source/assembly-supported repetition coordinates and leakage-relevant candidate regions for subsequent CPA or BSCA.

Keywords: Power analysis · Blind side-channel analysis · Trace structuring · Self-similarity analysis · Trace alignment · Points of interest · Block cipher implementations.

1 Introduction

Side-channel analysis (SCA) exploits physical leakage such as power, electromagnetic emanation, or timing to infer implementation-dependent secret state.

‡ Corresponding author.

In ordinary non-profiled SCA, the attacker uses known plaintext or ciphertext, enumerates key candidates, and tests a leakage model at selected points of interest (POIs). The quality of the POI selection and the alignment of repeated operations therefore strongly affect attack cost and success.

Blind side-channel analysis (BSCA) aims to reduce or remove the plaintext/ciphertext requirement by exploiting relationships among internal values. It has progressed from joint-distribution attacks to belief propagation, multivariate leakage, deep learning, and multi-POI methods, and has been applied beyond AES to AEAD, PQC, and ARX targets [1,2,3,4,8,9,10,11]. Yet many BSCA procedures still begin after the trace has already been localized: the target time region, repetition structure, POI candidates, or correspondence between repeated operations is assumed or obtained through profiling. In a realistic measurement setting, source code, assembly, execution markers, communication data, and reliable round boundaries may not be available together.

This paper addresses the missing upstream step as the *POI- and structure-blind trace structuring problem*. Given only measured traces and no plaintext, ciphertext, key, source code, assembly, execution markers, or known POIs, the goal is to estimate a repetition scale, start phase, stable repetition core, and intra-repetition candidate segments that can guide later CPA or BSCA. We use only the per-sample mean and standard deviation over traces. Stage 1 ranks self-similar lags across four summary-statistic features, aligns candidate repetition windows, and extracts the longest anchor-supported stable core. Stage 2 stacks that core into a representative repetition and partitions its relative time axis into high- and low-score segments. The result is a hierarchical coordinate system consisting of repetition index and intra-repetition relative position. The repetition scale is determined by the structure expressed in the measured trace and may therefore represent a compiled loop body, a grouped body, or a recurring internal operation.

Contributions. First, we formulate the trace-structuring stage that must precede key inference under a POI- and structure-blind condition. Second, we propose a two-stage trace-only method that derives repetition-window structure and internal segments from mean/std statistics alone. Third, using 16 block-cipher implementations on STM32F303 and XMEGA, we evaluate the generated coordinates at two levels: source and assembly analysis identifies the compiled repetition structures represented by the trace-derived scales, while known-key CPA evaluates whether the resulting internal segments overlap leakage-relevant regions. All three are used only after the trace-only outputs have been fixed.

Section 2 reviews related work and positions this study. Section 3 describes the proposed two-stage trace-only structuring technique, and Section 4 presents experimental results on 16 implementations across two platforms. Section 5 concludes and discusses future work.

Table 1. Input assumptions and output objectives of nearby methods.

Method	Input information	Prior structure	Output objective
SCATTER [5]	Trace, plaintext, key candidate, window	Fixed target and intermediate; alignment is relaxed	Distinguish key candidates
Generic SCARE [6]	Trace, input, output	Constant-time 8-bit execution with observed I/O	Recover an unknown procedure
This study	Trace mean/std only	No plaintext, key, POI, structure, alignment marker, or code	Repetition scale, phase, stable core, segments

2 Related Work

BSCA studies have substantially weakened the input/output assumptions of classical SCA. Joint-distribution key inference [1], multi-round belief propagation [2], multivariate attacks on masked implementations [4], blind SIFA and AEAD attacks [7,8], PQC attacks [9], deep-learning-based multi-POI inference [10], and ARX-oriented attacks [11] all show that key inference can proceed with less direct use of plaintext or ciphertext. However, these works usually rely on some prior localization: leakage windows, candidate POIs, algorithm structure, or temporal correspondence between repeated operations are known, profiled, or manually selected. Our work targets this earlier localization stage rather than the subsequent key-candidate scoring model.

Unlabeled POI search gives another point of comparison. Variance analysis labeling [3] and leakage-detection statistics [12] identify sample positions likely to carry data-dependent leakage. Such methods are valuable but operate mainly at the sample-POI level. Here the first output is a repetition-level coordinate system: period, phase, stable core, and relative segments. Alignment and reverse-engineering work is also adjacent. SCATTER [5] reduces sensitivity to jitter/shuffling once the target algorithm, intermediate value, key candidate, and leakage window have been set; Generic SCARE [6] reconstructs an unknown procedure from input, output, and execution traces. Generic SCARE is therefore closer to procedure reconstruction, whereas this work produces a segmentation and coordinate system for later side-channel analysis. In contrast, the proposed method uses only mean/std trace statistics and produces a compact candidate structure before any plaintext, ciphertext, key candidate, source/assembly reference, or execution marker is used.

Table 1 summarizes the position of this work relative to two nearby methods. These methods are not direct baselines because their input assumptions and output objectives differ; they are included to clarify the position of the proposed upstream structuring task.

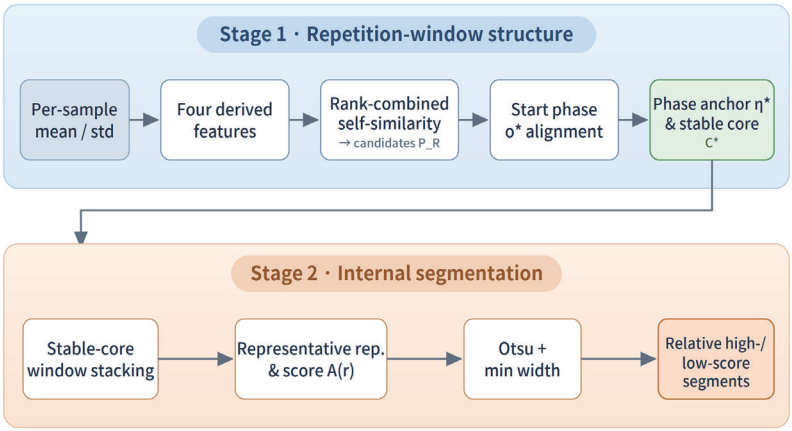


Fig. 1. The proposed two-stage trace-only repetition structuring process.

3 Proposed Method

3.1 Overview

The method receives only the per-sample mean μ_t and standard deviation σ_t of measured traces,

$$\mathcal{D} = \{\, (t, \mu_t, \sigma_t) \mid t = 0, \ldots, N-1 \,\}. \tag{1}$$

It does not use the algorithm name, number of rounds, source code, assembly, external markers, plaintext/ciphertext, key, or ground-truth POIs. Stage 1 structures the full trace into repetition windows and returns a dominant repetition scale, start phase, anchor, and stable core. Stage 2 stacks stable core windows into a representative repetition and divides its relative time axis into high- and low-score segments. Figure 1 summarizes the flow.

The design assumes that repeated execution leaves reproducible structure in the mean or standard deviation, that correctly cut repetition windows show high waveform consistency and recurring anchors, and that internal features are stable in relative rather than absolute time. The selected period represents the dominant implementation-level repetition scale expressed in the trace statistics, ranging from repeated loop bodies and grouped macro structures to finer-grained recurring operations.

3.2 Stage 1: Repetition-Window Structure Estimation

Four non-negative features capture waveform magnitude and boundary changes:

$$x_t^{(1)} = \sigma_t, \quad x_t^{(2)} = |\mu_t|, \quad x_t^{(3)} = |\nabla \mu_t|, \quad x_t^{(4)} = |\nabla \sigma_t|. \tag{2}$$

Table 2. Trace-only outputs and their intended interpretation.

Output	Role in the coordinate system	Use in later analysis
$\mathcal{P}_R$	Candidate repetition scales ranked by consensus self-similarity	Provides alternative trace-only coordinate systems when several repetition scales are visible
P^*, o^*	Rank-1 period and aligned start phase	Converts absolute samples into repetition index and relative position
$\eta^*, \mathcal{C}^*$	Recurring anchor and longest supported window run	Selects a continuous set of windows for the representative repetition
$\{S_q\}$	Relative high-/low-score internal segments	Prioritizes regions for later CPA or BSCA

The channels combine magnitude and boundary views. Each feature is centered and normalized before autocorrelation; raw autocorrelation scores are converted to within-channel percentile ranks, with larger ranks better, and then averaged:

$$C(\ell) = \frac{1}{4}\sum_{j=1}^{4} \mathrm{rank}_{\ell\in\mathcal{L}}\, R_j(\ell), \quad \mathcal{P}_R = \underset{\ell\in\mathcal{L}}{\text{TopR-LocalMax}}\, C(\ell), \quad P^* = P_1. \tag{3}$$

For candidate origin o, the k-th window starts at $o + kP^*$ and is represented by an in-window centered, L_2-normalized vector $z^{(j)}_{k,o}$. The alignment score is

$$B(o) = \frac{1}{4}\sum_{j=1}^{4} \underset{k<k'}{\mathrm{median}}\, \mathrm{corr}\left(z^{(j)}_{k,o}, z^{(j)}_{k',o}\right). \tag{4}$$

To ensure that windows share a recurring internal event, the median template $T_j(r)$ is searched for peak anchors $\eta = (j, a)$. The stable core is the longest consecutive run of anchor-supporting windows:

$$T_j(r) = \underset{k}{\mathrm{median}}\, x^{(j)}_{o^*+kP^*+r}, \quad \mathcal{C}^* = \mathrm{LongestConsecutiveRun}(\mathcal{K}_A(\eta^*)). \tag{5}$$

Thus Stage 1 outputs $(\mathcal{P}_R, P^*, o^*, \eta^*, \mathcal{C}^*)$. The stable core indicates continuous anchor support in the trace statistics; it is not an independently verified execution interval.

Procedure 1: Trace-only repetition estimation. Compute the four channels in Eq. 2. For every lag in $\mathcal{L}$, compute normalized autocorrelation per channel, convert each channel's lag scores to percentile ranks with larger ranks better, and average the ranks to obtain $C(\ell)$. A lag is a local maximum if its score is no smaller than both neighboring scores; the Top-R local maxima, ordered by $C(\ell)$, form $\mathcal{P}_R$, and $P^* = P_1$. Then scan 96 coarse origins, refine near the best origin, and among origins within the fixed correlation tolerance select o^* lexicographically by $(K(o), B(o), -o)$. Finally, enumerate median-template peaks, count a window as supporting an anchor only when a salient per-window peak falls within the fixed phase band, and choose η^* lexicographically by maximum support, minimum duplicate in-band peaks, minimum median phase error, channel, and phase. The longest consecutive supporting run is $\mathcal{C}^*$. All fixed thresholds are listed in Table 3.

3.3 Stage 2: Internal Segmentation of the Representative Repetition

Stage 2 keeps P^*, o^*, and $\mathcal{C}^*$ fixed. It stacks stable-core windows onto a common relative axis and builds a representative template. For relative position r,

$$\hat{T}_j(r) = \operatorname*{median}_{k \in \mathcal{C}^*} x^{(j)}_{o^*+kP^*+r}, \quad A(r) = \frac{1}{4}\sum_{j=1}^{4} \mathrm{rank}_{u\in[0,P^*)}\Big(\hat{T}_j(u)\Big)\Big|_{u=r}. \tag{6}$$

$A(r)$ is a dimensionless prominence score, not a leakage-strength estimate for a specific intermediate value.

Procedure 2: Internal segmentation. Choose the longest anchor-supported run, stack its aligned windows, and compute $A(r)$ by averaging per-channel percentile ranks on the relative axis. Smooth $A(r)$, apply Otsu thresholding to obtain high/low labels, and repeatedly merge runs shorter than the fixed minimum width into a neighbor: edge runs use the only neighbor, equal-label neighbors are joined, otherwise the wider neighbor is chosen, with ties broken by closer mean score. The resulting activity/context intervals are projected to each selected window at the same relative positions.

The output segments are operational candidate regions:

$$\begin{aligned} &\mathcal{A} = \{\, r \mid \bar{A}(r) \geq \tau_{\mathrm{Otsu}} \,\}, \qquad \{S_q\}_{q=1}^{Q} = \mathrm{Segments}_{w_{\min}}(\mathcal{A}), \\ &\Theta = (\mathcal{P}_R,\, P^*,\, o^*,\, \eta^*,\, \mathcal{C}^*,\, \{S_q\}_{q=1}^{Q}). \end{aligned} \tag{7}$$

Inter-repetition support and boundary strength are reported only as descriptive indicators of the generated segments, not as additional generation criteria.

4 Evaluation

4.1 Experimental Setup and Metrics

We evaluate 16 targets: AES, GIFT, HIGHT, PRESENT, RECTANGLE, SIMON, SM4, and SPECK on STM32F303 and XMEGA. We assume that traces are coarsely capture-aligned to the target invocation, but no internal execution marker, round boundary, PC trace, plaintext/ciphertext, key, source code, assembly, or POI label is used by the method. For each target, multiple measured traces were summarized into per-sample mean and standard deviation before analysis. Table 3 summarizes the fixed trace-only conditions.

Stage 1 is evaluated along two complementary axes. First, K, K_A, L_C, and B characterize the internal coherence of the trace-derived coordinate system. Here, K is the number of aligned complete windows, K_A is the number of anchor-supporting windows, L_C is the length of the longest consecutive anchor-supported run, and B measures waveform consistency. These quantities are evaluated without source or assembly information.

Second, after all trace-only outputs are fixed, source and compiled assembly are inspected to determine which implementation structure is represented by each selected repetition scale. N_{ref} denotes the count of the repeated source/assembly body used as the primary implementation reference. A direct count

Table 3. Fixed parameters used for all targets.

Stage	Fixed setting
Input	1,000 traces per target, summarized as mean/std from coarse invocation-trigger alignment only; no internal marker, round boundary, or PC trace
Period search	channels σ, $\lvert\mu\rvert$, $\lvert\nabla\mu\rvert$, $\lvert\nabla\sigma\rvert$; $\ell = 80, \ldots, \min(20000, \lfloor N/2 \rfloor - 1)$; percentile-rank autocorrelation consensus; local maxima; Top-$R = 5$
Phase/anchor	96 coarse origin steps plus local refinement; origin correlation tolerance 0.01; anchor peak quantile 0.97, spacing $0.01P^*$, phase band $\delta = 0.025P^*$
Internal segments	longest anchor-supported run; smoothing $0.0025P^*$; Otsu 128 bins; minimum width $0.002P^*$; short-run neighbor merge and window projection
Post-hoc use	plaintext/ciphertext, key, algorithm label, source/assembly, and POI labels are excluded during generation; source/ASM/CPA are used only after outputs are fixed

correspondence is reported when the aligned-window count differs from N_{ref} by at most one, allowing a boundary-like window to be included or omitted. Grouped-body and finer-grained labels are assigned only when the corresponding compiled structure is supported by source or assembly inspection.

Integer multiples and divisors are recorded only as descriptive count relations for organizing nested or grouped structures; they are not used as an accuracy criterion or aggregated success measure. K and L_C are reported separately because K describes the full window sequence, whereas L_C describes the longest stable subset. Likewise, the Top-5 periods are reported as ranked alternative coordinate systems rather than as a recovery-rate metric.

4.2 Cross-Implementation Readout

The primary cross-target evaluation examines whether the Rank-1 trace-only scale yields coherent repetition windows with continuous anchor support.

Figure 2 is used as a global stability overview for all targets; detailed waveform interpretation is reserved for the representative AES/XMEGA readout in Figure 3. In 12/16 cases, $K = K_A = L_C$, so every aligned window belonged to one continuous stable core. The median B was 0.989, and 12 targets had $B \geq 0.97$. These results show that, for 12 of the 16 implementations, the Rank-1 scale converts the trace into a fully supported consecutive window sequence, while the consistency and core-length measures separately identify the remaining unstable or partial cases.

Figure 3 shows AES/XMEGA, where $P^* = 4{,}400$, $o^* = 572$, and all 9 aligned windows were included in the stable core. The waveform consistency was 0.994 and the median correlation to the standard-deviation template was 0.996. This detailed exact-case readout complements the global overview in Figure 2. PRESENT/F303, RECTANGLE/XMEGA, and SM4/XMEGA instead produced short stable cores, while GIFT/XMEGA retained all windows but with low waveform consistency. This separation shows why both waveform consistency and anchor continuity are needed.

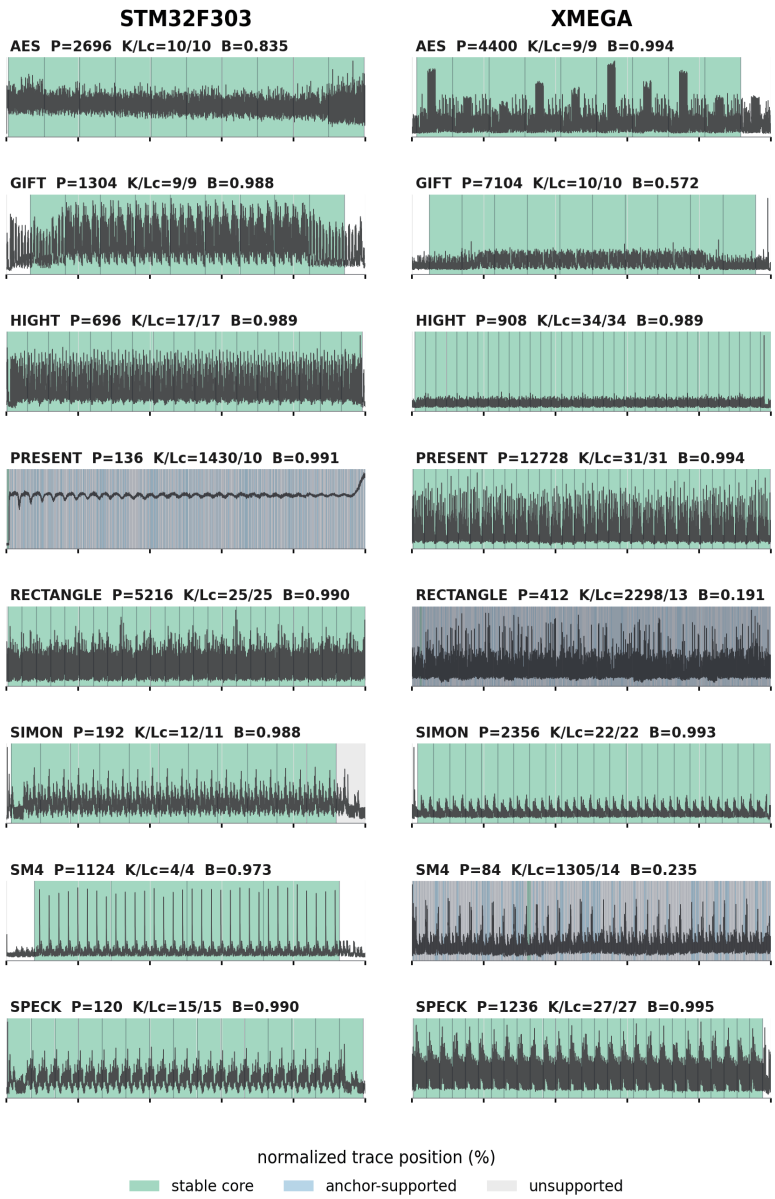


Fig. 2. Rank-1 trace-only repetition-window segmentation results for 16 targets. Green denotes the stable core, blue the anchor-supporting windows outside the stable core, and gray the anchor-unsupported aligned windows. The figure is intended to show global stability rather than detailed waveform differences.

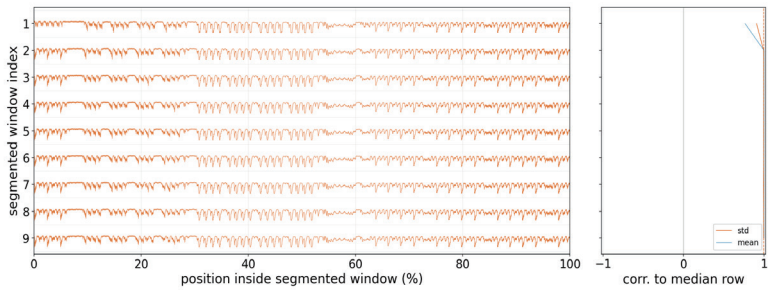


Fig. 3. Representative exact/stable case: standard-deviation waveforms of the 9 aligned AES/XMEGA repetition windows (left) and their correlation to the median repetition window (right).

4.3 Mapping Trace-Derived Repetitions to Compiled Implementation Structure

Table 4 maps each fixed Rank-1 trace repetition to the corresponding structure identified through post-hoc source and assembly inspection. The mapping distinguishes direct repeated bodies, boundary-inclusive bodies, grouped bodies, finer-grained internal motifs, and structurally ambiguous cases. These labels characterize the implementation level expressed by the trace-derived scale.

The Rank-1 outputs reveal several levels of compiled repetition structure. Five targets correspond directly to the repeated implementation body, and two additionally include boundary- or whitening-like execution. HIGHT/F303, SIMON/F303, and SM4/F303 expose grouped bodies, whereas PRESENT/F303, RECTANGLE/XMEGA, and SM4/XMEGA are dominated by finer instruction-, substitution-, or permutation-level motifs. The remaining three targets do not admit a unique source/assembly interpretation from the Rank-1 count alone. Direct or boundary-inclusive body correspondence is more frequent on XMEGA than on STM32F303, while grouped scales occur more often in the STM32F303 results. We report this as an observed platform difference without attributing it to a specific architectural or compiler cause.

4.4 Internal Segmentation Case Study

For AES/XMEGA, known-key CPA is performed only after the trace-only internal segments are fixed, and is used only for post-hoc interpretation. The CPA models are $\mathrm{HW}(PT)$, $\mathrm{HW}(PT \oplus K)$, and $\mathrm{HW}(\mathrm{Sbox}(PT \oplus K))$. At each relative sample we use the maximum absolute correlation over 16 bytes. Hotspot coverage is the fraction of top-p CPA samples falling in the fixed high-score set $\mathcal{A}$.

Figure 4 shows the relative activity profile and the internal segments of the representative repetition formed from 9 aligned windows. The same relative segments are projected to the median features and to each individual window, confirming a shared internal coordinate system.

Table 4. Mapping fixed Rank-1 trace repetitions to source/assembly-supported compiled implementation structures.

Target	P^*	K	L_C	Source/ASM repetition structure	Trace-to-implementation mapping
AES/F303	2696	10	10	nine repeated bodies with adjacent boundary-like execution	boundary-inclusive
GIFT/F303	1304	9	9	seven four-round macro bodies with packing/unpacking helpers	structurally ambiguous
HIGHT/F303	696	17	17	32 round stages plus whitening, with adjacent stages paired per window	grouped body
PRESENT/F303	136	1430	10	repeated S-box/permutation inner sequence	finer motif
RECTANGLE/F303	5216	25	25	25 compiled round bodies	direct body
SIMON/F303	192	12	11	22 two-round loop bodies, with adjacent bodies paired per window	grouped body
SM4/F303	1124	4	4	eight update bodies compiled as paired four-round groups	grouped body
SPECK/F303	120	15	15	27 one-round ARX loop bodies with an odd final round	structurally ambiguous
AES/XMEGA	4400	9	9	nine repeated encryption-state bodies	direct body
GIFT/XMEGA	7104	10	10	seven four-round macro bodies with packing/unpacking helpers	structurally ambiguous
HIGHT/XMEGA	908	34	34	32 round bodies with initial and final whitening	boundary-inclusive
PRESENT/XMEGA	12728	31	31	31 compiled outer-round bodies	direct body
RECTANGLE/XMEGA	412	2298	13	nested bit/permutation loops inside 25 outer rounds	finer motif
SIMON/XMEGA	2356	22	22	22 compiled two-round loop bodies	direct body
SM4/XMEGA	84	1305	14	recurring operations inside eight unrolled four-round groups	finer motif
SPECK/XMEGA	1236	27	27	27 compiled one-round ARX loop bodies	direct body

Note. P^*, K, and L_C are fixed trace-only outputs. The final two columns are assigned only after source and assembly inspection.

Figure 5 shows the result of post-hoc overlaying CPA waveforms on the fixed body-01 segments. The compact coverage summary below reports the same result numerically; high-score segments occupy 49.1% of the representative body.

CPA model	Top 0.5%	Top 0.1%
HW(PT)	45.5%	60.0%
HW($PT \oplus K$)	0.0%	0.0%
HW(Sbox($PT \oplus K$))	100.0%	100.0%

The Stage 2 segments therefore capture a strong S-box CPA hotspot while remaining candidate regions rather than ground-truth POIs. This representative case study does not establish cross-target CPA reduction; it tests whether fixed trace-only segments can align with a known leakage hotspot in one target.

4.5 Discussion and Limitations

The experiments show that Rank-1 and Top-5 results should be interpreted at different levels. Rank-1 evaluates the single period that the unsupervised procedure would use by default. It is therefore the strictest trace-only output: if

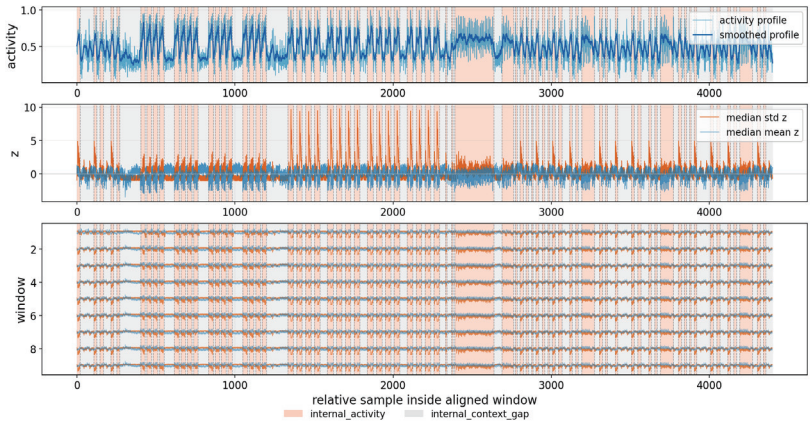


Fig. 4. Internal segmentation of the representative repetition: the relative activity profile and its smoothed version (top), the median standardized mean and standard-deviation features (middle), and the resulting internal segments commonly applied to the aligned repetition windows (bottom).

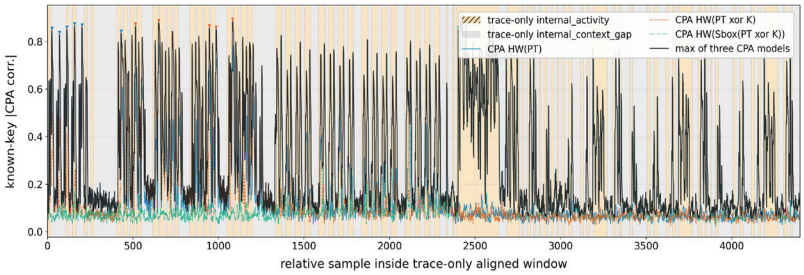


Fig. 5. Correspondence between the fixed internal segments in body 01 and the HW(PT), HW($PT \oplus K$), and HW(Sbox($PT \oplus K$)) CPA waveforms.

it matches the source/assembly body count, the method has found a directly usable implementation-level cadence. Top-5, in contrast, is not selected using source-level knowledge. It is an ordered set of alternative trace-only coordinate systems; downstream analysis can inspect candidates in that order and deprioritize candidates with very short cores, extreme window counts, or low waveform consistency. This distinction matters because embedded implementations often contain several valid repetition scales at once: loop bodies, unrolled macro bodies, table-lookup motifs, permutation substeps, and short instruction sequences can all recur periodically in the power trace.

The main failure modes are also informative. An edge-inclusive count, as in AES/STM32F303 or HIGHT/XMEGA, indicates that setup or whitening-like code has a waveform similar enough to the body to be absorbed into the window sequence. A grouped count indicates that the visible cadence combines several

semantic rounds or loop bodies into one trace window. A microperiod indicates that an inner operation dominates the mean/std statistics more strongly than the algorithmic body. A short stable core means that only a contiguous subset of the aligned windows maintains the selected anchor. These cases are not equivalent to sample-level segmentation errors; rather, they identify which implementation layer is most visible under the trace-only statistics.

For downstream CPA or BSCA, the intended use is conservative: try Rank-1 first, inspect Top-5 alternatives in trace-only order when Rank-1 is short, unstable, or too numerous, and use $\{S_q\}$ only to prioritize the first search region. The AES/XMEGA overlay covers one strong S-box hotspot, but other key-dependent correlations may still lie outside the high-score set; the output is therefore a prioritization mechanism, not a final POI oracle.

The method provides a trace-only candidate coordinate system, not a complete key recovery attack. P^* may select a round, macro body, harmonic period, or instruction-level microperiod; o^* is not sample-level verified without an execution marker or PC trace; and $\{S_q\}$ denotes relative statistical prominence, not exhaustive leakage locations. The next step is to connect this coordinate system to blind labeling and key-inference procedures and measure the resulting search-cost reduction.

5 Conclusion

This paper introduced the trace-only structuring problem that precedes blind side-channel key inference and proposed a two-stage method based only on the per-sample mean and standard deviation. Stage 1 discovers dominant repetition scales in the measured trace, aligns the corresponding windows, and extracts an anchor-supported stable core. Stage 2 organizes the stable windows into a representative repetition and identifies relative high- and low-score internal regions. The resulting output is a hierarchical coordinate system that links repetition index with intra-repetition position without using algorithm metadata or known leakage locations.

Across 16 block-cipher implementations on STM32F303 and XMEGA, 12 targets produced a continuous stable core and 12 achieved waveform consistency $B \geq 0.97$, with a median score of 0.989. Post-hoc source and assembly analysis identified which compiled execution level was expressed by each trace-derived repetition. Five Rank-1 outputs showed direct correspondence with the repeated implementation-body count, two included adjacent boundary or whitening structure, and six reflected grouped macro bodies or finer-grained internal motifs; three cases remained structurally ambiguous. These results demonstrate that the proposed method captures the dominant repetition hierarchy expressed by the compiled implementation, including both body-level and nested internal structures.

The AES/XMEGA case study provided complementary evidence at the intra-repetition level. The high-score segments selected from trace statistics occupied 49.1% of the representative repetition and covered all samples in the top 0.5%

and top 0.1% of the S-box-based CPA waveform. The source and assembly comparison therefore explains the implementation structure represented by the discovered repetition scale, while the CPA comparison confirms that its internal segmentation can prioritize regions containing strong data-dependent leakage.

Overall, the proposed pipeline transforms unlabeled measurements into an implementation-grounded coordinate system for subsequent side-channel analysis. It provides both repetition-level structure and leakage-relevant internal regions before plaintext-dependent CPA or blind key-inference procedures are applied. Future work will extend this evaluation to additional downstream attacks and quantify the resulting reduction in POI and key-inference search cost.

6 Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2025-02306395, Development and Demonstration of PQC-Based Joint Certificate PKI Technology, 25%) and this work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25394739, Development of Security Enhancement Technology for Industrial Control Systems Based on S/HBOM Supply Chain Protection, 25%) and this work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25519543, Development of PQC Migration Technology for Quantum-Safe IC Chip, 25%) and this work was supported by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2020-II201797, 25%).

References

1. Linge, Y., Dumas, C., Lambert-Lacroix, S.: Using the joint distributions of a cryptographic function in side channel analysis. In: Constructive Side-Channel Analysis and Secure Design (COSADE 2014). LNCS, vol. 8622, pp. 199–213. Springer (2014). https://doi.org/10.1007/978-3-319-10175-0_14
2. Le Bouder, H., Lashermes, R., Linge, Y., Thomas, G., Zie, J.-Y.: A multi-round side channel attack on AES using belief propagation. In: Foundations and Practice of Security (FPS 2016). LNCS, vol. 10128, pp. 199–213. Springer (2017). https://doi.org/10.1007/978-3-319-51966-1_13
3. Clavier, C., Reynaud, L.: Improved blind side-channel analysis by exploitation of joint distributions of leakages. In: Cryptographic Hardware and Embedded Systems (CHES 2017). LNCS, vol. 10529, pp. 24–44. Springer (2017). https://doi.org/10.1007/978-3-319-66787-4_2
4. Clavier, C., Reynaud, L., Wurcker, A.: Quadrivariate improved blind side-channel analysis on Boolean masked AES. In: Constructive Side-Channel Analysis and Secure Design (COSADE 2018). LNCS, vol. 10815, pp. 153–167. Springer (2018). https://doi.org/10.1007/978-3-319-89641-0_9

5. Thiebeauld, H., Gagnerot, G., Wurcker, A., Clavier, C.: SCATTER: a new dimension in side-channel. In: Constructive Side-Channel Analysis and Secure Design (COSADE 2018). LNCS, vol. 10815, pp. 135–152. Springer (2018). https://doi.org/10.1007/978-3-319-89641-0_8
6. Lashermes, R., Le Bouder, H.: Generic SCARE: reverse engineering without knowing the algorithm nor the machine. Journal of Cryptographic Engineering **14**(2), 399–414 (2024). https://doi.org/10.1007/s13389-024-00356-2
7. Azouaoui, M., Papagiannopoulos, K., Zürner, D.: Blind side-channel SIFA. In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 555–560. IEEE (2021). https://doi.org/10.23919/DATE51398.2021.9474245
8. Sarry, M., Le Bouder, H., Maaloouf, E., Thomas, G.: Blind side channel analysis against AEAD with a belief propagation approach. In: Smart Card Research and Advanced Applications (CARDIS 2023). LNCS, vol. 14530, pp. 127–147. Springer (2024). https://doi.org/10.1007/978-3-031-54409-5_7
9. Ravi, P., Jap, D., Bhasin, S., Chattopadhyay, A.: Invited paper: machine learning based blind side-channel attacks on PQC-based KEMs — a case study of Kyber KEM. In: 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD), pp. 1–7. IEEE (2023). https://doi.org/10.1109/ICCAD57390.2023.10323721
10. Rezaeezade, A., Yap, T., Jap, D., Bhasin, S., Picek, S.: Breaking the blindfold: deep learning-based blind side-channel analysis. In: 34th USENIX Security Symposium (USENIX Security 25), pp. 5777–5796. USENIX Association (2025)
11. Lee, I., Kim, G., Hong, S., Kim, H.: MALeak: blind side-channel key recovery exploiting modular addition leakage in ARX-based block ciphers. IACR Cryptology ePrint Archive, Paper 2026/067 (2026)
12. Durvaux, F., Standaert, F.-X.: From improved leakage detection to the detection of points of interests in leakage traces. In: Advances in Cryptology — EUROCRYPT 2016. LNCS, vol. 9665, pp. 240–262. Springer (2016). https://doi.org/10.1007/978-3-662-49890-3_10

Evaluating Hybrid KEM/DSA for KpqC and NIST PQC on ARM Cortex-M4

Minjoo Sim[1][0000-0001-5242-214X], Minwoo Lee[1][0000-0002-2356-3055], Subeen Cho[2][0009-0004-4352-8021], Yulim Hyoung[2][0009-0009-7530-6497], and Hwajeong Seo[2][0000-0003-0069-9061]

[1] Department of Information Computer Engineering, Hansung University, Seoul (02876), South Korea
{minjoos9797, minunejip}@gmail.com
[2] Department of Convergence Security, Hansung University, Seoul (02876), South Korea
{chosubin1208,yulim4hyoung, hwajeong84 }@gmail.com

Abstract. Primitive-only PQC benchmarks are insufficient for attributing composed hybrid costs on Cortex-M4 because shared hash backends, signing-time variability, and fixed classical/wrapper work affect measured performance. We implement a common bare-metal Cortex-M4 harness for representative KpqC/NIST families, measuring uniform Hash-CT hybrid KEM benchmark rows with X25519 and Bindel et al. hybrid-signature AND-combiner rows. The goal is composed-cost attribution under a uniform benchmark transcript rather than primitive-only ranking.

Our measurements show three effects relevant to cost attribution. First, replacing only Keccak-f[1600] changes SHAKE-heavy signing by up to 2.16×, while SPHINCS$^+$-SHA2 and FN-DSA control rows remain at 1.00×. Second, median-only signature tables do not characterize the observed signing-time variability. In the nominal level-5 signing rows, HAETAE5 has lower median and mean signing latency than FN-DSA-1024, but its largest observed $N = 100$ measurement reaches 4.53× its median while the FN-DSA-1024 maximum is less than 1% above its median. These maxima are descriptive and do not estimate tail probabilities. Third, a local SMAUG-T backend improves standalone SMAUG-T by 1.79–1.82×, but the composition-level speedup drops to 1.42–1.65× with SMAUG-T+X25519. The KEM rows report 3.7–8.8 M cycles of summed KeyGen/Encaps/Decaps work for lattice-based KEMs and 19.6–74.5 M cycles for HQC. Together, the results motivate reporting composed-cost attribution, backend provenance, and observed variability alongside primitive timings.

Keywords: Post-Quantum Cryptography · Hybrid KEM · Hybrid DSA · ARM Cortex-M4 · Embedded Benchmarking · KpqC

1 Introduction

Post-quantum deployment is now an integration problem, not only a primitive-selection problem. On microcontrollers, the combiner, transcript, byte layout, classical partner, and shared low-level backend can materially affect latency, wire size, stack, and flash [13,28]. This paper shows that Cortex-M4 hybrid post-quantum cryptography (PQC) cannot be understood from primitive timings alone, because shared symmetric backends, signing-time variability, and optimization dilution can change composed-cost conclusions even when the component primitives are fixed.

We evaluate representative families selected by the National Institute of Standards and Technology (NIST) and the Korean Post-Quantum Cryptography (KpqC) competition. NIST has finalized ML-KEM, ML-DSA, and SLH-DSA [24,25,26], while FN-DSA and HQC have been selected for ongoing NIST standardization [29,27]. KpqC has selected SMAUG-T, NTRU+, HAETAE, and AIMer [18]. These families cover lattice-based, code-based, multiparty computation-in-the-head, and hash-based designs, making them a useful portfolio for studying composed cost, backend sensitivity, and signing-time variability.

The key encapsulation mechanism (KEM) rows use a uniform ciphertext-hashing (Hash-CT) benchmark transcript with X25519, fixed encodings, labels, selected KpqC/NIST KEM components, and the Quantum Superiority Fighter (QSF) component interface. Section 2.1 defines the exact measurement-interface scope and X-Wing/QSF background. The digital signature algorithm (DSA) rows implement Bindel et al.'s hybrid signature AND-combiner with label-bound component signatures [4]. Unlike prior embedded hybrid-PQC studies that mainly target Transport Layer Security (TLS)/X.509 stacks [6,15,32], this paper measures the primitive/harness layer and focuses on composed-cost attribution under one bare-metal Cortex-M4 harness.

Contributions.

1. **Composed-cost attribution for hybrid PQC.** We implement a common bare-metal Cortex-M4 harness for uniform Hash-CT benchmark rows and Bindel et al. hybrid signature AND-combiner rows, fixing byte layouts, labels, interfaces, backends, and target configuration so that classical-leg cost, wrapper work, and shared backend effects are visible.
2. **Backend sensitivity and observed signing-time variability.** We show that a Keccak-$f[1600]$ backend swap changes SHAKE-heavy rows while control rows remain unchanged, and report the observed variability of rejection-sampling signatures alongside their medians. We also report cycles, wire sizes, peak stack, code estimates, and standalone checks. These experiments are implementation-sensitivity analyses and are not claimed as phenomena unique to hybrid composition.
3. **SMAUG-T backend and optimization dilution.** We contribute a local Cortex-M4 SMAUG-T arithmetic backend and use it as a case study show-

ing that standalone speedups shrink within SMAUG-T+X25519 compositions because fixed classical, hashing, and wrapper work remain.

Section 2 defines the measured constructions and security scope. Section 3 describes the implementation and measurement setup. Section 4 presents the measurements and backend case studies, and Section 5 concludes.

2 Preliminaries and Related Work

2.1 Hash-CT QSF Benchmark Rows

Barbosa et al. proposed X-Wing, an ML-KEM-768/X25519 KEM built from the QSF combiner [2]. The Internet Engineering Task Force (IETF) draft specifies ML-KEM-768/X25519 X-Wing [12], and the X-Wing KEM team publishes a C implementation [34]. The X-Wing paper also studies a Hash-CT transcript that includes the post-quantum ciphertext. We use Hash-CT QSF benchmark rows only as a local measurement label for the uniform transcript of Equation (1), not as a new KEM construction, a new combiner, or a measurement of the IETF X-Wing transcript. Under one benchmark interface, the measured rows hash the post-quantum ciphertext together with the component shared secrets and classical transcript fields for ML-KEM, SMAUG-T, NTRU+, and HQC. In particular, the IETF ML-KEM-768/X25519 X-Wing transcript omits the ML-KEM ciphertext from the final combiner hash, whereas our Hash-CT rows include ct_{pq}. For large-ciphertext KEMs such as HQC, this adds an additional Hash-CT hashing cost relative to a no-ct_{pq} transcript. The DSA rows are separate and implement Bindel et al.'s hybrid signature AND-combiner as treated in Section 2.2.

For a selected algorithm pair, public keys, secret keys, and ciphertexts are fixed concatenations of the post-quantum and classical byte strings. The measured Hash-CT transcript computes the final shared secret as

$$ss = \mathsf{SHA3\text{-}256}(\mathit{label} \parallel ss_{\mathrm{pq}} \parallel ss_{\mathrm{cl}} \parallel ct_{\mathrm{pq}} \parallel ct_{\mathrm{cl}} \parallel pk_{\mathrm{cl}}), \tag{1}$$

where ss_{pq} and ss_{cl} are component shared secrets, ct_{pq} and ct_{cl} are component encapsulation tokens, and pk_{cl} is the long-term classical public key. The label has a fixed prefix and both component names, and fixed field lengths make parsing injective within each pair. The post-quantum public key is not hashed as a separate field, so adding it would define a different public-key-bound transcript.

Figure 1(a) shows the IETF X-Wing key-generation background. For the reported KeyGen measurements, the harness instead invokes the post-quantum and X25519 key-generation routines independently. Figure 1(b) shows the Hash-CT sender encapsulation measured in this work. During decapsulation, the receiver recovers both component shared secrets and reapplies Equation (1).

Security scope. The tables quantify costs for the Hash-CT benchmark rows above. The cited X-Wing/QSF work provides construction background for the component flow and interface. This paper does not derive fresh reductions for every substituted KpqC/NIST KEM component or claim side-channel resistance, fault resistance, or formal constant-time guarantees.

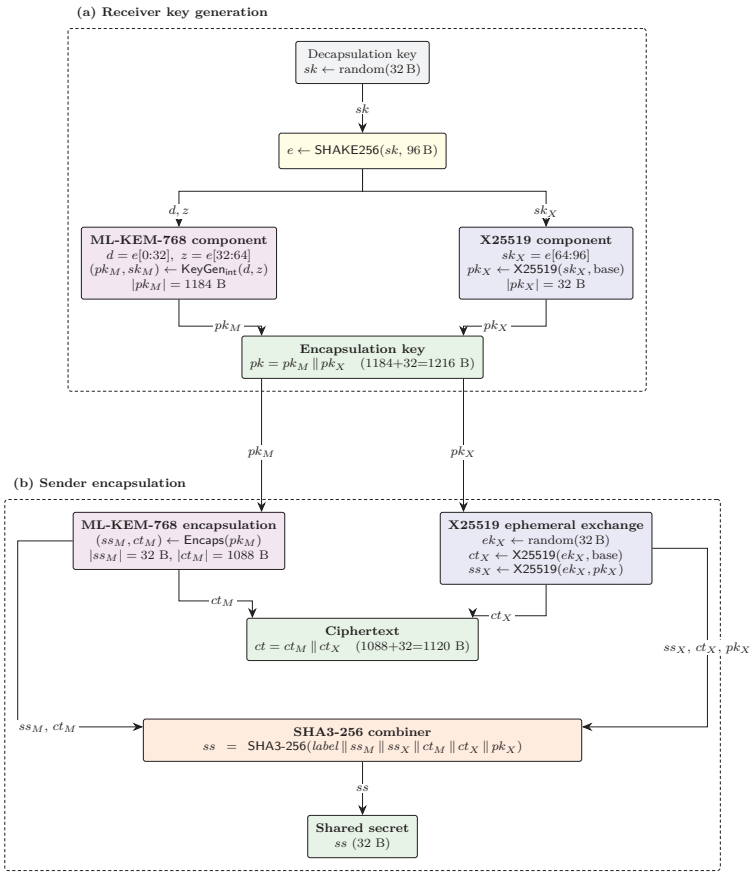
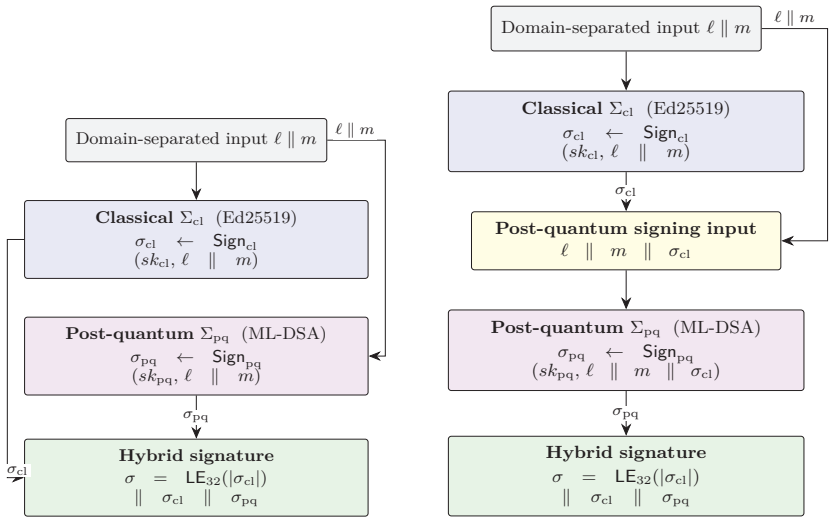


Fig. 1: X-Wing receiver key generation and Hash-CT sender encapsulation for ML-KEM-768/X25519.



(a) Concat mode: both components sign $\ell \parallel m$.

(b) Strong-nesting mode: the post-quantum component signs $\ell \parallel m \parallel \sigma_{\mathrm{cl}}$.

Fig. 2: Concat and strong-nesting hybrid-signature signing flows implemented in the harness.

2.2 Hybrid Signature Combiner

For signatures, we implement the AND-combiner of Bindel et al. [4], where the classical and post-quantum schemes sign label-bound inputs and verification accepts only if both component signatures verify. Let Σ_{cl} and Σ_{pq} denote the classical and post-quantum schemes. With fixed label ℓ, the primary concat mode is

$$\sigma_{\mathrm{cl}} \leftarrow \mathsf{Sign}_{\mathrm{cl}}(sk_{\mathrm{cl}}, \ell \parallel m), \qquad \sigma_{\mathrm{pq}} \leftarrow \mathsf{Sign}_{\mathrm{pq}}(sk_{\mathrm{pq}}, \ell \parallel m). \tag{2}$$

The hybrid signature is serialized with a 32-bit little-endian length prefix, denoted LE_{32}, as $\sigma = \mathsf{LE}_{32}(|\sigma_{\mathrm{cl}}|) \parallel \sigma_{\mathrm{cl}} \parallel \sigma_{\mathrm{pq}}$. The length prefix makes parsing unambiguous. A strong-nesting mode is retained only for overhead checks. In that mode, the post-quantum input is $\ell \parallel m \parallel \sigma_{\mathrm{cl}}$. If concat and nested modes are used in one protocol context, labels must be mode-specific.

2.3 Related Work

Related work falls into three groups. X-Wing, Starfighters, Request for Comments (RFC) 9794, NIST transition guidance, and Bindel et al. provide the hybrid-construction background [2,12,11,13,28,4]. Protocol-level studies measure Transport Layer Security (TLS)/X.509 PQC or hybrid costs, including TLS

based on ring learning with errors, Kyber/SPHINCS+ in mbed TLS, KEMTLS versus PQ-TLS, and KpqC in TLS/OpenSSL [5,6,15,32]. The closest KpqC-over-TLS study targets Cortex-A76 with OpenSSL, whereas we measure the bare-metal Cortex-M4 primitive/harness layer. Finally, pqm4, PQClean, and scheme-specific M4 work provide the implementation lineage [16,31,7,19,10,23]. We fix the KEM wrapper, classical leg, transcript layout, footprint accounting, and shared backend without adding a TLS stack. This makes backend sensitivity, observed signing-time variability, and optimization dilution visible below the protocol layer.

Ghinea et al. [14] evaluated an Elliptic Curve Digital Signature Algorithm (ECDSA)–Dilithium hybrid on an nRF52840 Cortex-M4F security key, including signing-time distributions and Client to Authenticator Protocol timeouts, thereby documenting rejection-sampling variability in an embedded hybrid setting. We do not claim this variability as a new observation. Our evaluation instead provides broader KpqC/NIST coverage under a common bare-metal harness with fixed interfaces, backends, classical partners, and target configuration.

3 Implementation and Experimental Setup

3.1 Implementation Design

The harness separates Hash-CT KEM orchestration from Bindel et al. hybrid signature orchestration. KEM wrappers call component key generation, encapsulation, and decapsulation, concatenate post-quantum (PQ) bytes before classical bytes, and apply the SHA3-256 Hash-CT transcript of Equation (1). Signature wrappers implement the concat AND-combiner and serialization defined in Section 2.2, with sizes taken from the algorithm descriptors.

Post-quantum components come from KpqC/NIST source trees, pqm4 [16], PQClean [31], and public Cortex-M4 repositories. We use optimized Cortex-M4 backends where available and identify portable or reference code when it remains on the measured path. Table 1 lists the measured backend choices and provenance references.

Artifact provenance. The artifact records source snapshots for the remaining components together with build flags, generated comma-separated value files and logs, stack and object-size outputs, and the scripts used to regenerate the tables and figures.

We contribute local arithmetic only for SMAUG-T. The backend replaces the reference C polynomial-multiplication path for the NTT-unfriendly SMAUG-T ring [9] with hand-written ARMv7-M Toom-Cook/Karatsuba code, and deterministic test-vector outputs match those of the reference backend. The kernel contains no explicit secret-dependent branches or table lookups, but no formal leakage analysis is claimed. Because prior Cortex-M4 SMAUG-T work exists [23], we treat this backend only as a case study of optimization dilution within a hybrid KEM stack.

The library uses static allocation where possible. Temporary wrapper buffers have compile-time maximum sizes. Scheme-owned code size is reported as

Table 1: Measured component backends and source lineage.

Component	Measured backend	References
KEM/DSA orchestration	Hash-CT KEM wrapper and Bindel et al. DSA AND-combiner wrapper/dispatch	[2,34,4]
Harness	pqm4 hardware abstraction layer (HAL) + SysTick harness	[16]
ML-KEM	PQClean/pqm4 `m4fspeed/m4fstack`	[24,31,16]
ML-DSA	PQClean/pqm4 `m4f`	[25,31,16]
FN-DSA	mupq/pqm4 Falcon/FN-DSA line	[29,30]
SPHINCS$^+$	PQClean portable scheme code + ARMv7-M SHA-256 or pqm4 Keccak	[26,31]
HQC	v5-compatible additive fast Fourier transform (FFT) M4 backend	[27,7]
SMAUG-T	KpqClean Ver2 + local `m4opt` polymul assembly	[17,9]
NTRU+	KpqClean Ver2 + public Cortex-M4 number-theoretic transform (NTT) backend	[17,10]
HAETAE	KpqClean Ver2 + Cortex-M4 backend	[17,8]
AIMer	KpqClean Ver2 port of streaming M4 backend	[17,19]
Classical curves	Lenngren X25519/P-256 assembly, TweetNaCl Ed25519 C, and mbedTLS-C ECDSA-P256	[21,20,22,3]
Symmetric	assembly Keccak-f[1600] + ARMv7-M SHA-256	[1,33]

Source commits: X-Wing `c696a197`, pqm4 `a24bb4b6`, mupq `ddccced`, and PQClean `c3e6861`. Standalone probes check that intended backend paths are linked.

an object-level estimate that sums the PQ `.text` sections and the measured classical-leg and combiner objects, while excluding shared Keccak, HAL, and harness code. These are therefore incremental code-size estimates, not complete linked firmware images.

3.2 Target, Toolchain, and Reproducibility

Measurements run on a bare-metal ARM Cortex-M4F using the pqm4 HAL. The primary campaign uses arm-none-eabi-gcc 13.2.1, assembly (ASM) Keccak-f[1600] unless stated otherwise, and mbedTLS 3.6.3 where no dedicated Cortex-M4 classical backend is available. It runs on the NUCLEO-L4R5ZI target at 20 MHz/0 flash wait-states with release build flags `-mcpu=cortex-m4 -mthumb -mfloat-abi=hard -mfpu=fpv4-sp-d16 -O3 -fomit-frame-pointer`. Cycle counts are the primary comparison metric. Functional validation includes component known-answer tests, hybrid shared-secret agreement, and SMAUG-T reference/ASM test-vector agreement.

3.3 Keccak Implementation Variants

The Keccak experiment compares the portable C Keccak-f[1600] permutation with the pqm4 ARMv7-M backend [1]. Only the innermost permutation changes. Absorption, squeezing, padding, arithmetic kernels, classical code, target configuration, and the harness stay fixed. Ratios therefore attribute cost to the shared Keccak permutation within this stack, rather than to changes in higher-level SHAKE/SHA3 code or arithmetic kernels.

3.4 Measurement Methodology

We measure processor cycles with the pqm4 HAL. The HAL extends the 24-bit SysTick counter with overflow tracking and returns 64-bit values. Primary KEM, HAETAE, ML-DSA, and FN-DSA measurements report $N = 100$ medians. For randomized or rejection-sampling operations, Table 5 also reports means and observed maximum-to-median ratios from the measurement records. These descriptive statistics do not support estimates of latency percentiles, exceedance probabilities, or worst-case execution time.

AIMer and SPHINCS$^+$ use smaller repetition counts, as stated in the corresponding table notes. Those rows support order-of-magnitude and backend-attribution comparisons. Variability comparisons are limited to the $N = 100$ rows.

4 Bare-Metal Cortex-M4 Evaluation

This section reports NUCLEO-L4R5ZI measurements at 20 MHz/0 flash wait-states (0 WS). The Cortex-M4F target provides 640 KB static random-access memory (SRAM) and 2 MB flash. All timing values are cycle counts obtained with the backends in Table 1 unless stated otherwise.

4.1 Hybrid KEM Results

Table 2 reports the main Hash-CT X25519 benchmark configuration with $N = 100$ medians. The KeyGen column uses the independent component calls described in Section 2.1. Total denotes the summed KeyGen, Encaps, and Decaps medians, not single-party or end-to-end protocol latency. ML-KEM, NTRU+, and SMAUG-T Total values are 3,719,838–8,770,987 cycles, whereas HQC Total values are 19,637,683–74,467,370 cycles with the v5-compatible additive-FFT backend [7]. These values include ct_{pq} hashing in the benchmark transcript, so large-ciphertext rows such as HQC include an additional Hash-CT hashing cost relative to a no-ct_{pq} transcript. The results characterize this benchmark interface and do not supply a security analysis for each substituted KEM.

Standalone measurements give 0.616/1.230/0.615 M cycles for X25519 key generation, encapsulation, and decapsulation and 5.94/11.92/5.99 M cycles for Ed25519 key generation, signing, and verification. Replacing X25519 with P-256 changes all twelve KEM Total values by at most 0.347% (per-row ratios stay within 0.997–1.000×) and does not change the family ordering. Deterministic concat-to-nested checks change representative signature rows by at most 0.35% and confirm backend selection and wrapper implementation.

4.2 Hybrid DSA Results and Observed Variability

Signature measurements use the Ed25519 concat baseline unless stated otherwise. Table 3 gives the $N = 100$ HAETAE, ML-DSA, and FN-DSA rows used

Table 2: Hash-CT hybrid KEM measurements on Cortex-M4.

X25519, 20 MHz/0 WS, $N = 100$ medians. Total is the sum of the three displayed operation medians and is not protocol latency. Cycles include ct_{pq} hashing.

PQ-KEM	KeyGen	Encaps	Decaps	Total
SMAUG-T1	1,210,134	1,820,980	1,412,908	4,444,022
SMAUG-T3	1,836,280	2,439,134	2,128,255	6,403,669
SMAUG-T5	2,606,616	3,189,011	2,975,360	8,770,987
NTRU+768	1,043,984	1,646,596	1,054,290	3,744,870
NTRU+864	1,156,249	1,753,290	1,175,628	4,085,167
NTRU+1152	1,314,452	1,910,138	1,342,069	4,566,659
ML-KEM-512	1,010,490	1,643,740	1,065,608	3,719,838
ML-KEM-768	1,258,669	1,909,217	1,342,781	4,510,667
ML-KEM-1024	1,635,681	2,281,523	1,728,766	5,645,970
HQC-128	3,596,009	6,536,578	9,505,096	19,637,683
HQC-192	8,854,451	16,036,945	23,970,881	48,862,277
HQC-256	13,802,173	24,020,196	36,645,001	74,467,370

for the main comparison, while Table 4 keeps low-repetition AIMer/SPHINCS$^+$ scale-only rows separate. ML-DSA has the lowest measured signing latency among these lattice schemes, HAETAE shows rejection-sampling variability, and FN-DSA combines fast verification with expensive key generation. Because several key-generation and signing paths are randomized, Table 5 reports descriptive variability summaries for the same set instead of relying only on medians.

Table 3: Primary hybrid DSA measurements on Cortex-M4.

PQ + Ed25519, concat mode, 20 MHz/0 WS. Values are cycle-count medians over $N = 100$ runs.

PQ-DSA	KeyGen	Sign	Verify
ML-DSA-44	7,364,995	15,037,912	7,563,684
ML-DSA-65	8,460,040	17,092,257	8,559,121
ML-DSA-87	10,220,574	18,087,747	10,316,385
HAETAE2	12,028,569	31,801,353	6,943,470
HAETAE3	15,915,932	36,271,144	7,682,232
HAETAE5	19,076,721	41,939,790	8,116,242
FN-DSA-512	75,087,719	34,401,102	6,545,554
FN-DSA-1024	263,700,870	60,320,802	6,921,437

Observed signing-time variability. Across the $N = 100$ measurements, HAETAE5 has a lower signing median than FN-DSA-1024 (41.94 versus 60.32 M cycles), but its observed maximum reaches 189.8 M cycles (4.53× its median). The FN-DSA-1024 maximum is 60.58 M cycles, less than 1% above its median. HAETAE has a wider observed signing-time range than ML-DSA and FN-DSA, but these sample maxima do not determine exceedance probabilities, latency percentiles, or worst-case execution times.

Table 4: Additional low-repetition hybrid DSA measurements.

PQ + Ed25519, concat mode, 20 MHz/0 WS. AIMer192f/256s use $N = 10$, AIMer128f is a single measurement, and SPHINCS$^+$ uses $N = 5$. These rows support order-of-magnitude comparisons only.

PQ-DSA	**KeyGen**	**Sign**	**Verify**
AIMer128f (m4opt)	6,404,117	47,935,060	32,960,014
AIMer192f (m4fast)	7,181,956	89,527,934	78,413,565
AIMer256s (m4stack)	10,190,847	1,971,649,069	1,550,023,748
SPHINCS$^+$-SHA2-128f	20,670,562	356,674,443	27,764,750
SPHINCS$^+$-SHA2-128s	948,458,267	7,170,561,799	13,657,848
SPHINCS$^+$-SHA2-256f	64,504,322	1,309,374,104	42,139,178
SPHINCS$^+$-SHAKE-128f	56,181,392	1,188,091,067	75,088,359
SPHINCS$^+$-SHAKE-256f	204,948,947	4,015,467,171	114,616,346
SPHINCS$^+$-SHAKE-192s	4,774,330,073	42,882,388,698	41,997,450

Table 5: Observed key-generation and signing variability ($N = 100$).

Ed25519 concat mode, 20 MHz/0 WS.

Scheme	**KeyGen**			**Sign**		
	Median (M cycles)	**Mean (M cycles)**	**Max/ median**	**Median (M cycles)**	**Mean (M cycles)**	**Max/ median**
HAETAE2	12.03	13.81	3.12×	31.80	37.22	4.62×
HAETAE3	15.92	18.04	2.71×	36.27	41.32	3.06×
HAETAE5	19.08	23.00	5.03×	41.94	51.23	4.53×
ML-DSA-44	7.36	7.37	1.00×	15.04	15.77	1.82×
ML-DSA-65	8.46	8.46	1.00×	17.09	17.87	1.85×
ML-DSA-87	10.22	10.22	1.00×	18.09	19.38	1.86×
FN-DSA-512	75.09	81.50	2.76×	34.40	34.40	1.01×
FN-DSA-1024	263.70	313.98	3.15×	60.32	60.33	1.00×

4.3 Keccak and Hash Backend Effects

The Keccak experiment changes only Keccak-f[1600]. Table 6 lists representative rows from the Keccak-isolation campaign. Control rows (SPHINCS$^+$-SHA2 and FN-DSA) remain at 1.00×, supporting attribution to the Keccak permutation rather than a global build change. Speedups reach 2.16× for SPHINCS$^+$-SHAKE-256f signing and 1.65× for AIMer192s, versus 1.18–1.51× for lattice rows and 1.10× for HQC-128. This larger effect on SHAKE-heavy signatures is not unique to hybrid composition and serves here as a backend-control experiment under the common harness.

4.4 SMAUG-T Local Backend and Optimization Dilution

Table 7 isolates the local SMAUG-T arithmetic backend. Standalone gains of 1.79–1.82× shrink to 1.42–1.65× with X25519 because non-PQ work remains fixed. The 2.56 M-cycle residual is 58%, 40%, and 29% of the primary X25519 total for T1/T3/T5.

Table 6: Keccak-f[1600] backend attribution.

This separate matched campaign need not reproduce the primary medians in Table 2. Speedup is the reference C cycle count divided by the ARM assembly cycle count. All non-permutation code and measurement settings are unchanged.

Class	Row	C-ref	ASM	Speedup
Lattice KEM	ML-KEM-1024 Total	8,513,176	5,645,764	1.51×
HQC	HQC-128 Total	21,575,619	19,647,266	1.10×
ML-DSA	ML-DSA-87 sign	23,042,134	18,087,747	1.27×
HAETAE	HAETAE5 sign	63,096,274	41,939,790	1.50×
AIMer	AIMer192s (m4stack) sign	1,541,036,168	934,921,177	1.65×
SPHINCS$^+$-SHAKE	SPHINCS$^+$-SHAKE-256f sign	8,663,447,709	4,015,467,171	2.16×
Control row	SPHINCS$^+$-SHA2-128f sign	356,601,179	356,674,443	1.00×
Control row	FN-DSA-512 sign	34,392,093	34,401,102	1.00×

Table 7: Standalone and composed costs of the local SMAUG-T backend.

Matched run with C Keccak. Residual is Hybrid ASM − Stand ASM in millions of cycles. Residual share is this value divided by the corresponding primary X25519 total with assembly Keccak.

Scheme	Stand C	Stand ASM	Hybrid C	Hybrid ASM	Hybrid speedup	Residual	Residual share
SMAUG-T1	5,093,516	2,811,333	7,653,239	5,371,074	1.42×	2.56	58%
SMAUG-T3	10,352,582	5,788,205	12,912,889	8,348,500	1.55×	2.56	40%
SMAUG-T5	16,834,070	9,226,662	19,395,083	11,787,708	1.65×	2.56	29%

Table 8: Wire size and implementation footprint of representative compositions.

All values are bytes. Code estimates exclude shared Keccak/HAL/harness. The final line gives representative linked-harness flash and memory.

Hybrid	Type	Public key	Ciphertext/ signature	Peak stack	Code estimate
ML-KEM-768 + X25519	KEM	1,216	1,120	6,748	20,460
SMAUG-T3 + X25519	KEM	1,120	1,024	18,336	15,988
HQC-128 + X25519	KEM	2,273	4,465	53,600	216,335
ML-DSA-44 + Ed25519	DSA	1,344	2,488	45,216	29,288
HAETAE2 + Ed25519	DSA	1,024	1,542	83,468	39,048
AIMer128f + Ed25519	DSA	64	5,956	11,952	22,820
SPHINCS$^+$-SHAKE-128f + Ed25519	DSA	64	17,156	18,860	13,880

Full linked harness flash/memory (text+data / data+bss) is 1,024,416/12,312 B for KEM rows and 643,484/23,852 B for DSA rows.

4.5 Size and Memory Considerations

Table 8 combines wire size and implementation footprint. The largest reported stack watermarks—83,468 B for HAETAE2+Ed25519 signing and 53,600 B for HQC-128+X25519 decapsulation—fit the target's 640 KB SRAM but do not directly generalize to smaller platforms.

Limitations. The campaign uses one NUCLEO-L4R5ZI board, one Cortex-M4 backend stack, and the 20 MHz/0 WS setting. It measures primitive/harness-layer timing and footprint. It does not evaluate TLS/X.509 integration, energy, resistance to side-channel or fault attacks, or worst-case execution time. The Hash-CT rows are not IETF X-Wing deployment costs. The $N = 100$ signing summaries are descriptive and do not establish protocol latency or tail probabilities.

5 Conclusion

Using one bare-metal ARM Cortex-M4 harness, we measured KpqC/NIST hybrid KEM benchmark rows with X25519 under a uniform Hash-CT transcript and Bindel et al. hybrid signature AND-combiner rows. The resulting baseline supports composed-cost attribution across the measured families.

The main lesson is that composed hybrid benchmarks behave differently from primitive-only rankings. Shared backends affect SHAKE-heavy signatures, median-only tables hide observed variability, and fixed costs from classical operations, hashing, and wrapper logic dilute standalone optimization gains. These results are presented as controls and case studies and are not claimed to be unique to hybrid composition.

Hybrid PQC reports should document backends and sample counts and evaluate optimizations within the full composition.

6 Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2025-02306395, Development and Demonstration of PQC-Based Joint Certificate PKI Technology, 25%) and this work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25394739, Development of Security Enhancement Technology for Industrial Control Systems Based on S/HBOM Supply Chain Protection, 25%) and this work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25519543, Development of PQC Migration Technology for Quantum-Safe IC Chip, 25%) and this work was supported by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2020-II201797, 25%).

References

1. Adomnicai, A.: An update on keccak performance on armv7-m. Cryptology ePrint Archive (2023)

2. Barbosa, M., Connolly, D., Duarte, J.D., Kaiser, A., Schwabe, P., Varner, K., Westerbaan, B.: X-wing: The hybrid kem you've been looking for. Cryptology ePrint Archive (2024)
3. Bernstein, D.J., Van Gastel, B., Janssen, W., Lange, T., Schwabe, P., Smetsers, S.: Tweetnacl: A crypto library in 100 tweets. In: International Conference on Cryptology and Information Security in Latin America. pp. 64–83. Springer (2014)
4. Bindel, N., Herath, U., McKague, M., Stebila, D.: Transitioning to a quantum-resistant public key infrastructure. In: International Workshop on Post-Quantum Cryptography. pp. 384–405. Springer (2017)
5. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the tls protocol from the ring learning with errors problem. In: 2015 IEEE symposium on security and privacy. pp. 553–570. IEEE (2015)
6. Bürstinghaus-Steinbach, K., Krauß, C., Niederhagen, R., Schneider, M.: Post-quantum tls on embedded systems: Integrating and evaluating kyber and sphincs+ with mbed tls. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. pp. 841–852 (2020)
7. Chen, M.S., Chien, T.Y., Chiu, C.M., Lin, H.H., Peng, C.T., Yang, B.Y.: Additive ffts for hqc on arm cortex-m4, revisited. Cryptology ePrint Archive (2026)
8. Cheon, J.H., Choe, H., Devevey, J., Güneysu, T., Hong, D., Krausz, M., Land, G., Möller, M., Stehlé, D., Yi, M.: Haetae: Shorter lattice-based fiat-shamir signatures. IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(3), 25–75 (2024)
9. Cheon, J.H., Choe, H., Seo, J., Seong, H.: Smaug (-t), revisited: Timing-secure, more compact, less failure. IEEE Access **12**, 188386–188397 (2024)
10. Cho, S.: NTRU+ KEM on ARM Cortex-M4: Optimized implementation and benchmarks. GitHub repository, https://github.com/SuBeen-Cho/Optimization_CortexM4_NTRUplus (2026)
11. Connolly, D., Hövelmanns, K., Hülsing, A., Kousidis, S., Meijers, M.: Starfighters—on the general applicability of x-wing. Cryptology ePrint Archive (2025)
12. Connolly, D., Schwabe, P., Westerbaan, B.: X-wing: General-purpose hybrid post-quantum kem. Internet-Draft draft-connolly-cfrg-xwing-kem-10, Internet Engineering Task Force (Mar 2026), https://datatracker.ietf.org/doc/draft-connolly-cfrg-xwing-kem/, version 10, 2 March 2026
13. Driscoll, F., Parsons, M., Hale, B.: Terminology for post-quantum traditional hybrid schemes. RFC 9794, IETF (2025)
14. Ghinea, D., Kaczmarczyck, F., Pullman, J., Cretin, J., Kölbl, S., Misoczki, R., Picod, J.M., Invernizzi, L., Bursztein, E.: Hybrid post-quantum signatures in hardware security keys. In: Applied Cryptography and Network Security Workshops. Lecture Notes in Computer Science, vol. 13907, pp. 480–499. Springer (2023). https://doi.org/10.1007/978-3-031-41181-6_26
15. Gonzalez, R., Wiggers, T.: Kemtls vs. post-quantum tls: Performance on embedded systems. In: International Conference on Security, Privacy, and Applied Cryptography Engineering. pp. 99–117. Springer (2022)
16. Kannwischer, M.J., Rijneveld, J., Schwabe, P., Stoffelen, K.: pqm4: Testing and benchmarking nist pqc on arm cortex-m4. NIST Second Post-Quantum Cryptography Standardization Conference (2019)
17. KpqC-cryptocraft: KpqClean Ver2: Reference implementations of the KpqC standardized algorithms. GitHub repository, https://github.com/kpqc-cryptocraft/KpqClean_ver2 (2026)

18. KpqC Organizing Committee: Selected algorithms from the KpqC competition round 2. https://kpqc.or.kr/competition_02.html (2025), final algorithms announced 16 January 2025, accessed 2026-06-26
19. Kwon, J., Lee, S., Lee, B., Seo, H., Cho, J.: Efficient implementations of aimer post-quantum signature scheme for low-end to high-end iot devices. IEEE Internet of Things Journal (2025)
20. Lenngren, E.: P256-Cortex-M4: P-256 ECDSA/ECDH for Cortex-M4. https://github.com/Emill/P256-Cortex-M4
21. Lenngren, E.: X25519-Cortex-M4. https://github.com/Emill/X25519-Cortex-M4
22. Mbed TLS Team: Mbed TLS. GitHub repository, https://github.com/Mbed-TLS/mbedtls (2024)
23. Narlı, O., Pehlivanoğlu, M.K., Akleylek, S.: A new optimized implementation of smaug-t for lightweight devices. In: Nordic Conference on Secure IT Systems. pp. 77–90. Springer (2025)
24. National Institute of Standards and Technology: FIPS 203: Module-lattice-based key-encapsulation mechanism standard (ML-KEM). Federal information processing standards publication, NIST (Aug 2024)
25. National Institute of Standards and Technology: FIPS 204: Module-lattice-based digital signature standard (ML-DSA). Federal information processing standards publication, NIST (Aug 2024)
26. National Institute of Standards and Technology: FIPS 205: Stateless hash-based digital signature standard (SLH-DSA). Federal information processing standards publication, NIST (Aug 2024)
27. National Institute of Standards and Technology: NIST IR 8545: Status report on the fourth round of the NIST post-quantum cryptography standardization process. Tech. rep., NIST (Mar 2025), HQC selected for standardization
28. National Institute of Standards and Technology: Recommendations for key-encapsulation mechanisms. NIST Special Publication 800-227, NIST (Sep 2025). https://doi.org/10.6028/NIST.SP.800-227
29. Perlner, R.: FIPS 206 status update: FN-DSA (Falcon). Presentation, 6th NIST PQC Standardization Conference (2025), nIST
30. Pornin, T.: Falcon on ARM Cortex-M4: An update. Cryptology ePrint Archive, Report 2025/123 (2025)
31. PQClean Contributors: PQClean: Clean, portable, tested implementations of post-quantum cryptography. https://github.com/PQClean/PQClean
32. Sim, M., Song, G., Eum, S., Lee, M., Yoon, S., Baksi, A., Seo, H.: Integrating and benchmarking kpqc in tls/x. 509. Electronics **14**(18), 3717 (2025)
33. Weatherley, R.: ARMv7-M SHA-256 assembly. GitHub repository, https://github.com/rweather/lwc-finalists
34. X-Wing KEM Team: X-Wing C implementation. GitHub repository, https://github.com/x-wing-kem-team/xwing (2024), commit c696a197, accessed 2026-06-27

Latency Analysis of ML-KEM and SMAUG-T in ROS 2

Sungwoo Yun[1[0009−0005−6333−1805]], Jung-Hyun Mok[1[0009−0008−4166−8118]], Jimin Park[1[0009−0005−8783−7958]], and Sokjoon Lee[1[0000−0002−4632−8322]]

Gachon University, Seongnam-si, Gyeonggi-do, Republic of Korea
{borok2311, johnmok, jimin030907, junny}@gachon.ac.kr

Abstract. ROS 2 adopts DDS as its standard middleware, and DDS Security establishes a shared secret between communicating participants through a handshake based on classical public-key cryptography. As quantum computing advances, this key-exchange mechanism needs to transition to a post-quantum cryptographic structure. In this paper, we replace the DH-based key exchange of the DDS Security handshake with a KEM and compare the NIST standard ML-KEM-512 with SMAUG-T TiMER, a KEM selected in the Korean Post-Quantum Cryptography (KpqC) competition, within the same DDS Security environment. Because the identity credential, permissions document, and digital signature fields remain identical regardless of the algorithm, the message-size difference between the two algorithms arises from the difference in public-key and ciphertext sizes. The experiments were conducted on two Raspberry Pi 4B boards running ROS 2 and CycloneDDS, and handshake latency was measured under four controlled wireless conditions: an unconstrained Wi-Fi baseline, a 20 Mbps bandwidth-limited condition, a 5% packet-loss condition, and a combined condition applying both a 20 Mbps limit and 5% loss. The 20 Mbps and 5% loss settings are not intended to represent all ROS 2 deployments; rather, they are used as controlled impairment conditions to analyze how the increased handshake message size introduced by PQC is reflected in latency when the transmission overhead is no longer negligible. The experimental results show that the KEM computation time is small relative to the overall handshake latency, and that the difference in KEM public-key and ciphertext sizes can be reflected in the median handshake latency under controlled constrained-link conditions.

Keywords: Post-quantum cryptography · Key encapsulation mechanism · ROS 2 · DDS Security

1 Introduction

ROS 2 is an open-source framework for developing robotic software across diverse domains such as drones, autonomous vehicles, and smart factories [6]. ROS 2 follows a distributed architecture in which multiple nodes exchange sensor measurements and control commands, and it uses DDS [8] as its standard

middleware for inter-node communication. However, default DDS communication is designed primarily for data distribution and real-time transmission, and security features such as authentication and encryption are not enabled unless DDS Security is separately configured. Because the data exchanged in ROS 2 may include information directly related to system safety, such as sensor readings and actuation commands, a security mechanism that prevents tampering and eavesdropping during communication is essential. To meet this requirement, the Object Management Group (OMG) established DDS Security [8], a security extension standard that provides authentication, encryption, and access-control functionality. DDS Security performs a handshake based on classical public-key cryptography to authenticate communicating participants and establish a shared secret key. However, because this handshake relies on public-key schemes such as RSA and DH, its long-term security may be affected by advances in quantum computing. In particular, Shor's algorithm [11] can solve the underlying mathematical problems of these schemes, including integer factorization and the discrete logarithm problem. Consequently, recent studies have explored integrating quantum-resistant KEMs into DDS Security, centered on the NIST standard ML-KEM [7]. Meanwhile, SMAUG-T, selected in the Korean Post-Quantum Cryptography (KpqC) competition [5], offers smaller public-key and ciphertext sizes than ML-KEM and provides the TiMER parameter set for lightweight IoT environments [3]. However, SMAUG-T has not been sufficiently evaluated in the DDS Security context. In addition, it has not been clearly analyzed how the difference in public-key and ciphertext sizes between KEM algorithms affects DDS Security handshake latency under bandwidth-constrained and lossy-link conditions. Therefore, in this paper, we integrate ML-KEM-512 and SMAUG-T TiMER into the DDS Security handshake and compare the two algorithms in the same CycloneDDS-based ROS 2 environment. This paper focuses on how the KEM public-key and ciphertext sizes affect the median DDS Security handshake latency under controlled wireless impairment conditions. Specifically, we evaluate the two algorithms under four controlled conditions: an unconstrained Wi-Fi baseline, a 20 Mbps bandwidth-limited condition, a 5% packet-loss condition, and a combined condition applying both a 20 Mbps limit and 5% loss.

Related Work. Several studies have examined applying post-quantum cryptography to DDS Security. Blanco-Romero et al. [1] proposed PQSec-DDS, a library for integrating PQC into DDS Security. Built on liboqs, the library supports Kyber and Dilithium and is designed to be usable across multiple DDS implementations, including CycloneDDS and OpenDDS. In addition, Safdar et al. [10] replaced DH key exchange with ML-KEM in the CycloneDDS DDS Security handshake using the OQS-provider. By evaluating the full handshake, including certificates and digital signatures, in terms of time and size on a Raspberry Pi 5, they demonstrated the feasibility of PQC-based DDS Security in ROS 2 environments. However, existing studies on DDS Security have focused primarily on NIST standard algorithms, and SMAUG-T has not yet been applied to or evaluated in DDS Security. Moreover, existing evaluations were conducted

under fixed network conditions, leaving insufficient analysis of how the difference in KEM public-key and ciphertext sizes is reflected in actual DDS Security handshake latency under controlled bandwidth-limited and packet-loss conditions. This is an important issue because quantum-resistant key exchange can increase the handshake message size, and a post-quantum handshake with larger data may affect transmission performance in real network environments [4].

Our Contributions. The contributions of this paper are as follows.

- We replace the DH-based key exchange of the DDS Security handshake with a KEM-based structure while preserving the existing identity-credential verification, permission checking, nonce-based challenge, and digital-signature verification procedures.
- We implement ML-KEM-512 and SMAUG-T TiMER using a native KEM backend in the same CycloneDDS-based ROS 2 DDS Security environment, and compare the KEM computation time and handshake latency of the two algorithms along the same implementation path.
- We evaluate the median DDS Security handshake latency under four controlled wireless conditions: an unconstrained Wi-Fi baseline, a 20 Mbps bandwidth-limited condition, a 5% packet-loss condition, and a combined condition applying both a 20 Mbps limit and 5% loss.
- We analyze how the difference in KEM public-key and ciphertext sizes is reflected in the median handshake latency under controlled constrained-link conditions in which the transmission overhead is no longer negligible.

2 Background

2.1 Post-Quantum Cryptography and Key Encapsulation Mechanisms

Post-quantum cryptography refers to cryptographic techniques designed to remain secure even against attacks using quantum computers. From the standardization and deployment perspective, post-quantum primitives are mainly considered in the form of KEMs and digital signatures. Among them, a KEM is an asymmetric scheme for securely agreeing on a symmetric key between communicating parties, and it consists of three operations. Key generation produces a public-key and private-key pair; encapsulation uses the recipient's public key to generate a shared secret and a ciphertext that encapsulates it; and decapsulation uses the private key to recover the shared secret from the ciphertext. The shared secret established in this way is subsequently used as a session key for symmetric-key cryptography. In this paper, we select ML-KEM [7], designated as a NIST standard, and SMAUG-T [5], selected in the Korean Post-Quantum Cryptography (KpqC) competition, as the comparison targets.

ML-KEM. ML-KEM is a module-lattice-based KEM standardized by NIST as FIPS 203 and derived from CRYSTALS-Kyber [2, 7]. In this study, ML-KEM-512 is used as the NIST-standardized baseline at security level 1.

SMAUG-T. SMAUG-T is a lattice-based KEM selected in the Korean Post-Quantum Cryptography competition [5, 3]. Its TiMER parameter set targets lightweight environments and provides smaller public-key and ciphertext sizes than ML-KEM-512, as summarized in Table 1.

Table 1. Key-exchange field sizes of the classical baseline and the two KEMs (B). For ECDH P-256, the second field denotes the peer public key rather than a ciphertext.

Algorithm	Public key(pk)	Ciphertext(ct)	pk + ct
ECDH P-256 (baseline)	65	65	130
ML-KEM-512	800	768	1,568
SMAUG-T TiMER	672	608	1,280

2.2 DDS Security Handshake

Figure 1 shows the three-way handshake procedure of DDS Security.

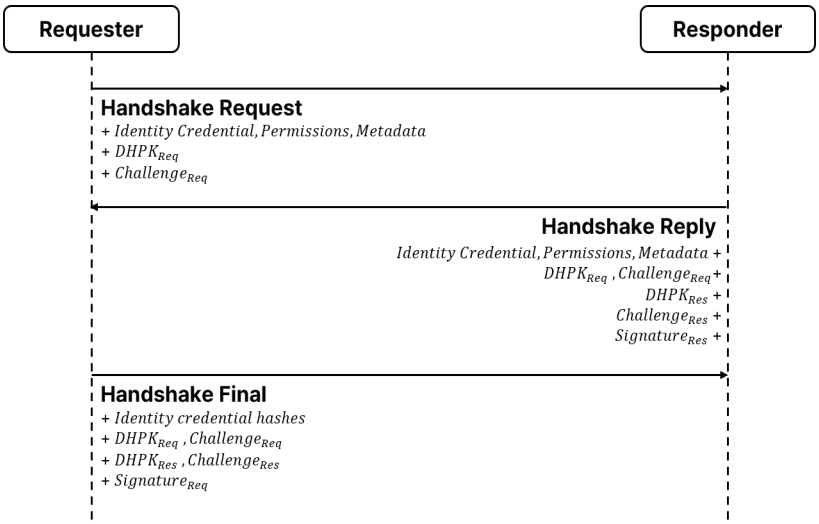


Fig. 1. The DDS Security three-way handshake procedure.

DDS Security performs a three-way handshake between a requester and a responder to authenticate participants and establish a shared secret, as shown in Fig. 1. The requester sends a HandshakeRequest containing its identity credential, permissions, participant metadata, a DH public key $DHPK_{\text{req}}$, and a fresh

challenge $Challenge_{\text{req}}$. The responder verifies the requester, generates its own DH key pair and challenge, computes the shared secret, and returns a HandshakeReply containing $DHPK_{\text{res}}$, $Challenge_{\text{res}}$, echoed requester values, and $Signature_{\text{res}}$.

After verifying the responder and the echoed values, the requester computes the same shared secret and sends a HandshakeFinal containing the relevant credential hashes, DH public keys, challenges, and $Signature_{\text{req}}$. The responder verifies the final message, after which both participants derive the session key from the established shared secret and challenge values.

3 Proposed Structure

In this paper, we propose a structure that replaces only the DH-based key exchange part of the DDS Security handshake with a KEM, while preserving identity-credential verification, permission checking, challenge-based freshness, and proof of possession through digital signatures. Figure 2 shows the proposed structure in which the DH public-key exchange of the original handshake is replaced with KEM-based key establishment.

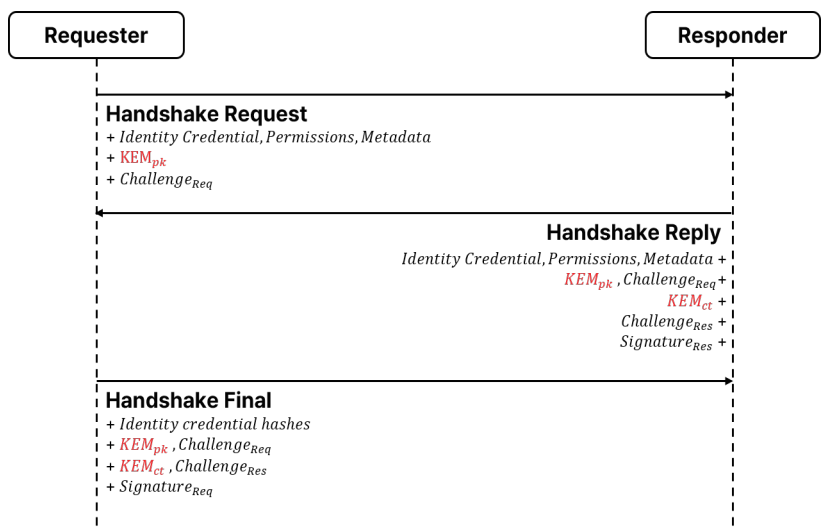


Fig. 2. The proposed KEM-based DDS Security handshake structure.

In the proposed structure, $DHPK_{\text{req}}$ in the HandshakeRequest is replaced with the KEM public key KEM_{pk}. Similarly, the DH public key $DHPK_{\text{res}}$ in the HandshakeReply is replaced with the KEM ciphertext KEM_{ct}, which is generated by the responder through encapsulation using KEM_{pk}.

The requester first generates a KEM key pair and sends KEM_{pk} together with its identity credential, permissions document, participant metadata, and challenge value $Challenge_{\mathrm{req}}$ in the HandshakeRequest. After verifying the requester, the responder performs encapsulation using KEM_{pk} and obtains both KEM_{ct} and the shared secret. The responder then sends KEM_{ct}, $Challenge_{\mathrm{res}}$, the echoed KEM_{pk} and $Challenge_{\mathrm{req}}$, and $Signature_{\mathrm{res}}$ in the HandshakeReply.

After receiving the HandshakeReply, the requester verifies the responder's identity credential, permissions document, echoed values, and $Signature_{\mathrm{res}}$. The requester then decapsulates KEM_{ct} using its KEM private key and obtains the same shared secret. The HandshakeFinal message binds the previous handshake transcript by including the identity-credential hashes, KEM_{pk}, KEM_{ct}, $Challenge_{\mathrm{req}}$, $Challenge_{\mathrm{res}}$, and $Signature_{\mathrm{req}}$.

In our implementation, the identity credential and handshake digital signatures retain the existing classical signature scheme based on ECDSA P-256 with SHA-256. Only the DH-family key-exchange fields are replaced with KEM_{pk} and KEM_{ct}. Accordingly, the handshake message-size estimation does not include any PQC digital-signature size. The resulting total size of each handshake message and the corresponding number of fragments under a 1500 B MTU are listed in Table 2.

Table 2. Total handshake message sizes and MTU-based fragmentation (B). The classical ECDH P-256 handshake is included as a transmission-level baseline. Fragment counts assume a 1500 B MTU.

Message	ECDH P-256	ML-KEM-512	SMAUG-T TiMER	Frag
HandshakeRequest	3,777	4,512	4,384	3 / 4 / 3
HandshakeReply	3,274	4,712	4,424	3 / 4 / 4
HandshakeFinal	234	1,672	1,384	1 / 2 / 1
Total	7,285	10,896	10,192	7 / 10 / 8

Because the transmission of the HandshakeFinal is not included in the measured handshake latency, the relevant transmission path consists of the HandshakeRequest and HandshakeReply. Along this measured path, ML-KEM-512 transmits 9,224 B, whereas SMAUG-T TiMER transmits 8,808 B. Under the applied MTU-based fragmentation, these messages correspond to eight and seven fragments, respectively. Thus, although the measured-path size difference is 416 B, it results in one additional fragment for ML-KEM-512.

3.1 Security Considerations.

The proposed modification changes only the key-establishment primitive; identity credentials, permissions, challenges, and ECDSA-based transcript signatures are retained. The KEM public key and ciphertext are included in the authenticated handshake transcript, while fresh challenges are generated for each hand-

shake. A formal security analysis of the modified protocol is outside the scope of this work.

4 Experimental Setup

4.1 Experimental Environment

Two Raspberry Pi 4B boards were used as the requester (Req) and responder (Res) to evaluate ML-KEM-512 and SMAUG-T TiMER within the proposed CycloneDDS-based DDS Security handshake. The hardware, software, and measurement configuration is summarized in Table 3.

Table 3. Experimental environment.

Item	Description
Device	Raspberry Pi 4B × 2
OS / Kernel	Ubuntu 24.04 (aarch64), kernel 6.8.0-1057-raspi
Network	Single shared Wi-Fi network
Middleware	ROS 2 Jazzy + CycloneDDS (ROS_DOMAIN_ID = 77)
KEM algorithms	ML-KEM-512, SMAUG-T TiMER
Network impairment	tc/netem applied to the wlan0 transmit direction
Clock source	CLOCK_MONOTONIC, requester side only
Repetition	1,005 runs per condition (first 5 discarded as warm-up)
Inter-run interval	0.1 s

For a fair comparison, both algorithms were integrated through the same native KEM interface in the DDS Security authentication plugin rather than using the OpenSSL provider path. ML-KEM-512 was implemented using liboqs, whereas SMAUG-T TiMER used its reference implementation; thus, the interface and invocation path were shared, although the underlying implementations differed.

4.2 Network Conditions

Four network conditions were configured using `tc/netem`, as summarized in Table 4. Native used the default `fq_codel` qdisc without artificial impairment. BW-only limited bandwidth to 20 Mbps using `htb` with a `netem` leaf qdisc, while Loss-only applied 5% independent packet loss [9]. BW+Loss combined the same bandwidth limit and loss rate. No artificial delay or jitter was introduced, and the impairments were applied identically to the transmit direction of both Req and Res.

4.3 Experimental Procedure and Measurement Metrics

Each experimental unit is defined as a combination of one KEM algorithm and one network condition. In each unit, Res was launched first, followed by Req, so

Table 4. Network conditions.

Condition	Bandwidth	Loss	Purpose
Native	None	0%	Unconstrained laboratory Wi-Fi baseline
BW-only	20 Mbps	0%	Effect of bandwidth limitation
Loss-only	None	5%	Effect of packet loss
BW+Loss	20 Mbps	5%	Effect of bandwidth limit & packet loss

that DDS Discovery and the DDS Security handshake were carried out. Req was run for eight seconds per execution and then terminated, and this was repeated 1,005 times under each condition. During the experiments, packets were captured on both the Req and Res sides, and the KEM computation time and per-stage handshake times were recorded inside the DDS Security authentication plugin. To prevent slow drift of the wireless environment from being attributed to the algorithm, the two algorithms were measured in an interleaved manner within each network condition rather than in separate consecutive blocks.

We define the handshake latency as the interval, measured on the requester side, between the instant at which the requester hands the HandshakeRequest message to the transport layer and the instant at which it hands the HandshakeFinal message to the transport layer after having verified the HandshakeReply and derived the shared secret. Both timestamps are taken inside the DDS Security authentication plugin using CLOCK_MONOTONIC, so the metric is unaffected by any clock offset between the two boards. The metric consequently covers the transmission of the HandshakeRequest and the HandshakeReply, together with all requester-side verification and KEM operations, but not the transmission of the HandshakeFinal itself; the 288 B size difference of the HandshakeFinal reported in Table 2 therefore does not contribute to the measured latency, which is driven mainly by the HandshakeRequest and the HandshakeReply.

The preceding DDS Discovery phase is excluded: participant discovery must complete before the first handshake message is emitted, and its duration is not included in any value reported in Table 6. A run is counted as successful only if the requester emits the HandshakeFinal within the eight-second lifetime of the Req process; otherwise it is recorded as a timeout.

Runs were classified using the RTPS packet traces captured on both endpoints. A run was considered retransmission-affected when an RTPS retransmission associated with the HandshakeRequest or HandshakeReply was observed within the measured handshake interval; otherwise, it was classified as retransmission-free.

The first five runs of each unit were discarded as warm-up, and the remaining 1,000 measurements were used for the statistical analysis. The 95% confidence interval for the median difference was estimated using a nonparametric bootstrap with 10,000 independent resamples. Cliff's delta was additionally computed as a distribution-free effect-size measure.

5 Experimental Results and Analysis

5.1 KEM Computation Time

Table 5 presents the KEM computation times of the two algorithms. The measurements show that the KEM computation time of both algorithms is less than 1 ms and very small relative to the overall DDS Security handshake latency. This indicates that, when interpreting the difference in handshake latency, the transmitted handshake message size and the network conditions must be considered together with the standalone KEM computation cost.

Table 5. KEM computation time (ms).

Algorithm	KeyGen	Encaps	Decaps	Total
SMAUG-T TiMER	0.124	0.106	0.138	0.368
ML-KEM-512	0.156	0.130	0.120	0.406

5.2 Handshake Latency under Controlled Wireless Conditions

Table 6 presents the distribution of the handshake latency of the two algorithms under the four controlled wireless conditions. Each cell aggregates 1,000 measurements, and no failed or timed-out handshakes were observed. SMAUG-T TiMER showed the lower median in all four conditions.

Native and BW-only conditions. Without packet loss, both algorithms exhibited relatively compact latency distributions, and no retransmission-affected runs were observed. Under the Native condition, the median handshake latency was 16.17 ms for ML-KEM-512 and 15.35 ms for SMAUG-T TiMER, corresponding to a difference of 0.82 ms. Under BW-only, the corresponding medians were 16.47 ms and 15.55 ms, giving a slightly larger difference of 0.92 ms. SMAUG-T TiMER also showed smaller IQR and standard deviation in both conditions, indicating a more compact distribution in the absence of packet loss.

Because all 1,000 runs were retransmission-free, the retransmission-free analysis used the full set of measurements. The 95% confidence intervals for the median differences were [0.66, 0.98] ms under Native and [0.78, 1.06] ms under BW-only. The corresponding Cliff's delta values were 0.32 and 0.40, respectively, showing that the latency difference was maintained under bandwidth limitation.

Lossy conditions. Packet loss produced markedly different latency distributions under both Loss-only and BW+Loss. Although the median handshake latency remained in the tens-of-milliseconds range, the distributions became strongly right-skewed, with IQR values of approximately 1 s and upper-percentile latencies reaching approximately 2–3 s (Table 6). Under Loss-only, the median

Table 6. Distribution of the DDS Security handshake latency under controlled wireless conditions (ms).

Condition	Algorithm	Median	IQR	Mean	SD	P90	P95
Native	ML-KEM-512	16.17	2.22	16.16	1.57	18.03	18.53
	SMAUG-T TiMER	15.35	1.19	15.63	1.26	16.76	18.14
BW-only	ML-KEM-512	16.47	1.79	16.72	1.54	18.73	19.33
	SMAUG-T TiMER	15.55	0.87	15.70	0.87	16.56	17.05
Loss-only	ML-KEM-512	17.83	1003.72	578.32	776.36	2018.46	2022.83
	SMAUG-T TiMER	16.72	1003.47	617.09	941.95	2017.72	2018.96
BW+Loss	ML-KEM-512	19.37	1004.37	669.01	972.57	2022.61	3019.29
	SMAUG-T TiMER	17.46	1003.71	662.46	919.82	2019.79	3018.86

Table 7. Handshake latency for retransmission-free runs and the 95% confidence interval of the median difference between the two algorithms (ms). δ denotes Cliff's delta.

Condition	ML-KEM-512		SMAUG-T TiMER		Diff. 95% CI	δ
	n	Median	n	Median		
Native	1000/1000	16.17	1000/1000	15.35	[0.66, 0.98]	0.32
BW-only	1000/1000	16.47	1000/1000	15.55	[0.78, 1.06]	0.40
Loss-only	591/1000	16.52	609/1000	15.40	[0.93, 1.31]	0.43
BW+Loss	572/1000	16.89	593/1000	15.75	[0.96, 1.32]	0.47

latency was 17.83 ms for ML-KEM-512 and 16.72 ms for SMAUG-T TiMER, corresponding to a difference of 1.11 ms. Under BW+Loss, the corresponding values were 19.37 ms and 17.46 ms, increasing the median difference to 1.91 ms. The large separation between the median and mean under packet loss indicates that the mean and standard deviation were strongly influenced by long-delay observations. Nevertheless, SMAUG-T TiMER retained lower median, P90, and P95 latency under Loss-only, while the upper-percentile values of the two algorithms remained similar under BW+Loss. Thus, the median provides a more representative measure of typical handshake latency in these strongly skewed distributions. To distinguish the typical handshake behavior from runs affected by retransmission, Table 7 reports the retransmission-free subset. Under Loss-only, 591 ML-KEM-512 runs and 609 SMAUG-T TiMER runs remained, with median latencies of 16.52 ms and 15.40 ms, respectively. Under BW+Loss, the corresponding counts were 572 and 593, with medians of 16.89 ms and 15.75 ms. The resulting median differences were 1.12 ms and 1.14 ms, with 95% confidence intervals of [0.93, 1.31] ms and [0.96, 1.32] ms, respectively. Cliff's delta was 0.43 under Loss-only and 0.47 under BW+Loss. SMAUG-T TiMER therefore maintained lower typical handshake latency even after retransmission-affected runs were excluded. It also showed slightly higher retransmission-free proportions under both lossy conditions: 60.9% versus 59.1% under Loss-only and 59.3% versus 57.2% under BW+Loss. Together with its smaller measured-path message size and lower fragmentation count, these results provide supporting evidence that SMAUG-T TiMER may reduce exposure to loss-related transmission overhead.

However, the modest differences in retransmission-free proportions and the pronounced upper tail indicate that stochastic packet loss and retransmission behavior remain important determinants of tail latency.

6 Conclusion

This paper replaced the DH-based key-exchange component of the ROS 2 DDS Security handshake with a KEM-based structure and compared ML-KEM-512 and SMAUG-T TiMER under four controlled wireless conditions. SMAUG-T TiMER achieved lower median handshake latency in all conditions, with median differences of 0.82, 0.92, 1.11, and 1.91 ms under Native, BW-only, Loss-only, and BW+Loss, respectively. Under packet loss, both algorithms exhibited strongly right-skewed distributions with retransmission-related long-delay runs. Nevertheless, the latency advantage of SMAUG-T TiMER remained in the retransmission-free subset, indicating that the observed median difference was not solely attributable to retransmission events. SMAUG-T TiMER also showed slightly higher retransmission-free proportions under both lossy conditions. Together with its smaller measured-path message size and lower fragmentation count, these results provide supporting evidence that message size can affect DDS Security handshake latency under constrained links, although stochastic packet loss and retransmission behavior remain important factors in tail latency.

This study has three main limitations. First, ECDSA P-256 with SHA-256 was retained for digital signatures, so the implementation is not fully post-quantum secure. Second, the evaluation was limited to a single requester–responder pair. Third, ECDH P-256 was included only as a transmission-level baseline; its latency was not measured under the same network conditions. Future work will evaluate post-quantum authentication, the classical latency baseline, and multi-node scalability.

Acknowledgments. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2026-25585348, Graduate School of Security Convergence(AI)(Gachon University)) and (RS-2024-00396797, Development of Core Technology for Intelligent O-RAN Security Platform)

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

References

1. Blanco-Romero, J., Lorenzo, V., Almenares, F., Díaz-Sánchez, D., Serrano-Navarro, A.: PQSec-DDS: Integrating Post-Quantum Cryptography into DDS Security for robotic applications. In: Proceedings of the IX Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2024). pp. 396–403 (2024)

2. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367. IEEE (2018). https://doi.org/10.1109/EuroSP.2018.00032
3. Cheon, J.H., Choe, H., Seo, J., Seong, H.: SMAUG(-T), Revisited: Timing-Secure, More Compact, Less Failure. IEEE Access **12**, 188386–188397 (2024). https://doi.org/10.1109/ACCESS.2024.3511346
4. Kampanakis, P., Childs-Klein, W.: The impact of data-heavy, post-quantum TLS 1.3 on the time-to-last-byte of real-world connections. In: Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb), NDSS Symposium (2024). https://doi.org/10.14722/madweb.2024.23010
5. KpqC Team: Selected algorithms from the KpqC competition round 2. https://kpqc.or.kr/competition_02.html (2025), last accessed 2025-06-26
6. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics **7**(66), eabm6074 (2022). https://doi.org/10.1126/scirobotics.abm6074
7. National Institute of Standards and Technology: Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS Publication 203, National Institute of Standards and Technology, Gaithersburg, MD (2024). https://doi.org/10.6028/NIST.FIPS.203
8. Object Management Group: DDS Security, Version 1.2. OMG specification, Object Management Group (2024), https://www.omg.org/spec/DDS-SECURITY/1.2
9. Open Robotics: Using quality-of-service settings for lossy networks. https://docs.ros.org/en/jazzy/Tutorials/Demos/Quality-of-Service.html (2024), last accessed 2026-06-26
10. Safdar, U., Qadir, S., Nazir, M.S.: Enhancing ROS 2 (robot operating system 2) security with standardized post-quantum cryptosystems. International Journal of Information Security **24**, 219 (2025). https://doi.org/10.1007/s10207-025-01133-w
11. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science (FOCS). pp. 124–134. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365700

Double-Layered Identity Verification for UAVs via Telemetry Analysis

Shincheol Lee[1], Alimov Abdulboriy Abdulkhay ugli[2], Karimov Azamatjon Alijon ugli[2], and Ji Sun Shin[3]

[1] National Institute of Information and Communications Technology, Tokyo, Japan
shincheol.lee@nict.go.jp

[2] Department of Computer and Information Security, and the Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea
adamalimov98@gmail.com, a_karimov2001@mail.ru

[3] Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea
jsshin@sejong.ac.kr

Abstract. The growing deployment of unmanned aerial vehicles (UAVs) raises security concerns such as unauthorized access and identity spoofing in regulated airspace, where verifying a drone's identity typically relies on external RF, acoustic, or radar infrastructure. We propose a double-layered UAV airframe-attribute verification framework that operates solely on in-flight telemetry, needs no external sensing, and assumes an authenticated telemetry channel. A supervised first layer classifies the UAV type—quadrotor, hexarotor, octorotor, fixed-wing, or VTOL—from window-level telemetry features, reaching 93.3% whole-flight accuracy under flight-grouped cross-validation. For quadrotors, a second layer classifies the frame size among six classes (200–680 mm) using a bank of size-specific AutoEncoders scored by reconstruction error. Evaluated on 94 labeled PX4 flight logs spanning multiple airframes, with flight-grouped cross-validation, the frame-size stage attains 89.36% whole-flight accuracy (88.79% macro F1). The results show that telemetry alone provides a lightweight, infrastructure-free basis for verifying the airframe attributes of a UAV. We stress that this is an attribute-verification primitive intended as one factor within a larger authentication system, not a stand-alone cryptographic identity proof.

Keywords: UAV security · Drone identification · Telemetry analysis · AutoEncoder · Machine learning.

Unmanned aerial vehicles (UAVs) have moved rapidly from specialist tools to commodity devices, with civilian and commercial deployments spanning logistics, surveillance, mapping, infrastructure inspection, and precision agriculture. This growth has been accompanied by a parallel rise in security concerns: drones can enter restricted or sensitive airspace, impersonate authorized aircraft, or carry payloads whose risk depends directly on the airframe involved. Knowing *what*

kind of drone is operating in a given airspace—its type and physical scale—is therefore a prerequisite for any meaningful access-control or threat-assessment decision [15, 2].

We use three terms in a specific and deliberately narrow sense. Identification is the assignment of an observed UAV to a class of airframes. Attribute verification is the confirmation that an observed UAV's physical attributes are consistent with a declared or expected profile. Authentication is the establishment of a unique, unforgeable binding between an observed aircraft and a claimed identity. The framework in this paper performs the first two. It does not perform the third: two physically identical quadrotors of the same frame size are, by construction, indistinguishable to it. Our contribution is therefore a lightweight consistency check, one that can corroborate or contradict a declared identity, for example a Remote ID broadcast, without any external sensing infrastructure, rather than a replacement for cryptographic authentication.

Existing technical countermeasures largely fall into three families. Radio-frequency (RF) approaches identify drones from their communication signals or control-link emissions; acoustic approaches exploit the characteristic sound of rotor and motor noise; and radar approaches use micro-Doppler signatures. While effective in their intended settings, these modalities share a common drawback: they depend on dedicated external sensing infrastructure (RF front ends, microphone arrays, radar) and on favorable propagation conditions, which constrains where and how cheaply they can be deployed [15, 11]. A complementary line of work fingerprints drones through physical-layer hardware imperfections, for example unintentional electromagnetic emissions [4] or inertial-sensor manufacturing variation [3]; these are powerful but typically require close-range, controlled measurement of the target device.

In this paper we pursue a different basis for identification: the drone's own *in-flight telemetry*. Modern flight controllers such as Pixhawk, running PX4 or ArduPilot, continuously log inertial and power data—accelerometer, gyroscope, and magnetometer readings, and battery state—as a natural by-product of flight. These signals encode behavioral and dynamical signatures that differ across airframe configurations and physical scales. Because the data already exists onboard, a verification system built on telemetry needs no visual sensors, no RF acquisition, and no external infrastructure, and depends on no specialised accelerator.

We propose a double-layered UAV airframe-attribute verification framework that operates solely on in-flight telemetry. The first layer is a supervised classifier that assigns the UAV type—quadrotor, hexarotor, octorotor, fixed-wing, or VTOL—from window-level telemetry features. Because quadrotors dominate real-world deployments, the second layer refines the identity of any UAV classified as a quadrotor by estimating its frame size. The second layer treats frame-size classification as an unsupervised reconstruction problem: a bank of size-specific AutoEncoders, each trained only on flights of a single frame size, scores an incoming flight by reconstruction error, and the lowest-error class is selected.

A central concern in telemetry-based classification is that reported accuracy can be inflated by evaluation leakage. Flight logs are long and internally correlated, so windows or samples drawn from the same flight are highly similar; if such windows are split across training and test sets, a model can score well merely by recognizing neighbors of flights it has already seen. We address this directly by evaluating the frame-size stage with grouped cross-validation, in which every window of a given flight is held out together, so that reported scores reflect generalization to entirely unseen flights. We further expand the frame-size dataset well beyond a single airframe, combining self-collected flights, real flights drawn from a published multicopter dataset [9], and curated public PX4 logs spanning multiple manufacturers and custom builds.

Contributions. This paper makes the following contributions:

- We present a double-layered, telemetry-only UAV airframe-attribute verification framework that first classifies UAV type with supervised models and then, for quadrotors, classifies frame size among six classes (200–680 mm) using size-specific AutoEncoders. The pipeline requires no visual, RF, or external infrastructure and is compatible with standard PX4/ArduPilot controllers.
- We assemble a frame-size dataset of 94 labeled flight logs spanning six frame sizes and multiple airframes, and we describe a transparent inclusion procedure for public logs that mitigates label noise.
- We adopt a leakage-resistant evaluation protocol—grouped cross-validation by source flight—and report honest, whole-flight generalization performance (89.36% accuracy, 88.79% macro F1) rather than window-level scores that overstate real-world reliability.

The remainder of the paper is organized as follows. Section 1 reviews related work on drone identification, authentication, and telemetry-based learning. Section 2 presents the system overview, and Section 3 describes the dataset. Section 4 details the two-stage methodology, and Section 5 reports experimental results. Section 6 discusses findings and limitations, and Section 7 concludes.

1 Related Work

We organize prior work into four areas: drone detection and identification by external sensing, telemetry- and log-based UAV classification, behavior-based drone and operator authentication, and AutoEncoder-based anomaly detection for UAV flight data.

Drone detection and identification by external sensing. A large body of work identifies drones using signals external to the flight controller. RF-based methods classify drones from their communication or control-link signals; surveys by Taha and Shoufan [15] and Mrabet et al. [11] catalog RF, radar, and

vision pipelines and report that, while individual systems achieve high accuracy, they depend on dedicated sensing hardware and controlled conditions. Acoustic approaches exploit rotor/motor noise: Al-Emadi et al. [1] applied deep networks to drone audio for detection and multi-class identification, and Kolamunna et al. [8] proposed DronePrint, using acoustic signatures with LSTM models for open-set drone identification. Radar micro-Doppler methods classify multirotors and fixed-wing craft from reflected-signal signatures. More recent physical-layer fingerprinting work identifies devices from hardware imperfections—for example Drone-Mag [4], which authenticates drones from unintentional electromagnetic emissions, and IMU-fingerprinting schemes that exploit sensor manufacturing variation [3]. These methods are accurate but require close-range or infrastructure-dependent measurement of the target. In contrast, our framework uses only the drone's own logged telemetry and requires no external sensing.

Telemetry and log-based UAV classification. Closer to our setting is work that classifies UAVs directly from flight logs. Crnovrsanin et al. [2] train an LSTM to classify UAV type (quadrotor, hexarotor, fixed-wing) from the public PX4 Flight Review repository, reporting strong accuracy for the well-represented quadrotor class (about 98% macro F-score) but markedly lower accuracy for fixed-wing (about 73%) and hexarotor (about 42%) flights under severe class imbalance. Importantly, they identify the PX4 Flight Review database itself as a challenge: it was accumulated over roughly a decade with heterogeneous, voluntarily uploaded, and sometimes inconsistently described logs. This observation directly motivates our conservative inclusion rule for public logs. Our work differs in three respects: we extend type classification to five categories (adding octorotor and VTOL); we add a second *frame-size* stage absent from prior log-based classifiers; and we adopt an explicitly leakage-resistant, flight-grouped evaluation.

Behavior-based drone and operator authentication. A related thread authenticates drones or their operators from behavioral data. Karimibiuki et al. [6] showed that drones can be authenticated from their flight trajectories using ML classifiers, and Jiang et al. [5] proposed an intelligent UAV identity-verification scheme using flight data to flag anomalous or unauthorized control. A substantial line of work by Shoufan and colleagues [14] identifies drone *operators* from the behavioral patterns in their control-command sequences, achieving roughly 76–90% identification accuracy, and subsequent work has extended operator identification with ensemble and incremental-learning methods. These approaches generally rely on the secure transmission of flight or command data, and they target *who* is flying rather than *what* is flying. Our framework is complementary: it verifies the type and physical scale of the airframe itself and can support both self- and third-party authentication when telemetry integrity is ensured.

AutoEncoder-based anomaly detection for UAV flight data. Our frame-size stage builds on the established use of AutoEncoders for UAV flight-data

analysis, where a model trained on nominal data yields low reconstruction error on in-distribution inputs and higher error on out-of-distribution inputs. Recent examples include LSTM-AutoEncoder pipelines for edge fault detection on PX4 logs [12], wavelet-plus-stacked-denoising-AutoEncoder anomaly detectors [16], and self-supervised period-aware AutoEncoders for UAV time series [10]. Almost all of this work uses reconstruction error as an *anomaly* signal—normal versus faulty. We repurpose the same principle for a *multi-class* task: rather than one model separating normal from abnormal, a bank of class-specific AutoEncoders competes by reconstruction error to assign one of several frame sizes, with errors aggregated over a full flight to produce the final decision. The published multi-copter dataset we draw real flights from, RflyMAD [9], was itself developed for fault detection and health assessment; we use only its normal real-flight logs.

2 System Overview

The proposed system performs UAV airframe-attribute verification using only in-flight telemetry, as a modular pipeline that requires no visual, radio-frequency (RF), or external infrastructure. Each drone carries a Pixhawk-compatible flight controller running PX4 or ArduPilot, which logs inertial data—accelerometer, gyroscope, and magnetometer readings across three axes—as a natural by-product of flight. When a drone enters the monitored airspace, a short segment of this telemetry is captured and passed to a two-stage classification pipeline. In the first stage, a supervised classifier determines the UAV type among five categories: quadrotor, hexarotor, octorotor, fixed-wing, and VTOL. If, and only if, the UAV is classified as a quadrotor, the second stage estimates its frame size by casting frame-size classification as an unsupervised reconstruction problem: a bank of size-specific AutoEncoders, each trained on a single frame-size class, scores the flight by reconstruction error, and the lowest-error class is selected. The system distinguishes six quadrotor frame sizes—200, 250, 450, 500, 650, and 680 mm—by diagonal motor-to-motor distance. The result is forwarded to an access controller that checks the inferred airframe profile against a user-defined whitelist of permitted profiles; profiles outside the list are flagged, logged, or denied. Because a profile denotes a class of airframes rather than an individual aircraft, this check constrains what may operate in the airspace and does not by itself establish which aircraft is operating. An overview is shown in Fig. 1.

The pipeline consumes telemetry as ground truth, so the trust boundary lies at the telemetry source. We assume telemetry reaches the verifier over an authenticated, integrity-protected channel, for example a MAVLink 2 link with message signing enabled, or direct offline reading of the onboard log by the operator. Under that assumption the framework detects a mismatch between an aircraft's declared profile and its observed flight dynamics. Outside that assumption its guarantees are limited, and we state the limits explicitly. An adversary who can inject unsigned telemetry can replay a previously captured log from an authorised aircraft, and our per-window features carry no freshness or nonce, so the system alone cannot distinguish a replay from a live flight; replay resistance

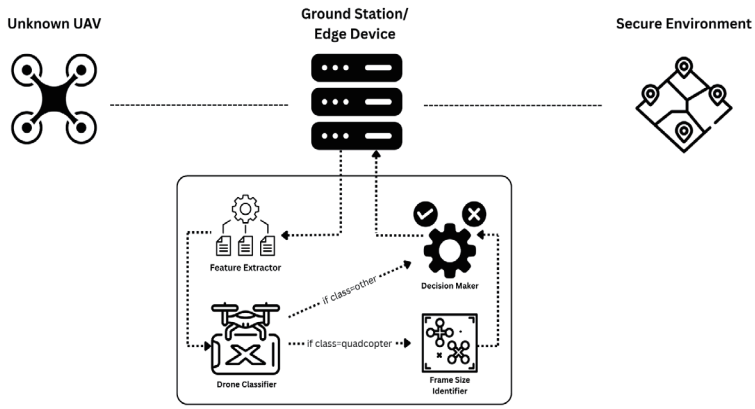


Fig. 1. Overview of the proposed double-layered airframe-attribute verification pipeline.

must come from the transport layer, through sequence numbers, timestamps, or challenge–response. An adversary who can modify telemetry in transit can bias the statistical descriptors of Section 4.2 arbitrarily. Finally, and independently of any channel protection, an adversary flying a genuinely similar airframe will legitimately produce telemetry consistent with the authorised profile, because the framework verifies airframe attributes rather than aircraft identity. This is a property of the approach, not a deficiency of the implementation, and it is why we position the method as one factor among several rather than a stand-alone access-control mechanism.

3 Dataset Description

Reliable UAV airframe-attribute verification requires training and evaluation on diverse flight logs spanning multiple airframe configurations. We therefore compiled a dataset covering the two tasks of the framework: (i) UAV-type classification across five categories, and (ii) quadrotor frame-size classification across six size classes. All logs are PX4 ULog recordings containing raw onboard sensor telemetry.

3.1 UAV Types (Stage 1)

For type classification, the dataset spans five widely deployed UAV configurations: quadrotor, hexarotor, octorotor, fixed-wing, and VTOL. The flight logs were primarily sourced from the official Pixhawk Flight Review repository [13],

Table 1. Labeled ULog files and analysis windows by UAV type.

UAV Type	Labeled ULog Files	Analysis Windows
Quadrotor	96	5,233
Hexarotor	26	1,686
Octorotor	16	1,378
VTOL	14	2,028
Fixed-wing	13	2,503
Total	**165**	**12,828**

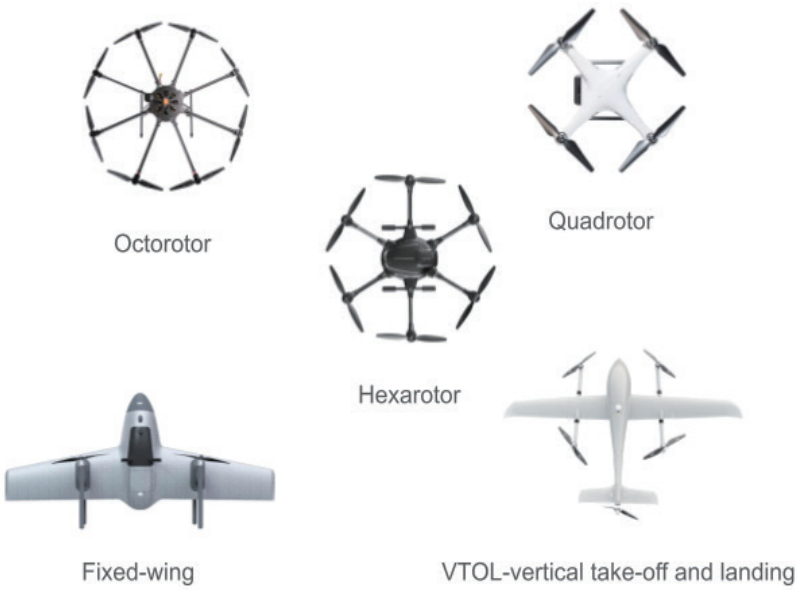


Fig. 2. Illustration of the five UAV types considered in this study: quadrotor, hexarotor, octorotor, fixed-wing, and VTOL.

which provides high-quality datasets recorded during real flights under varying environmental conditions. Additionally, we included custom-collected quadrotor data to strengthen UAV type detection experiments, as quadrotors constitute the most common class in practical deployments. Examples are shown in Figure 2.

The Stage 1 dataset comprises 165 labeled ULog files yielding 12,828 analysis windows, distributed across the five types as shown in Table 1. The distribution is deliberately unbalanced in favour of quadrotors, which dominate real deployments and which alone gate Stage 2. We note that 94 of the 96 quadrotor logs are the same files used for the frame-size task of Section 3.2, so the two stages share the majority of their quadrotor data.

3.2 Quadrotor Frame Sizes (Stage 2)

For frame-size classification, the dataset covers six quadrotor frame sizes—200, 250, 450, 500, 650, and 680 mm—where the size designation corresponds to the diagonal motor-to-motor distance, a standard metric for defining drone frames. The dataset is not tied to a single airframe or a single flight environment; instead, it combines three complementary sources:

- **Self-collected flights.** We assembled physical quadcopters and flew them under controlled conditions, recording each flight from take-off to landing in ULog format. These flights provide frame-size labels that are verified by direct knowledge of the airframe, and are released as part of the DroneLog flight-telemetry dataset [7].
- **A published UAV dataset.** We incorporated real-flight logs from the publicly released RflyMAD multicopter dataset [9]. Although that dataset was developed for fault detection and health assessment, it includes a normal-flight (fault-free) subset, and we used only those normal real-flight logs. These flights were recorded on Droneyee quadcopters at three diagonal frame sizes—200, 450, and 680 mm—using PX4 autopilots, which aligns directly with our frame-size labels. We did not use the simulation portion of that dataset.
- **Curated public logs.** We additionally incorporated flights from the public PX4 Flight Review database to introduce airframe and condition diversity. To avoid label noise, public logs were included only when the airframe was *explicitly and unambiguously* identified in the log metadata; entries with generic or version-ambiguous descriptions (e.g., an unlabeled "quadcopter") were excluded.

The frame-size dataset spans multiple manufacturers and custom builds, but the hardware diversity is distributed unevenly across the six classes, and we characterise it explicitly rather than assert it. Every PX4 ULog records the unique serial number of its flight-control board (sys_uuid) together with the board type (ver_hw) and the airframe configuration identifier (SYS_AUTOSTART). Using these fields we resolved each of the 94 logs to the physical flight-control board that produced it. The 94 logs originate from 19 distinct boards, distributed as shown in Table 2. Two classes are dominated by a single aircraft: all 30 logs of the 200 mm class come from one board, and 10 of the 13 logs of the 680 mm class come from another. Conversely, one board appears in both the 450 mm and the 680 mm class, having been flown on two different airframes — a configuration we exploit in Section 5.3 to separate frame-size effects from hardware effects. The intent of combining sources was to discourage the AutoEncoders from keying on one specific airframe; Section 5.3 and Section 6 report how far that intent was actually achieved.

The final frame-size dataset contains 94 labeled ULog files distributed across the six classes, as summarized in Table 3. From these logs, the feature-extraction pipeline (Section 4.2) produces 5,210 fixed-length analysis windows.

Table 2. Distribution of logs and flight controller boards by frame size.

Frame Size	Logs	Distinct Boards	Board Types Observed	Largest Single Board
200 mm	30	1	DRONEYEE_ZYFC_H7	30 / 30
250 mm	12	5	PX4_FMU_V5, PX4_FMU_V6C	5 / 12
450 mm	15	4	DRONEYEE_ZYFC_H7, PX4_FMU_V4PRO, PX4_FMU_V6X	10 / 15
500 mm	13	4	PX4_FMU_V6C, PX4_FMU_V6X	8 / 13
650 mm	11	4	PX4_FMU_V6C, PX4_FMU_V6X	4 / 11
680 mm	13	2	DRONEYEE_ZYFC_H7, PX4FMU_V4PRO	10 / 13
Total	**94**	**19**	—	—

Table 3. Frame-size dataset: labeled ULog files and extracted analysis windows per class. Each window spans five seconds with 50% overlap and is described by 176 features.

Frame size	Labeled ULog files	Analysis windows
200 mm	30	1,052
250 mm	12	781
450 mm	15	438
500 mm	13	1,324
650 mm	11	696
680 mm	13	919
Total	**94**	**5,210**

4 Methodology

The framework processes each flight in two stages. Stage 1 classifies the UAV type from window-level telemetry features using supervised models. Stage 2, activated only for quadrotors, classifies the frame size from window-level statistical features using a bank of size-specific AutoEncoders. We describe the shared preprocessing, then each stage.

4.1 Stage 1: UAV-Type Classification

Stage 1 classifies the UAV type from flight telemetry. To keep a single coherent pipeline, it uses the same window-level representation as Stage 2 (Section 4.2): each ULog file is segmented into five-second windows with 50% overlap, and every window is described by the same 176 statistical features extracted from magnetometer, gyroscope, accelerometer, and battery/power telemetry. A supervised classifier assigns each window one of the five UAV types, and the log-level

prediction is obtained by majority vote over all windows of the flight. We evaluate five classifiers under grouped cross-validation by source ULog file, so that all windows of a flight remain entirely in either the training or the test fold. Stage 2 is triggered only when Stage 1 labels the UAV a quadrotor.

4.2 Stage 2: Quadrotor Frame-Size Classification

Stage 2 estimates the physical frame size of a quadrotor from its PX4 ULog telemetry. Stage 2 operates on *window-level* statistical descriptors of the flight, which capture the temporal signatures that distinguish frame sizes.

Telemetry sources and derived signals. Stage 2 draws on four telemetry categories: magnetometer, gyroscope, and accelerometer measurements, together with battery and power telemetry. Gyroscope and accelerometer values are read from the `sensor_combined` stream; magnetometer values are read from `vehicle_magnetometer` when available, falling back to the corresponding `sensor_combined` fields; battery values are read from `battery_status`. Streams are aligned by timestamp, using the frequently sampled gyroscope and accelerometer as the temporal anchor, with magnetometer and battery values interpolated to the anchor timestamps.

The aligned telemetry yields 17 signals: the three axes of each of the magnetometer, gyroscope, and accelerometer, plus per-sensor vector norms, and four battery/power signals (voltage, current, discharged capacity, and remaining charge), together with instantaneous power. The vector norms and power are derived as

$$\|\mathbf{s}\| = \sqrt{s_x^2 + s_y^2 + s_z^2}, \qquad P = V \cdot I, \tag{1}$$

where $\mathbf{s} \in \{\text{mag}, \text{gyro}, \text{acc}\}$, and V and I are the measured voltage and current.

Window-level feature extraction. Each ULog file is segmented into five-second windows with 50% overlap; windows with fewer than 20 sensor rows are discarded. The overlap allows even short flights to yield usable windows while longer flights contribute several. For each of the 17 signals, ten statistical descriptors are computed—mean, standard deviation, minimum, maximum, median, interquartile range, root mean square, range, end-to-start delta, and slope—giving $17 \times 10 = 170$ features. Six window-level features are appended (window duration, sample count, battery-presence fraction, estimated energy consumption, discharged-capacity delta, and remaining-charge drop), for a final representation of **176 features per window**.

Size-specific AutoEncoders. Stage 2 uses six AutoEncoders, one per frame-size class. Each AutoEncoder is trained only on windows of its own class and

learns to reconstruct the telemetry patterns typical of that frame size. All six share the same architecture: a symmetric encoder–decoder with layer sizes $176 \rightarrow 64 \rightarrow 16 \rightarrow 64 \rightarrow 176$, where the 16-unit bottleneck forms the latent representation. Models use ReLU activations, the Adam optimizer, and z-score–standardized inputs, and are trained to minimize mean absolute reconstruction error (MAE). The training budget is capped at 600 epochs with early stopping; in practice the final models converged between 187 and 404 epochs.

Let E_i denote the reconstruction MAE produced by the AutoEncoder for class $i \in \{200, 250, 450, 500, 650, 680\}$ on a given input. For a single window, the predicted frame size is the class of minimum reconstruction error,

$$\hat{s} = \arg\min_i \; E_i. \tag{2}$$

Inference and log-level aggregation. An unknown quadrotor log is processed through the same pipeline—stream alignment, five-second windowing, 176-feature extraction, and standardization—after which every window is scored by all six AutoEncoders. Per-window predictions are then aggregated to a single log-level decision. The primary decision rule selects the frame size with the lowest *mean* reconstruction MAE across all valid windows of the flight:

$$\hat{s}_{\log} = \arg\min_i \; \frac{1}{N} \sum_{w=1}^{N} E_i^{(w)}, \tag{3}$$

where N is the number of valid windows and $E_i^{(w)}$ is the reconstruction error of class i on window w. A majority vote over per-window predictions is also computed as a supplementary log-level output.

5 Experimental Results

5.1 Stage 1: UAV-Type Classification

Stage 1 was evaluated on the five UAV types using the 176 window-level features under grouped cross-validation by source ULog file. Table 5 reports whole-flight accuracy and macro F1 for the five classifiers under grouped cross-validation. Linear SVC performs best (93.33% accuracy, 88.35% macro F1), with the tree- and boosting-based models close behind. Quadrotor logs, the operationally critical class that gates Stage 2, are classified without error (96/96), while fixed-wing and VTOL show occasional confusion owing to their smaller per-class log counts. Perfect quadrotor recall is favourable for the pipeline, since it means Stage 1 never withholds a quadrotor from Stage 2 on this dataset; we caution, however, that the quadrotor class is also the class with the least device diversity relative to its size (Table 3), so this figure should not be read as evidence of generalisation to unseen quadrotor hardware.

The matrix also bounds the cost of Stage 1 errors for the combined pipeline. Because no quadrotor is misclassified, no quadrotor is denied frame-size analysis.

Table 4. Stage 1 confusion matrix for Linear SVC under grouped cross-validation with whole-flight majority voting.

True \ Predicted	Fixed-wing	Hexarotor	Octorotor	Quadrotor	VTOL
Fixed-wing	**10**	0	1	2	0
Hexarotor	0	**25**	0	0	1
Octorotor	0	0	**13**	3	0
Quadrotor	0	0	0	**96**	0
VTOL	1	0	1	2	**10**

Table 5. Stage 1 UAV-type classification under grouped cross-validation (whole-flight majority vote). Best result in bold.

Model	Accuracy	Macro F1
Linear SVC	**93.33%**	**88.35%**
LightGBM	91.52%	85.58%
XGBoost	90.91%	85.14%
Extra Trees	89.70%	82.50%
Random Forest	89.09%	79.34%

In the opposite direction, seven non-quadrotor logs, three octorotor, two fixed-wing, and two VTOL, are labeled quadrotor and would therefore be passed to Stage 2, which would assign each of them a frame size from its six-class vocabulary even though no such label is meaningful. Stage 2 as currently formulated is closed-set and has no reject option, so these seven cases would yield confidently wrong output. Adding a reconstruction error threshold to Stage 2, so that a flight whose minimum error exceeds all six training distributions is rejected rather than labeled, is the natural remedy and we leave it to future work.

5.2 Stage 2: Frame-Size Classification

Evaluation protocol. We evaluate Stage 2 with grouped three-fold cross-validation in which the grouping unit is the source ULog file, using `StratifiedGroupKFold` so that all windows of a flight remain entirely within either the training or the test fold while the class distribution is approximately preserved. Within each fold the standardization statistics and all six AutoEncoders are fit on the training logs only. As motivated in Section 1, grouping by flight prevents the window-overlap leakage that a random split would introduce, so the reported scores measure generalization to *previously unseen flights* rather than recognition of neighboring windows of an already-observed flight.

Results. Table 7 reports performance at three levels: individual five-second windows, whole-flight majority vote, and whole-flight minimum-mean-MAE (the primary rule). Aggregating window scores to a log-level decision substantially improves accuracy, as expected: individual windows are noisy, but their errors average out over a full flight. Under the primary minimum-mean-MAE rule, the

Table 6. Stage 2 whole-flight confusion matrix under the minimum-mean-MAE rule.

True \ Predicted	d200	d250	d450	d500	d650	d680
d200	**29**	0	1	0	0	0
d250	0	**11**	0	0	0	1
d450	1	0	**13**	1	0	0
d500	2	0	0	**10**	1	0
d650	0	0	0	0	**11**	0
d680	0	0	3	0	0	**10**

Table 7. Stage 2 frame-size classification under grouped three-fold cross-validation (grouped by source ULog file). The whole-flight minimum-mean-MAE rule is the primary decision criterion.

Evaluation level	Accuracy	Macro Prec.	Macro Rec.	Macro F1
Individual 5 s windows	75.57%	75.95%	77.68%	75.32%
Whole flight: majority vote	88.30%	89.64%	86.86%	87.76%
Whole flight: min. mean MAE	**89.36%**	**90.10%**	**88.14%**	**88.79%**

system correctly classifies 84 of 94 held-out flight logs, for a whole-flight accuracy of 89.36% and a macro F1-score of 88.79%.

Table 9 gives the per-class breakdown under the primary rule. Recall is highest for d200 (96.7%) and d650 (100%), and lowest for d500 and d680 (both 76.9%); notably, d680 — the largest frame size in the dataset — does not share the strong performance of the other size extreme, d200. The dominant residual error is a confusion between certain 680 mm and 450 mm flights, indicating that additional diverse logs, particularly from physically distinct drones and a wider range of operating conditions, would further sharpen the decision boundary between these classes.

5.3 Frame size versus airframe hardware

Concern about hardware confounding motivates two additional analyses of the results already reported, both keyed on the physical board identifiers of Table 3. Flight-grouped cross-validation holds out whole flights but not whole aircraft, so a board contributing several logs will generally appear in both the training and the test partition. Hardware held constant. One board was flown on both a 450 mm and a 680 mm airframe, contributing 10 logs to each class. Restricted to these 20 logs, every factor visible to the flight controller is identical — the same board, the same firmware, the same battery configuration — and only the airframe differs. All 20 are classified correctly. This is direct evidence that the AutoEncoders capture frame-size–dependent dynamics separable from the identity of the flight-control hardware, and it is the strongest single piece of evidence we can offer that the task is genuinely learnable rather than a hardware-recognition artifact.

Accuracy on sparsely represented boards is 31.8 percentage points lower. The pattern is visible in the individual errors: all three 680 mm failures are the

Table 8. Sensitivity to device representation. The 94 logs are partitioned by how many logs their board contributes to its class.

Board representation in its class	Logs	Correct	Accuracy
Boards contributing ≥ 4 logs	67	66	98.5%
Boards contributing ≤ 3 logs	27	18	66.7%

Table 9. Stage 2 per-class results under the whole-flight minimum-mean-MAE rule.

Frame size	Correctly classified logs	Recall
200 mm (d200)	29 / 30	96.7%
250 mm (d250)	11 / 12	91.7%
450 mm (d450)	13 / 15	86.7%
500 mm (d500)	10 / 13	76.9%
650 mm (d650)	11 / 11	100.0%
680 mm (d680)	10 / 13	76.9%
Overall	**84 / 94**	**89.36%**

three logs from the one 680 mm board with no sibling logs, whereas the ten 680 mm logs from the well-represented board are all correct. We therefore conclude that a material part of the 89.36% figure reflects recognition of previously seen airframes rather than frame-size dynamics alone, and that the figure should be read as an upper bound on performance against genuinely unfamiliar aircraft. The two analyses are not contradictory: the first shows frame size is learnable independently of hardware when both are observed, while the second shows the present dataset does not contain enough distinct aircraft per class for the learned representation to transfer reliably to new ones.

6 Discussion and Limitations

The results indicate that in-flight telemetry alone carries enough behavioral signal to support both UAV-type and quadrotor frame-size classification, without any external sensing infrastructure. Aggregating noisy per-window predictions to a whole-flight decision raises frame-size accuracy from 75.6% to 89.4%, confirming that frame-size signatures are stable over a flight. Flight-grouped cross-validation ensures these numbers reflect generalisation to unseen flights; it does not establish generalisation to unseen aircraft, since a given flight-control board typically contributes several flights and is therefore present in both partitions. Section 5.3 quantifies the gap: accuracy is 98.5% on aircraft well represented in training and 66.7% on aircraft represented by three or fewer flights.

Limitations: Four limitations bound the scope of our claims. First, the dataset contains 18 distinct flight-control boards across six classes, and two classes are dominated by a single aircraft; consequently the frame-size stage is validated for unseen flights but only partially for unseen airframes, and closing this gap requires flights from additional physically distinct drones at each frame size rather than more flights of the same drones. Second, frame size is correlated

with other hardware choices in any real fleet, battery cell count in our data ranges from 3S to 6S and covaries with frame size, so we cannot fully attribute the discriminative signal to airframe geometry as opposed to the propulsion and power hardware that typically accompanies it, and an ablation isolating the contribution of individual feature groups remains future work. Third, Stage 2 is closed-set: it always returns one of six frame sizes and cannot reject an input belonging to none of them, which as noted in Section 5.1 means non-quadrotors surviving Stage 1 receive a meaningless label. Fourth, we characterise the pipeline's computational cost only structurally, 176 features per five-second window and six small AutoEncoders with a 16-unit bottleneck, and leave measured latency, memory, and minimum-observation-duration profiling to future work.

7 Conclusion

We presented a double-layered UAV airframe-attribute verification framework that operates solely on in-flight telemetry. A supervised first stage classifies the UAV type across five categories (93.3% whole-flight accuracy under grouped cross-validation), and an unsupervised second stage classifies quadrotor frame size among six classes (200–680 mm) using a bank of size-specific AutoEncoders scored by reconstruction error. Evaluated on 94 labeled flight logs spanning multiple airframes under leakage-resistant, flight-grouped cross-validation, the frame-size stage attains 89.36% whole-flight accuracy and an 88.79% macro F1-score. The framework needs no visual, RF, or external infrastructure and depends on no specialised accelerator, making it a practical consistency check on declared UAV airframe attributes in multi-drone environments. As set out in Section 2, it is intended as one factor within an authentication system rather than a stand-alone identity proof. Future work will disentangle frame size from airframe hardware by collecting flights from multiple physically distinct drones at each frame size, ablate individual feature groups to identify which telemetry channels drive the decision, profile end-to-end latency and memory on representative hardware, add a reject option so that Stage 2 can decline inputs outside its six classes, and assess robustness under adversarial and noisy telemetry conditions.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03038540.

References

1. Al-Emadi, S., Al-Ali, A., Mohammad, A., Al-Ali, A.: Audio based drone detection and identification using deep learning. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). pp. 459–464. IEEE (2019)

2. Crnovrsanin, T., Yu, C., Hankamer, D., Dunne, C.: Predicting uav type: An exploration of sampling and data augmentation for time series classification. arXiv e-prints pp. arXiv–2403 (2024)
3. Gong, B., Li, Z., Gong, M., Zhu, H., Meng, W., Guo, C.: Lightweight continuous authentication via imu fingerprinting for v2x. IEEE Transactions on Intelligent Transportation Systems (2025)
4. Ibrahim, O., Di Pietro, R.: Drone-mag: Uav identification and authentication via electromagnetic emissions. ACM Transactions on Cyber-Physical Systems **9**(3), 1–25 (2025)
5. Jiang, C., Fang, Y., Zhao, P., Panneerselvam, J.: Intelligent uav identity authentication and safety supervision based on behavior modeling and prediction. IEEE Transactions on Industrial Informatics **16**(10), 6652–6662 (2020)
6. Karimibiuki, M., Aibin, M., Lai, Y., Khan, R., Norfield, R., Hunter, A.: Drones' face off: authentication by machine learning in autonomous iot systems. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). pp. 0329–0333. IEEE (2019)
7. Karimov, A.A.u., Alimov, A., Makhammadjonov, S.A.u., Shin, J.S.: Dronelog: A flight telemetry benchmark dataset for uav airframe identification and payload state recognition (May 2026), https://doi.org/10.5281/zenodo.20279005
8. Kolamunna, H., Dahanayaka, T., Li, J., Seneviratne, S., Thilakaratne, K., Zomaya, A.Y., Seneviratne, A.: Droneprint: Acoustic signatures for open-set drone detection and identification with online data. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **5**(1), 1–31 (2021)
9. Le, X., Jin, B., Cui, G., Dai, X., Quan, Q.: Rflymad: A dataset for multicopter fault detection and health assessment. The International Journal of Robotics Research **44**(7), 1081–1092 (2025)
10. Li, J., Zhou, Y., Dong, Z.: Tsae-uav: A time-sense autoencoder for unsupervised anomaly detection of uav based on flight data. Aerospace Science and Technology p. 111431 (2025)
11. Mrabet, M., Sliti, M., Ammar, L.B.: Machine learning algorithms applied for drone detection and classification: benefits and challenges. Frontiers in Communications and Networks **5**, 1440727 (2024)
12. Reis, M.J., Reis, A.J.: Edge-based real-time fault detection in uav systems via b-spline telemetry reconstruction and lightweight hybrid ai. Sensors **25**(16), 4944 (2025)
13. Review, P.F.: Px4 flight review - public log database:https://review.px4.io, https://review.px4.io, accessed: July 18, 2025
14. Shoufan, A., Al-Angari, H.M., Sheikh, M.F.A., Damiani, E.: Drone pilot identification by classifying radio-control signals. IEEE Transactions on Information Forensics and Security **13**(10), 2439–2447 (2018)
15. Taha, B., Shoufan, A.: Machine learning-based drone detection and classification: State-of-the-art in research. IEEE access **7**, 138669–138682 (2019)
16. Zhou, S., He, Z., Chen, X., Chang, W.: An anomaly detection method for uav based on wavelet decomposition and stacked denoising autoencoder. Aerospace **11**(5), 393 (2024)

The Shift Ladder: Locating the Gains and Failures of Rule-Based Synthetic CAN Attack Augmentation

Hyunjin Jang[0009−0004−2649−8350], Dawit Shin[0009−0008−3593−4301], Upjun Ha[0009−0004−6726−2926], Thi-Thu-Huong Le[0000−0002−8366−9396], and Howon Kim*

School of Computer Science and Engineering, Pusan National University
flyho77@naver.com; ekdnlt0227@pusan.ac.kr; dqws304@gmail.com; lehuong7885@gmail.com; howonkim@pusan.ac.kr

Abstract. Rule-based synthetic CAN attacks can improve an intrusion detector on generated tests, yet a single score does not reveal whether the model learned transferable attack structure or generator-local cues. We present the Shift Ladder, a leakage-controlled protocol that tracks an augmentation gain from source performance through controlled variants, grammar exits, external datasets, and attribution probes. Applied to Car-Hacking-trained CAN IDS models, the ladder locates a repeatable but construction-local gain. Under a matched 7,992-update budget, Rule +30% raises fixed-variant binary recall from 0.497 to 0.9999 and coupled-sensitivity recall from 0.551 to 0.720. Both gains exceed equal-size real-attack oversampling, are positive in all five paired pipeline seeds, and persist on raw-frame-disjoint evaluation subsets. The gain weakens for low-severity variants and becomes mostly binary detection under a compound grammar exit as exact Gear/RPM construction-family retention collapses. On OTIDS, can-train-and-test, and ROAD, the source-policy CNN FPR remains near 1.0; a target-label-dependent ranking diagnostic finds no observed Rule +30% AUROC or average-precision improvement, and TPR is zero at empirical FPR ceilings of 0.1% and 1%. ID mismatch reproduces the external failure, while ID-channel removal does not repair can-train or OTIDS. The Shift Ladder converts a single augmentation score into a sequence of scope-specific claims, distinguishing controlled-variant coverage from cross-domain robustness.

Keywords: CAN bus · Intrusion detection · Synthetic data · Data augmentation · Automotive security

1 Introduction

Controller Area Network (CAN) buses lack sender authentication and encryption, enabling an attacker with bus access to inject, replay, or flood messages [9]. CAN intrusion-detection systems (IDSs) often approach saturation on public

* Corresponding author.

source splits, leaving variant coverage and deployment shifts unresolved [1]. Synthetic augmentation compounds this ambiguity because generated training and evaluation samples may share construction rules. A staged evaluation is therefore needed to separate source discrimination, controlled-variant coverage, and cross-domain transfer.

We introduce the *Shift Ladder* to locate how far a rule-based augmentation gain survives. Models train on Car-Hacking [5, 15], encounter generated variants and grammar-exit stresses derived from held-out Car-Hacking normal traffic, and then cross-test on OTIDS [4, 11], can-train-and-test [10], and ROAD [18]. Matched-update and real-oversampling controls test the gain against update-count and equal-volume real-data alternatives; separability, overlap, ranking, and support probes then locate its persistence and decay across construction and domain shifts. The resulting contribution is an evaluation protocol and an empirical account of construction-local boundary shaping in rule-augmented CAN IDS.

The paper contributes an evaluation protocol and three empirical findings.

- **Evaluation protocol.** The Shift Ladder keeps fitting and selection source-only, then stages source tests, controlled variants, grammar exits, external transfer, and attribution under matched-update, real-oversampling, and overlap controls.
- **Controlled gain.** Rule augmentation improves grammar-shared and grammar-adjacent variants while preserving source performance. The CNN gain survives matched update count, equal-size real-attack oversampling, and raw-frame-disjoint evaluation; an RF provides a second model-family check.
- **Gain decay.** Performance declines on low-severity variants, and a compound grammar exit changes the result from exact construction-family recognition to mostly binary detection.
- **External transfer diagnosis.** All three external datasets expose source-policy operating-point collapse and no observed Rule ranking improvement. CAN-ID mismatch reproduces the failure, while ID-channel removal reveals additional non-ID shift on can-train and OTIDS.

2 Related Work

2.1 CAN IDS Saturation

CAN IDS methods range from entropy and timing rules [12, 16] to neural sequence models [7, 13]. Many achieve near-perfect public-split scores, while benchmark analyses show that dataset design, vehicle support, and attack realism determine what those scores imply [2]. The Shift Ladder begins at this saturated source setting and measures how augmentation gains change under progressively larger shifts.

2.2 Datasets and Shifts

ROAD emphasizes physically verified attacks and the limitations of older public CAN datasets [18]; can-train-and-test and can-sleuth show that dataset choice and vehicle axis strongly affect measured IDS capability [10, 8]; and multi-dataset studies report cross-dataset degradation [3, 17]. These findings establish cross-dataset failure as a known condition. The Shift Ladder decomposes that condition by tracking the augmentation gain across controlled rungs and then separating ID sufficiency, non-ID necessity, score ranking, and target-normal repair.

2.3 Synthetic Augmentation for CAN IDS

Synthetic security data can balance classes and cover rare patterns, but its value ultimately depends on downstream behavior [14]. In CAN IDS, GIDS uses GAN-based detection and released the Car-Hacking data [15]; CANet introduces Syn-CAN [6]; and recent augmentation work studies OOD and balanced-training settings [20, 19]. Rule-based generation exposes attack structure explicitly, but shared rules between training and evaluation can concentrate gains inside the generator grammar. The Shift Ladder measures that concentration with content and budget controls, generated-variant sensitivity, grammar exits, external FPR, and attribution probes.

3 Shift Ladder Evaluation Protocol

3.1 Data Roles and Threat Scope

We consider injection-style CAN attacks: DoS/flooding, Fuzzy injection, and Gear/RPM spoofing-style construction families. ROAD masquerade captures additionally assume suppression and impersonation of a legitimate ECU; we evaluate them externally but do not synthesize them. The IDS observes 128-frame windows with stride 32 (75% adjacent-window overlap) and predicts either normal/attack or one of four construction families.

Car-Hacking supplies main training, validation, synthetic generation, and in-dataset testing. Its chronological episode split contains 8.39M/2.72M/6.45M train/validation/test rows; the corresponding windows are 262k/85k/202k. OTIDS contributes 144k binary cross-test windows. can-train contributes 4.49M external test windows and a separate ‘train_01’ split for the diagnostic RF native/transfer check. ROAD contributes 856,605 windows from 12 ambient and 29 frame-attributable attack captures; accelerator captures contain no injected frames and are excluded. Model selection uses Car-Hacking exclusively; target-normal and native/transfer experiments serve as diagnostics.

All datasets are mapped to the same schema: timestamp, CAN ID parsed from hexadecimal to an integer, DLC, eight payload bytes, and Δt. Missing payload bytes are zero padded by DLC. Car-Hacking is split before windowing by source-local chronological attack episodes; normal-only sources use contiguous row blocks. Windows never cross split boundaries. A window is binary attack

if any frame is labeled attack; its multiclass label is the majority attack family among attack frames, with class 0 normal. Feature standardization is fit only on real Car-Hacking training windows and then reused unchanged for every generated, stress, and external evaluation. This preserves the source-trained deployment setting and makes external score saturation interpretable rather than silently target-normalized away.

3.2 Generator, Models, and Evaluation Design

The rule-based generator creates synthetic DoS/Fuzzy/Gear/RPM windows from Car-Hacking train-split normal windows. The reported pool is one fixed construction realization (seed 314159) containing 260,000 attack windows balanced over four families. Fixed and sensitivity variants are generated separately from held-out test normals and used only for evaluation. Table 1 summarizes the training and fixed-variant rules; the low/medium/high sensitivity settings jointly vary burst length, step, amplitude, and timing.

Table 1. Training synthetic rules and fixed generated evaluation variants. The fixed variant is a grammar-shared positive control.

Attack	Training rule	Fixed evaluation variant
DoS	ID 0x000; burst 32–90, step 1–2; mostly zero payload; shortened Δt	ID 0x000; burst 48–96, step 1–2; zero payload; $\Delta t \times 0.25$
Fuzzy	random ID 0–2047; DLC 2/5/8; burst 24–88, step 1–4; uniform payload	burst 40–96, step 1–3; DLC 8; uniform payload; short random Δt
Gear	ID 0x43F; burst 32–90, step 1–3; bytes 0/1 level and inverse-level mutation	ID 0x43F; burst 48–96, step 2; byte 2 ramp, byte 3 inverse ramp, byte 4 perturbation
RPM	ID 0x316; burst 32–90, step 1–3; bytes 0/1 RPM-derived and byte 2 perturbation	ID 0x316; burst 48–96, step 2; byte 2 sinusoid, byte 3 ramp, byte 5 perturbation

Each rung measures a distinct scope of robustness. A fixed generated variant tests grammar-shared coverage, while later rungs introduce coupled severity, coordinate, grammar, and domain shifts. Figure 1 shows the data roles and evaluation sequence; Figure 2 then visualizes the measured progression across those rungs.

The main model is a 1D-CNN over 128-by-11 windows with Conv1D 64–128–128 blocks, batch normalization, ReLU, pooling, dropout 0.2, and a five-class head trained by class-weighted cross entropy and AdamW (learning rate 10^{-3}, weight decay 10^{-4}). Training uses batch size 512 and selects the best checkpoint by validation multiclass Macro-F1. The full-ratio comparison uses different maximum schedules: real-only trains for at most 20 epochs with early-stopping patience 4, while augmented and oversampling arms use at most 12 epochs and patience 3. A matched-update analysis therefore compares real-only, Rule +30%, and equal-size real-attack oversampling +30%. Each arm trains for exactly 7,992 updates, is evaluated on the same source-validation grid every 666

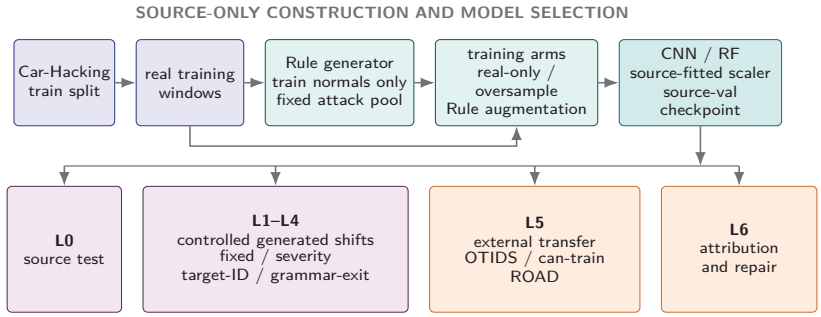


Fig. 1. Leakage-controlled training and evaluation architecture. Car-Hacking training data determine the generator, standardizer, model parameters, and checkpoint. The ladder separates source performance, construction-local generated tests, external transfer, and diagnostic attribution.

updates, and selects the highest validation Macro-F1 checkpoint (earlier update on a tie). Architecture, optimizer, preprocessing, and arm-specific balanced class weights remain unchanged. Fixed and sensitivity outcomes are accessed after all 15 fits finish, and standardization statistics are fit on real Car-Hacking training windows.

Main source/external CNN results use seeds 7/42/123; sensitivity, stress, and matched-update analyses additionally use seeds 2026/3407. These paired pipeline seeds vary initialization, sampling, and minibatch order conditional on one split and one fixed Rule construction. RFs use aggregate window features (mean, std, min, max, last-minus-first over 11 frame features), 160 trees, min leaf 2, and balanced-subsample weights.

We bit-exactly replay the frozen generator to reconstruct base origins and deterministically retain 128-frame-aligned bases. This evaluation-only sensitivity removes within-set raw-frame sharing without retraining or checkpoint reselection; the full evaluations remain primary.

4 Construction-Local Gains Under Matched Training

The saturated Car-Hacking source split provides little separation among training arms: the real-only CNN reaches binary Macro-F1 0.9992 ± 0.0001 at FPR 0.00022 ± 0.00007, and Rule +30% reaches 0.9994 ± 0.0001. Generated variants expose the difference. Real-only fixed-variant recall is 0.4862 ± 0.0284, whereas Rule +10% reaches 1.0000 ± 0.0000. The pattern also appears in an RF: Rule +30% raises fixed-variant recall from 0.5000 to 0.99997 and reaches sensitivity detection recall 0.749 ± 0.020.

The coupled-severity sweep tests coverage beyond the fixed positive control. Table 2 shows detection recall rising from 0.5481 ± 0.0326 to 0.7800 ± 0.0307 at Rule +100%, with FPR differences at the 1/12,000 quantization level. Every seed shows a positive gain, but the ratios are nonmonotone: +50% falls below +30%

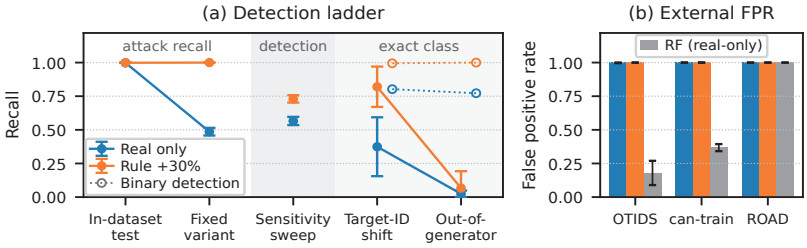


Fig. 2. Shift Ladder summary. The positive gain is concentrated in generated variant rungs; external CNN FPR remains at the ceiling on OTIDS, can-train, and ROAD. Hollow markers on grammar-exit rungs mark binary detection recall, where exact class recall has already collapsed.

within seed variation. Synthetic quantity alone therefore does not determine variant coverage.

Table 2. Variant parameter sensitivity. Detection recall is binary attack recall (mean ± std over seeds 7/42/123/2026/3407).

Setting	Macro-F1	Detection recall	FPR
Real only	0.7273 ± 0.0203	0.5481 ± 0.0326	0.00032 ± 0.00007
Real oversampling +50%	0.7328 ± 0.0290	0.5572 ± 0.0467	0.00037 ± 0.00005
Rule +10%	0.8091 ± 0.0175	0.6822 ± 0.0292	0.00027 ± 0.00004
Rule +30%	0.8352 ± 0.0162	0.7258 ± 0.0273	0.00023 ± 0.00009
Rule +50%	0.8232 ± 0.0339	0.7059 ± 0.0571	0.00027 ± 0.00004
Rule +100%	0.8673 ± 0.0182	0.7800 ± 0.0307	0.00025 ± 0.00000

The matched-update analysis compares training content while holding update count fixed (Table 3). Source-test performance remains saturated: Rule +30% and matched oversampling differ by only $+0.000034 \pm 0.000079$ in paired Macro-F1, and their small FPR differences change sign across seeds. Generated evaluations separate the arms. Rule +30% improves fixed-variant recall by 0.5025 ± 0.0157 over real-only and 0.5125 ± 0.0166 over oversampling; sensitivity recall improves by 0.1690 ± 0.0406 and 0.1919 ± 0.0267, respectively. All four contrasts are positive in every paired seed.

Matching reduces the Rule-minus-real sensitivity gap from 0.1777 to 0.1690, an absolute attenuation of 0.0086. The gain is thus modestly schedule-sensitive, while the Rule-specific advantage persists under this budget. Aggregate sensitivity exact recall rises to 0.6440 ± 0.0751 from 0.4451 ± 0.0315 (real-only) and 0.4209 ± 0.0406 (oversampling). This improvement varies by family: Gear exact recall is 0.5346 ± 0.1560 for Rule and 0.5571 ± 0.1615 for real-only.

Raw-frame-disjoint evaluation retains 15,976 fixed and 7,465 sensitivity windows. Rule-minus-real binary-recall contrasts are 0.5002 ± 0.0165 (full: 0.5025) and 0.1715 ± 0.0399 (full: 0.1690); Rule-minus-oversampling contrasts

Table 3. Matched-update sensitivity (mean ± sample std over five paired pipeline seeds; rounded here, full precision in the artifact). All arms receive 7,992 updates and common source-validation selection; generated sets are evaluation-only.

Source test		
Setting	Macro-F1	FPR
Real only	0.99917 ± 0.00021	0.000169 ± 0.000010
Rule +30%	0.99945 ± 0.00007	0.000126 ± 0.000071
Real-attack oversampling +30%	0.99941 ± 0.00004	0.000145 ± 0.000030
Generated evaluation		
Setting	Fixed recall	Sensitivity recall
Real only	0.4975 ± 0.0157	0.5510 ± 0.0314
Rule +30%	0.9999 ± 0.0001	0.7200 ± 0.0132
Real-attack oversampling +30%	0.4874 ± 0.0165	0.5282 ± 0.0335

are 0.5115 ± 0.0169 and 0.1951 ± 0.0257. All four remain positive in every paired seed and differ from their full-evaluation values by at most 0.0033.

5 Gain Decay Across Severity and Grammar

The gain decays first at low severity. Even at Rule +100%, exact class recall for low Fuzzy, Gear, and RPM is 0.1785, 0.3452, and 0.0021; binary detection recall is 0.2440, 0.5581, and 0.0023. The generator-based gain therefore provides little coverage for these low-severity variants.

Coordinate stresses further locate the learned boundary. Applying the Gear/RPM generator relations to train-normal-supported shifted IDs barely changes Rule +30% retention: Gear $0.9227 \rightarrow 0.9213$ and RPM $0.7128 \rightarrow 0.7114$. Canonical-ID memorization alone therefore does not explain the fixed gain within this construction. The compound L4 stress shifts the ID and moves perturbations outside all disclosed payload positions; exact retention then collapses (Table 4) while nearly all windows remain binary attacks. Rule augmentation has learned an attack boundary that extends across target IDs but carries little exact construction-family information beyond the payload grammar.

Real and synthetic windows are also nearly perfectly separable (AUC 0.999994), while the measured CAN-ID, timing, and payload marginals show small JS divergences of 0.0050, 0.0348, and 0.0391. This combination points to discriminative distributional cues not summarized by those three marginal distances. Equal-added-window real-attack oversampling does not reproduce the fixed-variant gain (0.4951 versus 1.0000 at +50%). Together, these results locate the observed benefit in artifact-aligned boundary shaping around the generated construction.

Table 4. Gear/RPM coordinate stress (exact construction-family recall, mean ± std over seeds 7/42/123/2026/3407). Shifted IDs/bytes are artificial and not DBC-validated vehicle signals; L4 is compound.

Setting	Attack	ID stress canonical	ID stress shifted	L4 out-of-gen shifted
Real only	Gear	0.6889 ± 0.2616	0.6730 ± 0.2606	0.0000 ± 0.0000
	RPM	0.0324 ± 0.0232	0.0311 ± 0.0242	0.0297 ± 0.0358
Rule +30%	Gear	0.9227 ± 0.0472	0.9213 ± 0.0536	0.1612 ± 0.1839
	RPM	0.7128 ± 0.1615	0.7114 ± 0.1620	0.0193 ± 0.0263
Rule +100%	Gear	0.9325 ± 0.0840	0.9325 ± 0.0826	0.1561 ± 0.2918
	RPM	0.7255 ± 0.2257	0.7136 ± 0.2308	0.0238 ± 0.0318

6 External Rungs Reveal Operating-Point Collapse and Support Shift

Support diagnostics distinguish the three datasets before operating-point interpretation. OTIDS normal traffic has top-20 source-ID coverage 0.5683 yet a median attack score of 1.0. can-train has train-normal ID coverage 0.0392 with modest median standardized payload and Δt shifts (0.215 and 0.082), whereas ROAD has ID coverage 0.0287 and larger shifts (0.975 and 0.478). These distinct profiles locate the external shift across multiple support channels rather than a single CAN-ID mechanism. Metric denominators, transforms, and aggregation rules are retained in the artifact.

The external rungs identify two barriers to transfer: the source operating point collapses, and target ranking remains at or below the real-only baseline. On all three datasets, the source policy alarms on nearly every normal window (Table 5). Every paired Rule +30% minus real-only AUROC/AP difference is non-positive, and TPR is zero at empirical FPR ceilings of 0.1% and 1%. The ladder therefore distinguishes a degenerate operating point from absent ranking transfer despite near-perfect attack recall.

Score ties eliminate low-FPR operation. In every one of the 54 dataset–setting–seed cells, the largest normal-score tie exceeds both empirical false-positive budgets; the attainable threshold above 1.0 gives FPR 0 and TPR 0. Real-attack oversampling +50% produces a secondary OTIDS ranking signal (AUROC 0.6764 ± 0.0281, AP 0.6716 ± 0.0510), yet its source-policy FPR is 0.8591 ± 0.1466 and low-FPR TPR remains zero. ROAD capture-macro and can-train file-macro results agree with the pooled pattern.

A native/transfer RF check separates dataset learnability from transfer. A binary RF trained on a logged 300k-window can-train cap reaches FPR 0.003 and recall 0.577 on held-out can-train. The same model reaches recall 0.030 on Car-Hacking and FPR 0.975 on ROAD. can-train is therefore learnable under its own support while its learned attack boundary transfers poorly; the full direction table is retained in the artifact.

CAN-ID mismatch reproduces but does not fully explain the external failure. A learning-free ID whitelist reaches FPR 1.0 on OTIDS, can-train, and

Table 5. Prespecified Rule +30% external diagnostic (mean ± sample std over seeds 7/42/123; rounded here). AP is average precision; attack prevalence is 0.4864/0.0557/0.06992 for OTIDS/ROAD/can-train. Δ is Rule minus real-only; target labels are diagnostic only.

Source operating point

Dataset	Rule FPR	Rule recall
OTIDS	0.99997 ± 0.00004	0.99993 ± 0.00009
ROAD	1.0000 ± 0.0000	1.0000 ± 0.0000
can-train	0.999997 ± 0.000004	0.99985 ± 0.00026

Target-label ranking diagnostic

Dataset	Rule AUROC	Rule AP	ΔAUROC	ΔAP	TPR@0.1% / 1% FPR
OTIDS	0.5011 ± 0.0035	0.4870 ± 0.0018	-0.0484	-0.0458	0 / 0
ROAD	0.5000 ± 0.0000	0.0557 ± 0.0000	-0.0167	-0.0019	0 / 0
can-train	0.4999 ± 0.0004	0.06990 ± 0.00005	-0.0002	-0.00002	0 / 0

ROAD while preserving Car-Hacking test behavior, establishing ID mismatch as a sufficient vocabulary-style failure. CNNs retrained without CAN ID still reach high FPR on can-train and OTIDS, so non-ID channels also carry dataset shift. ROAD attribution instead uses support, calibration, and OOD probes. ID-only CNN results are unstable and remain qualitative diagnostics.

Repair probes confirm the dataset dependence. ROAD ambient-threshold calibration leaves held-out FPR 0.939 ± 0.103 (real-only) or 1.0 (Rule), and target restandardization gives 1.0. Max-logit/energy lowers FPR to 0.07–0.08 but inverts attack/ambient ranking (AUROC 0.31–0.38). Adding 100k can-train calibration-normal windows lowers can-train FPR to 0.00034 while preserving source performance, but recall remains 0.090. Target-normal synthetic attacks reach recall 0.173 at FPR 0.0012 and leave ROAD saturated. RF repair and source-fitted entropy/timing baselines show the same separation between local false-positive repair and attack transfer.

7 Discussion

The combined evidence locates the augmentation gain at the generator construction. Under a common data volume and optimizer-update budget, the Rule arm retains a large advantage over real-only and equal-size real-attack oversampling on generated variants. Coupled severity carries part of that advantage beyond a fixed positive control, while low-severity failures, near-perfect real/synthetic separability, and the compound grammar exit expose its boundary. The external rungs then separate degenerate source-policy operation from absent target-ranking improvement.

The Shift Ladder turns this localization into a reporting discipline. A synthetic-augmentation study should fit generators and scalers on training

data, compare synthetic content with equal-volume real data under a matched budget, audit overlapping evaluation units and synthetic separability, cross a generator-grammar boundary, and report external FPR before attack recall. External operating-point and ranking analyses answer separate questions: the OTIDS oversampling result shows that improved ranking can coexist with an unusable source policy. Together, the rungs replace a single robustness label with a sequence of explicit, testable claims.

8 Threats to Validity

The construction scope comprises one fixed Rule pool (seed 314159), fixed and sensitivity variants that share its high-level grammar, and coupled-severity settings that vary several parameters jointly. Gear/RPM labels denote construction families, and shifted coordinates lack DBC or vehicle-physics validation. L4 therefore measures a compound grammar exit rather than payload-position causality; near-perfect real/synthetic separability limits the mechanism claim to artifact-aligned boundary shaping around these constructions.

Adjacent windows overlap by 75%, and the source-local episode/time split is not capture-disjoint. The raw-frame-disjoint sensitivity removes within-evaluation reuse but does not create independent captures. Seed variation remains conditional on one split and Rule construction. ROAD capture-macro and can-train file-macro analyses support the pooled result; OTIDS scenario files are class-separated. The uncertainty is therefore pipeline-level rather than vehicle-population-level.

The evaluated scope covers one source dataset, one Rule construction, one 7,992-update budget and validation cadence, and CNN/RF models. External label structures determine the available metrics: OTIDS supports binary cross-testing, can-train stems do not map to Car-Hacking multiclass labels, and ROAD labels are reconstructed from published injection intervals, target IDs, and payload patterns. Model fitting and selection remain source-only, while target-label-dependent metrics are diagnostic. Non-ID attribution covers can-train and OTIDS; ROAD contributes calibration, OOD, and repair evidence. Learned generators, other architectures, construction replications, and byte-level mechanisms require separate evaluation.

9 Conclusion

The Shift Ladder localizes how far a synthetic-augmentation gain survives progressively larger shifts. Under matched training and overlap-aware evaluation, rule augmentation produces repeatable controlled-variant coverage. Low-severity and compound grammar-exit rungs reduce that coverage, while the three external rungs expose degenerate source-policy FPR and no Rule ranking gain. The contribution is both empirical—a construction-local gain with a measured failure boundary—and methodological: a protocol that replaces a single robustness score with explicit, testable scope claims.

Artifact Availability

To make the Shift Ladder auditable, we provide a public artifact package at `https://github.com/flyho77/shift-ladder-can-ids-artifact`. The package contains the experiment configuration, preprocessing and evaluation scripts, source result tables, paper-ready figures, dataset download and provenance instructions, and a manifest mapping reported tables and figures to generated artifacts. Raw datasets and trained weights are not redistributed.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00402782, Development of virtual commissioning technology that interconnects manufacturing data for global expansion).

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

References

1. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: Proceedings of the 31st USENIX Security Symposium. pp. 3971–3988 (2022)
2. Blevins, D.H., Moriano, P., Bridges, R.A., Verma, M.E., Iannacone, M.D., Hollifield, S.C.: Time-based can intrusion detection benchmark. In: Proceedings of the 2021 Workshop on Automotive and Autonomous Vehicle Security (2021). https://doi.org/10.14722/autosec.2021.23013
3. Dupont, G., den Hartog, J., Etalle, S., Lekidis, A.: Evaluation framework for network intrusion detection systems for in-vehicle can. In: Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo. pp. 1–6 (2019). https://doi.org/10.1109/ICCVE45908.2019.8965028
4. Hacking and Countermeasure Research Lab: Can-intrusion dataset (otids). `https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset` (2017), accessed 2026-06-10
5. Hacking and Countermeasure Research Lab: Car-hacking dataset. `https://ocslab.hksecurity.net/Datasets/car-hacking-dataset` (2017), accessed 2026-06-10
6. Hanselmann, M., Strauss, T., Dormann, K., Ulmer, H.: Canet: An unsupervised intrusion detection system for high dimensional can bus data. IEEE Access **8**, 58194–58205 (2020). https://doi.org/10.1109/ACCESS.2020.2982544
7. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLOS ONE **11**(6), e0155781 (2016). https://doi.org/10.1371/journal.pone.0155781
8. Kidmose, B., Kidmose, A., Meng, W.: can-sleuth: Sleuthing out the capabilities, limitations, and performance impacts of automotive intrusion detection datasets. International Journal of Information Security (2025). https://doi.org/10.1007/s10207-025-01038-8

9. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S.: Experimental security analysis of a modern automobile. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy. pp. 447–462 (2010). https://doi.org/10.1109/SP.2010.34
10. Lampe, B., Meng, W.: can-train-and-test: A curated can dataset for automotive intrusion detection. Computers & Security **140**, 103777 (2024). https://doi.org/10.1016/j.cose.2024.103777
11. Lee, H., Jeong, S.H., Kim, H.K.: Otids: A novel intrusion detection system for in-vehicle network by using remote frame. In: Proceedings of the 15th Annual Conference on Privacy, Security and Trust. pp. 57–66 (2017). https://doi.org/10.1109/PST.2017.00017
12. Müter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: 2011 IEEE Intelligent Vehicles Symposium (IV). pp. 1110–1115 (2011). https://doi.org/10.1109/IVS.2011.5940552
13. Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M.: Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys **55**(11), 237:1–237:40 (2023). https://doi.org/10.1145/3570954
14. 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
15. Seo, E., Song, H.M., Kim, H.K.: Gids: Gan based intrusion detection system for in-vehicle network. In: Proceedings of the 16th Annual Conference on Privacy, Security and Trust. pp. 1–6 (2018). https://doi.org/10.1109/PST.2018.8514157
16. Song, H.M., Kim, H.R., Kim, H.K.: Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In: 2016 International Conference on Information Networking (ICOIN). pp. 63–68 (2016). https://doi.org/10.1109/ICOIN.2016.7427089
17. Vahidi, A., Rosenstatter, T., Mowla, N.I.: Systematic evaluation of automotive intrusion detection datasets. In: Proceedings of the 6th ACM Computer Science in Cars Symposium. pp. 6:1–6:12 (2022). https://doi.org/10.1145/3568160.3570226
18. Verma, M.E., Bridges, R.A., Iannacone, M.D., Hollifield, S.C., Moriano, P., Hespeler, S.C., Kay, B., Combs, F.L.: A comprehensive guide to can ids data and introduction of the road dataset. PLOS ONE **19**(1), e0296879 (2024). https://doi.org/10.1371/journal.pone.0296879
19. Wang, X., Zhao, J., Liu, P., Yao, N., Xu, Z.: Can bus intrusion detection based on deep learning with data augmentation for connected autonomous vehicles. IEEE Transactions on Vehicular Technology **75**(2), 2253–2266 (2026). https://doi.org/10.1109/TVT.2025.3603056
20. Zhao, Q., Chen, M., Gu, Z., Luan, S., Zeng, H., Chakraborty, S.: Can bus intrusion detection based on auxiliary classifier gan and out-of-distribution detection. ACM Transactions on Embedded Computing Systems **21**(4) (2022). https://doi.org/10.1145/3540198

Unprotected at Access: Testing Random Access Response Processing in 5G Basebands

Hazem Issa*, Quang Huy Pham*, Beomseok Oh, Mincheol Son, Kwangmin Kim, Duckwoo Kim, Tuan Dinh Hoang, and Yongdae Kim

Korea Advanced Institute of Science and Technology (KAIST)
{hazem, huy.phamquang, beomseoko, mcson, kwangmin.kim, 0731kdw, tuan.hoangdinh, yongdaek}@kaist.ac.kr

Abstract. While 5G introduces strong security mechanisms after authentication, messages exchanged before security-context establishment remain without encryption or integrity protection. One such message is the Random Access Response (RAR), which is processed during the initial access procedure. Lack of protection in this early-stage message creates an attack surface for adversaries capable of delivering crafted RAR messages over the air.
In this work, we present a specification-guided over-the-air testing framework for identifying crash-inducing baseband failures in 5G RAR processing. Unlike prior approaches that rely on random mutation of simple RAR messages, our framework uses specification-derived rules to generate diverse configurations with syntax-level and field-semantic violations, then delivers them to commercial UEs through a modified open-source 5G base station. We evaluate our framework on four commercial 5G UEs using MediaTek and Samsung Exynos baseband platforms. Our testing identifies a MediaTek memory-corruption vulnerability triggered by a syntactically valid RAR carrying an abnormal value, causing a baseband crash and temporary denial of service. These findings show that pre-authentication RAR processing is an important attack surface that warrants systematic specification-guided robustness testing.

Keywords: 5G Baseband Security · Random Access · OTA Testing

1 Introduction

In recent years, 5G networks have been widely deployed and increasingly support smartphones, IoT devices, and industrial systems. To access these networks, user equipment (UE) relies on a dedicated 5G baseband processor that implements cellular protocol stacks and processes radio messages received over the air. Due to the open nature of wireless interfaces, implementation vulnerabilities in baseband processors can expose security-critical attack surfaces to remote attackers.

One of the initial messages that 5G basebands receive during network access is the RAR message [3], which is carried at the Medium Access Control (MAC)

* Both authors contributed equally to this research.

layer. During the Random Access (RA) procedure, the UE transmits a preamble and the base station responds with a RAR carrying parameters for subsequent communication. If the UE fails to decode this message, radio connection establishment cannot complete, forcing the UE to retry the access procedure.

Unfortunately, the RAR message is exchanged during the earliest stages of the connection, before the baseband establishes a security context with the network [1]. Consequently, it is transmitted without encryption or integrity protection. This makes RAR processing security-critical: an attacker capable of delivering a crafted RAR message during the random-access window can cause the baseband to process attacker-controlled input before authentication is available. Failures can arise during parsing, when the baseband disassembles the RAR message according to its grammar, or after parsing, when decoded field values are consumed to update modem state. Such failures can cause denial of service and, in more severe cases, may expose memory-safety errors that enable stronger compromise of the baseband processor.

Despite the critical implications of memory-safety vulnerabilities in baseband firmware, 5G RAR processing has received limited attention. Prior work has largely focused on earlier generations such as LTE [10,7], and existing 5G OTA testing relies mainly on random mutation of simple messages generated by open-source base stations [4]. These inputs are useful for basic testing, but they cover only a limited range of RAR structures and field-semantic edge cases, leaving specification-guided RAR testing of commercial 5G basebands underexplored.

To address this gap, we propose a specification-guided OTA testing framework for identifying crash-inducing failures in 5G RAR processing on commercial basebands, including memory-corruption vulnerabilities. Through manual analysis of the 3GPP MAC and physical layer (PHY) specifications, we extract reachability, grammar, and field-semantic constraints that guide the generation of controlled RAR test cases. The generated test cases are delivered over the air through a modified 5G base station, while Android radio logs provide a crash oracle. Since OTA testing provides only external visibility, crash candidates are confirmed through reproduction and post-analysis.

We evaluate our framework on four commercial 5G UEs covering MediaTek and Samsung Exynos baseband platforms. Our testing identifies a memory-corruption vulnerability in two MediaTek-based devices, where a syntactically valid RAR message carrying a reserved Timing Advance value triggers a baseband crash and temporary denial of service.

To summarize, our contributions are as follows:

- We design a specification-guided RAR generator that produces diverse test cases with controlled grammar and field-semantic violations.
- We implement an automated OTA testing framework for RAR messages using a modified 5G base station.
- We evaluate our framework on four commercial 5G UEs and identify a memory-corruption vulnerability in two MediaTek basebands.

2 Related Work

Prior work on finding memory corruption vulnerabilities in cellular basebands generally falls into three categories: static reverse-engineering, emulation-based dynamic testing, and over-the-air runtime fuzzing.

Static reverse-engineering [9] and emulation-based approaches [6] primarily target higher-layer protocols such as Non-Access Stratum (NAS) and Radio Resource Control (RRC). These approaches provide deep firmware visibility and support root-cause analysis, but they require substantial expertise and offline access to firmware images. Although recent emulation-based work has extended fuzzing to Layer-2 attack surfaces such as GPRS [5], such efforts remain limited to legacy protocols and have not demonstrated support for 5G NR RAR processing under realistic lower-layer conditions.

OTA runtime fuzzing addresses some of these limitations by testing real devices under realistic wireless conditions. Early work demonstrated this approach on LTE control-plane protocols [10] and LTE lower-layer implementations [7], showing that commercial basebands can expose remotely triggerable failures observable during live protocol execution. More recent work has extended OTA fuzzing to 5G device testing [4]. However, its RAR-related testing mainly relies on random mutation of simple RAR messages generated by an open-source base station. Such inputs are useful for basic robustness testing, but they provide limited control over which parts of RAR processing are exercised and may fail to reach deeper handling logic in a systematic way. Separately, semantics-guided testing has been applied to pre-authentication 5G control-plane messages [8], but this line of work targets field semantics at the RRC layer rather than MAC RAR processing. As a result, systematic testing of 5G RAR processing remains underexplored.

Our approach complements OTA fuzzing by making RAR test generation specification-guided rather than relying only on mutations of simple baseline messages. In particular, we separate the constraints needed for a crafted RAR to reach UE-side processing from the grammar and field-semantic constraints used as mutation targets. This lets us construct controlled tests for different parts of RAR processing while preserving the conditions needed for OTA delivery.

3 Preliminaries

3.1 Background

5G overview.

Figure 1a illustrates the overall cellular network architecture and protocol stack. 5G architecture consists of User Equipment (UE), Next Generation Node B (gNB), and the Core Network (5GC). UE is the end-user device that accesses the 5G network through radio interface. The gNB provides wireless connectivity between the UE and the 5G Core Network (5GC). Inside the UE, the Application Processor (AP) runs the operating system, while the Baseband Processor (BP)

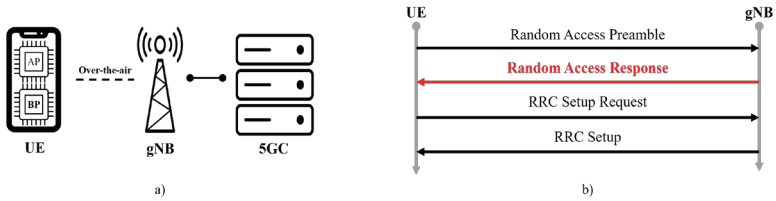


Fig. 1. a) 5G Architecture b) 5G Random Access Procedure

implements the cellular protocol stack and processes radio messages from the gNB.

5G Random Access Procedure. The UE initiates the RA procedure to establish a radio connection with gNB (Figure 1b) [3]. It first transmits a randomly selected preamble on the Physical Random Access Channel (PRACH), after which the gNB responds with a RAR containing one or more Random Access Preamble Identifiers (RAPIDs) and corresponding payload entries. The UE parses the RAR; if a RAPID matches its preamble, it applies the corresponding payload and transmits a RRC Setup Request (Msg3) using the granted uplink resources, allowing the RA procedure to continue.

Memory corruption vulnerability. Memory corruption vulnerabilities arise from programming errors such as missing bounds checks, unsafe pointer use, or flawed input parsing. In UE basebands, they are security-critical because the BP processes over-the-air messages before authentication, including messages that an attacker may spoof or overshadow. If not properly validated, such inputs can corrupt memory and lead to denial of service, information leakage, or arbitrary code execution.

3.2 Threat model

We consider an adversary whose goal is to disrupt a victim UE's cellular service by triggering a crash-inducing RAR-processing vulnerability. We assume that the adversary knows only the victim's coarse location (i.e. the cell in which the UE is camped), has no physical access to the device, and does not possess the subscriber's secret keys. The adversary can transmit crafted RAR messages during the pre-security random-access procedure using a modified open-source base-station stack with a software-defined radio (SDR) as the RF front end. This can be done either by operating as a fake base station (FBS) or by performing a signal overshadowing attack [12].

4 Testing Methodology

Our methodology, depicted in Figure 2, has four steps: specification analysis (§4.1), test-case generation (§4.2), OTA execution (§4.4), and post-analysis (§4.5). We first analyze the 5G MAC/PHY layer specifications [3,2] to extract

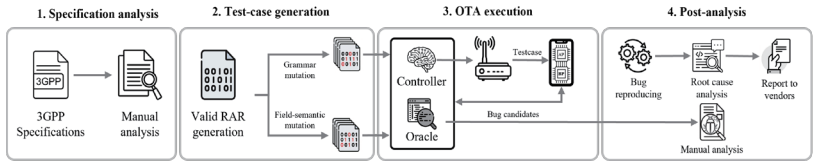


Fig. 2. Design overview

RAR-processing constraints; use these constraints to generate specification-guided RAR test cases; inject the test cases during the UE's random-access procedure; monitor crash candidates through radio logs; reproduce and analyze candidates during post-analysis; and report confirmed findings to the vendor.

4.1 Specification analysis

We analyze the specifications in the order a UE handles a RAR message. A crafted RAR must first be delivered under conditions where the UE will process it, then follow a byte-level grammar that the parser can disassemble, and finally carry decoded field values that are applied to UE state. This gives three specification-derived constraint categories for test generation: reachability constraints, grammar constraints, and field-semantic constraints.

Reachability constraints. Reachability constraints determine whether a crafted RAR can exercise UE-side RAR handling in the intended RA attempt. We identify two reachability requirements. First, the RAR must be delivered within RA window, when the UE is monitoring for a response during the random access procedure; after completion or timeout, later RARs are ignored. Second, for RAPID-bearing subPDUs, the RAPID must match the preamble ID transmitted by the UE in the preceding PRACH; otherwise, the corresponding payload is discarded as unrelated.

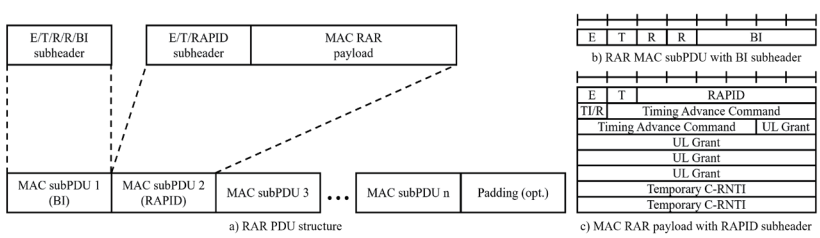


Fig. 3. RAR message structure

Grammar constraints. Once a RAR is delivered during the RA window, the baseband must parse its MAC RAR PDU structure. As shown in Figure 3, a MAC RAR PDU consists of one or more MAC RAR subPDUs. The specification defines the structure of each subPDU and its chaining rules, but does not

define a RAR-specific upper bound on the number of subPDUs beyond lower-layer transport constraints. Each subPDU contains a subheader and, depending on the subheader type, an optional payload. A Backoff Indicator (BI) subheader carries no payload, while a RAPID subheader is followed by a RAR payload containing the Timing Advance (TA), uplink grant, and Temporary C-RNTI (TC-RNTI). Each subheader also contains an Extension (E) bit indicating whether another subPDU follows. From this structure, we extract grammar constraints for subheader types, payload presence, payload length, E-bit chaining, and valid subPDU layouts.

Field-semantic constraints. Once a RAR is parsed, decoded fields must be validated against specification-defined constraints before being applied to UE state. Our analysis identifies four RAR fields with post-parser constraints relevant to test generation: BI, TA, UL grant, and TC-RNTI. For BI, TA, and TC-RNTI, the specification defines explicit value domains with valid and reserved ranges. The UL grant is context-dependent: its validity is not captured by a single global numeric range, but by how the encoded grant is interpreted under the current serving-cell configuration. Table 1 summarizes the extracted constraints for these fields.

Tab. 1. Extracted RAR field-semantic constraints.

Field	Type	Specification-derived constraint
BI	Explicit domain	4-bit index; 0–13 valid, 14–15 reserved.
TA	Explicit domain	12-bit command; 0–3846 valid, 3847–4095 reserved.
UL grant	Structured encoding	27-bit PHY-defined scheduling grant; interpreted according to PHY scheduling context.
TC-RNTI	Explicit domain	16-bit identity; 0x0001–0xFFF2 valid, 0x0000 N/A, 0xFFF3–0xFFFF reserved or used in special RNTIs.

4.2 Test-case generation

Test-case generation follows directly from the constraint categories identified in §4.1. Reachability constraints determine whether a crafted RAR can reach UE-side processing, so they are preserved rather than used as mutation targets; runtime-dependent reachability conditions are bound later during OTA execution §4.4. The remaining two constraint categories define our mutation targets. Parser tests violate grammar constraints to exercise RAR parsing and structural validation, while field-semantic tests preserve grammar but violate field constraints to exercise post-parser logic.

Valid RAR seed generation. We first generate a diverse corpus of structurally and semantically valid RAR seeds that satisfy grammar and field-semantic constraints. The corpus varies the number and composition of MAC RAR subPDUs, including BI-only subPDUs, RAPID subPDUs carrying RAR payloads, mixed

chains, and long but OTA-deliverable subPDU sequences. We include long sequences to exercise implementation limits while staying within the valid input space defined by the extracted grammar constraints. For context-dependent field semantics such as the UL grant, we instantiate valid seeds with values derived from the base station's current serving-cell configuration.

Parser tests. Parser tests target the logic that disassembles the MAC RAR PDU into subheaders and payloads. Starting from valid seeds, these tests intentionally violate the grammar while keeping the resulting PDU OTA-deliverable. The goal is to exercise structural validation before decoded RAR fields are applied. We apply targeted mutations that break byte-level framing assumptions, including truncating RAR payloads after RAPID subheaders, truncating multi-subPDU chains at different byte offsets, setting an Extension bit that indicates another subPDU when no bytes remain, and constructing subPDU layouts that violate the extracted grammar. These tests check whether the implementation validates subheader chaining, payload presence, and remaining-buffer length before accessing each indicated payload.

Field-semantic tests. Field-semantic tests target the post-parser logic that consumes decoded RAR fields after successful parsing. These tests preserve the grammar and intentionally violate only field-semantic constraints, making reproducible failures easier to attribute to post-parser behavior rather than malformed packet structure. We mutate one field at a time according to the field-semantic constraints in Table 1. For fields with explicit valid and reserved domains, such as BI, TA, and TC-RNTI, we generate boundary values, reserved values, and values near specification-defined transitions. For the context-dependent UL grant, we treat the seed value as the semantic anchor and perturb fields within the UL-grant encoding.

4.3 Oracle

The oracle determines whether an injected RAR test case causes the UE baseband to crash during OTA execution. Since OTA testing lacks memory-level instrumentation, we leverage Android radio logs exposed through ADB to monitor modem-failure indicators, following prior work [7]. These logs are available on commercial Android devices without root access or firmware modification.

Our crash indicators include `RADIO_OFF_OR_UNAVAILABLE`, `Modem Reset`, and `Every body panic!`. Compared with generic symptoms such as temporary disconnection or acknowledgment failure, these strings provide a stronger signal that the baseband has entered an abnormal failure state. However, an oracle hit is not proof of memory corruption: a crash may also result from a reachable assertion, an explicit panic path, a watchdog reset, or another controlled failure. We therefore treat each observed indicator as a crash candidate and defer root-cause classification to post-analysis §4.5.

4.4 OTA execution

In this stage, the framework turns the generated test cases into live over-the-air experiments against commercial UEs. OTA execution has three main responsibil-

ities: it drives the UE into a fresh RA attempt, delivers each generated test case under the required reachability conditions, and invokes the oracle to monitor for externally visible modem-failure indicators.

For each test case, the controller first triggers random access by cycling airplane mode via ADB commands. When the base station detects the resulting PRACH, the framework records the UE's preamble ID. When RAPID matching is required, it sets the RAPID field to the observed preamble ID. The framework then replaces the scheduled RAR payload in the downlink scheduler with the generated test case and transmits the crafted response within the RA response window. At this point, the transmitted test case is bound to the current RA attempt and satisfies the applicable reachability constraints.

After injection, the framework invokes the oracle described in §4.3 to monitor for baseband-crash indicators. When an indicator is detected, the framework records the indicator string, timestamp, ADB logs, base-station outputs, and the most recent 5 test cases to account for delayed crash manifestation. Finally, the controller cycles airplane mode to recover the UE from incomplete RA attempts or modem instability and trigger the next random-access attempt, allowing the campaign to proceed without manual intervention.

4.5 Post-analysis

Post-analysis turns crash candidates into confirmed crash findings. We first replay the exact test case multiple times to filter out transient failures. Because each test case applies a single mutation, reproducible crashes can be associated with a specific grammar violation or field-semantic violation. We then vary the mutated field or structural pattern to identify affected ranges, boundary conditions, and nearby non-crashing cases.

Finally, we inspect available diagnostic logs to classify the likely failure cause. We label a finding as memory corruption only when supported by diagnostic evidence or vendor confirmation. The resulting crash report, including the triggering test case, reproduction steps, affected boundary conditions, and root-cause hypothesis, is submitted to the baseband vendor for analysis and patching.

5 Implementation

We implemented our testing framework by extending the open-source 5G base station srsgNB [11]. srsgNB supports the 5G standalone (SA) random access procedure, allowing us to connect commercial UEs and deliver crafted RAR messages over the air.

Our implementation adds three main components. First, we added a C++ RAR construction module for generating the parser and field-semantic test cases described in §4.2. The module emits configurable RAR PDUs with controllable subPDU composition, E-bit chaining, payload length, and field values.

Second, we modified the RAR scheduling path to deliver generated test cases during live random access, as required by §4.4. When srsgNB detects a PRACH transmission from the target UE, the framework replaces the normal scheduled

Tab. 2. Tested commercial 5G UEs and discovered vulnerability.

Vendor	Smartphone	Baseband	Version	Finding
MediaTek	Xiaomi Redmi Note 14 Pro 5G	Dimensity 7300 Ultra	MOLY.NR17.R1. MP3.TC8.PR4. SP.V2.P62	High reserved TA crash
MediaTek	Xiaomi 14T	Dimensity 8300 Ultra	MOLY.NR17.R1. MP3.TC8.PR3. SP.V1.P72	High reserved TA crash
Samsung	Google Pixel 8 Pro	Exynos Modem 5300	BP3A.251105.015	-
Samsung	Google Pixel 9 Pro XL	Exynos Modem 5400	BP3A.251105.015	-

RAR response with the selected test case. Because generated test cases may differ in length from standard RAR responses, we also adjusted the scheduler's buffer allocation logic to accommodate custom MAC RAR PDUs. The final packet is inserted immediately before PHY-layer processing. This late insertion point prevents later MAC-layer code from rewriting, normalizing, or rejecting the crafted contents before transmission.

Finally, we added an automation thread that coordinates UE control and failure monitoring. The thread cycles airplane mode via ADB, monitors the crash oracle, and advances through the test-cases without manual intervention.

6 Results

6.1 Experiment setup

We evaluate our framework in a controlled 5G SA testbed. The testbed consists of a PC running our modified srsgNB, a USRP B210 software-defined radio as the RF front-end, and a commercial 5G UE connected to the PC through USB for ADB-based control and logging. The PC is equipped with an Intel Core i7-14700K CPU and 32GB of RAM.

Each experiment executes the generated corpus using the OTA procedure described in §4.4. All OTA experiments are conducted on author-owned devices inside a Faraday cage, preventing test signals from leaking into the surrounding radio environment or interfering with commercial networks and nearby devices.

6.2 Tested devices

We evaluate four commercial 5G UEs covering two baseband platforms: MediaTek and Samsung Exynos. Each device is tested with the same generated RAR test-case corpus. Table 2 summarizes the tested devices and observed outcomes.

Our results show that the two MediaTek-based devices, the Redmi Note 14 Pro 5G and Xiaomi 14T, exhibit reproducible baseband crashes. In contrast, the two Samsung Exynos-based devices complete the testing campaign without triggering the crash indicators monitored by our oracle.

6.3 Testing performance

We generate 761 RAR test cases and execute the same corpus on each tested device. To characterize testing performance, we measure the average time required for a non-crashing test case and the total time required to complete the full corpus on each device.

For non-crashing executions, a single test case completes in approximately 1 second in the steady state. This includes the OTA execution cycle described in §4.4, from triggering a fresh RA attempt to observing the oracle output. Execution time increases when a test case causes the UE to crash or hang, because the framework must wait for the baseband to recover before continuing. In our experiments, the MediaTek devices require approximately 25–35 seconds to recover from each crash.

Overall, executing the full test-case corpus took 32, 25, 42, and 46 minutes on the Xiaomi Redmi Note 14 Pro 5G, Xiaomi 14T, Pixel 8 Pro, and Pixel 9 Pro XL, respectively.

6.4 Discovered vulnerability

Triggering input. The crash is triggered by mutating only the TA field of an otherwise syntactically valid RAR. The triggering RAR contains a RAPID subheader matching the UE's preamble, a complete 7-byte payload, and non-mutated UL grant and TC-RNTI values. Because the RAR satisfies the grammar, the failure is reached after parsing, when the modem consumes the decoded reserved TA value. We therefore classify the issue as a field-semantic validation failure, rather than a parser failure.

Crash boundary. Table 3 shows that the observed crash boundary does not coincide with the specification boundary for reserved TA values. Some reserved values are processed without failure, while larger reserved values reliably trigger the crash. This gap suggests that reserved TA values are not rejected by an initial field-semantic check and may instead reach later post-parser logic.

Tab. 3. Observed crash boundary for RAR Timing Advance values.

TA value	Spec status	Observed behavior
0–3846	Valid	No crash
3847–3927	Reserved	No crash
3928–4095	Reserved	**Crash**

Failure observation. After receiving the triggering RAR, the UE does not continue the RA procedure. Android radio logs report modem-failure indicators detected by our oracle, and the device temporarily loses cellular connectivity. The baseband then recovers and reattaches after approximately 25–35 seconds.

6.5 Implications

The discovered vulnerability shows that RAR processing is a security-critical pre-authentication attack surface, not merely a simple MAC-layer setup step. A syntactically valid RAR carrying a reserved TA value reaches the modem's post-parser field-handling logic and triggers a memory-corruption crash. Because RAR is processed before security-context establishment, this code path is reachable without subscriber credentials by an attacker capable of delivering a crafted RAR during the random-access window.

7 Discussion

Lessons learned. A practical lesson from this study is that specification guidance is critical for effective testing of early-stage RAR processing. RAR inputs contain constraints with different roles. Specification analysis makes these roles explicit, allowing test cases to preserve the constraints needed to reach the target logic while deliberately violating the constraints that exercise it. Random mutation lacks this distinction and can accidentally produce inputs that are discarded before the intended processing stage. Specification guidance therefore turns RAR mutation from arbitrary byte changes into controlled tests of distinct parts of the RAR processing path.

Limitations. First, the framework currently targets 5G MAC RAR messages. Extending it to RLC, PDCP, RRC, or NAS would require layer-specific grammars, state constraints, and mutation strategies. However, the underlying design principles, including grammar-aware generation, OTA delivery, crash monitoring, and automated UE control, are not specific to RAR and can be adapted to test other pre-security control-plane messages.

Second, although MediaTek confirmed the discovered issue as memory corruption, we do not claim remote code execution or persistent compromise. Assessing exploitability beyond denial of service would require reverse engineering and deeper firmware-level analysis, which is outside the scope of this work.

Responsible disclosure. We reported the findings to MediaTek through their vulnerability disclosure program. The affected firmware versions were the latest available security vendor updates for the evaluated devices at testing time. MediaTek confirmed that the crash corresponds to a memory-corruption vulnerability and did not disclose root-cause or exploitability details. MediaTek further stated that the issue duplicated an internally reported issue and had already been fixed in the latest product.

8 Conclusion

A single pre-authentication MAC RAR message can expose a security-critical baseband attack surface. To systematically test this surface, we developed an OTA framework for 5G RAR processing. The framework uses specification-derived constraints to generate and deliver diverse RAR test cases with syntactic

and field-semantic violations, allowing us to exercise distinct RAR-processing stages in a controlled way. Testing four commercial 5G UEs, we identified a field-semantic validation failure in two MediaTek basebands: a reserved Timing Advance value triggers a baseband crash and temporary denial of service. Overall, our results highlight the value of specification-guided RAR testing for systematically exploring RAR processing. They suggest that even small pre-authentication control messages in the random access procedure deserve careful robustness testing.

Acknowledgements.

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. 3GPP. TS 33.501, v19.6.0: 5G; Security architecture and procedures for 5G System (2026)
2. 3GPP. TS 38.213, v19.3.0: 5G; NR; Physical Layer Procedures for Control (2026)
3. 3GPP. TS 38.321, v19.2.0: 5G; NR; Medium Access Control (MAC) protocol specification (2026)
4. Garbelini, M.E., Shang, Z., Luo, S., Chattopadhyay, S., Sun, S., Kurniawan, E.: 5Ghoul: Unleashing Chaos on 5G Edge Devices via Stateful Multi-Layer Fuzzing. IEEE Transactions on Dependable and Secure Computing (TDSC) (2025)
5. Goos, D., Muench, M.: Overcoming State: Finding Baseband Vulnerabilities by Fuzzing Layer-2. Black Hat USA (2024)
6. Hernandez, G., Muench, M., Maier, D., Milburn, A., Park, S., Scharnowski, T., Tucker, T., Traynor, P., Butler, K.: FIRMWIRE: Transparent Dynamic Analysis for Cellular Baseband Firmware. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2022)
7. Hoang, T.D., Oh, T., Park, C., Yun, I., Kim, Y.: LLFuzz: An Over-the-Air Dynamic Testing Framework for Cellular Baseband Lower Layers. In: Proceedings of the 34th USENIX Security Symposium (USENIX Security) (2025)
8. Huang, Q., Wang, X., Guo, W., Gu, G., Hu, H.: Semantics Over Syntax: Uncovering Pre-Authentication 5G Baseband Vulnerabilities. In: Proceedings of the 35th USENIX Security Symposium (USENIX Security) (2026)
9. Kim, E., Kim, D., Park, C., Yun, I., Kim, Y.: BaseSpec: Comparative Analysis of Baseband Software and Cellular Specifications for L3 Protocols. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2021)
10. Kim, H., Lee, J., Lee, E., Kim, Y.: Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane. In: IEEE Symposium on Security and Privacy (S&P) (2019)
11. Software Radio Systems: srsRAN Project: Open Source 5G RAN. https://www.srsran.com/ (2026), accessed: 2026-06-04
12. Yang, H., Bae, S., Son, M., Kim, H., Kim, S.M., Kim, Y.: Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE. In: Proceedings of the 28th USENIX Security Symposium (USENIX Security) (2019)

Detecting LiDAR Spoofing Attacks via NDT-based Inter-Frame Anomalies

Jihun Choi and Wonsuk Choi[0000−0003−3253−4827] ⋆

School of Cybersecurity, Korea University, Seoul, Republic of Korea
{myami,beb0396}@korea.ac.kr

Abstract. LiDAR is a core sensor that enables three-dimensional environmental perception in autonomous driving systems, but its point clouds can be distorted by physical attacks such as external laser injection. In this paper, we focus on the temporal consistency maintained between consecutive LiDAR point cloud frames under normal driving conditions, proposing an attack detection method that repurposes the Normal Distributions Transform (NDT), conventionally used for registration and localization, as a novel detection indicator. The proposed method relies on a single LiDAR sensor and determines whether an attack has occurred by analyzing inter-frame variations, reusing the output of a registration step that the vehicle already performs for localization rather than introducing a dedicated analysis stage. Experimental results show that the proposed method can detect external laser-based injection and removal attacks with low computational overhead.

Keywords: Automobile security · LiDAR Security · Sensor Security · Autonomous Driving · Normal Distributions Transform.

1 Introduction

LiDAR is a sensor that measures distance using laser signals and represents the surrounding environment as a three-dimensional point cloud. Because it can precisely capture the position and shape of nearby objects, it is widely used as a core sensor for object recognition and collision avoidance in autonomous driving systems. In particular, since its effectiveness was demonstrated at the 2007 DARPA Urban Challenge, LiDAR has become an essential sensor for Level-4 autonomous driving and has been deployed in robotaxi services such as Waymo and Zoox, as well as by autonomous-driving companies such as Nuro and Motional [15, 4, 3, 2, 1].

Although autonomous driving can substantially improve mobility, convenience, and efficiency, it is also highly critical from a security perspective, since even a minor malfunction may lead to fatal, life-threatening accidents. Because an autonomous driving system perceives its surroundings and makes driving decisions based on sensor input, distortion of this input by an attacker can cause

⋆ Corresponding author.

not only object-recognition errors but also erroneous driving decisions. LiDAR emits laser pulses in a predefined scanning order and estimates distance based on the time of flight of the reflected signals. However, since it lacks a separate mechanism to verify the origin of the reflected signals, it remains fundamentally vulnerable to malicious laser injection attacks. Prior work has shown that both point removal and point injection are achievable through external laser interference [6, 10], distorting the perception of autonomous driving systems and potentially leading to erroneous driving decisions or accidents. Under normal driving conditions, temporal consistency between consecutive LiDAR frames remains relatively stable. However, when an attack occurs, point insertion or removal can abruptly disrupt this consistency. Motivated by this observation, this paper leverages the characteristics of the Normal Distributions Transform (NDT), a technique utilized for point cloud registration and localization, for attack detection. By using the inter-frame variation reflected in the NDT results as a new detection indicator, we propose a method that detects attacks using only a single LiDAR sensor and only the preceding frame, at a marginal cost close to that of the registration the vehicle already performs.

2 Background

2.1 Operating Principle of LiDAR

A rotating LiDAR fires laser pulses in a predefined order while rotating 360 degrees, estimating the distance to an object from the time of flight (ToF) of each returned signal. Each measurement is converted into a three-dimensional point from the distance and the scanning direction. When multiple signals are observed from the same direction, the LiDAR determines the final recorded measurement according to its return mode. In the Strongest Return mode, which is the default in many LiDAR sensors, the strongest signal is selected. Consequently, if an attacker injects an external laser pulse with a higher intensity in synchronization with the scanning timing, the legitimate reflected signal can be overwritten. This allows the attacker to manipulate the ToF, thereby altering the perceived distance and distorting the final measurement result.

2.2 Laser-Based LiDAR Attack

As LiDAR-based perception has become central to autonomous driving, a growing body of research has shown that it is vulnerable to adversarial attacks, which can be broadly categorized into object-based and laser-based attacks. Object-based attacks place physical adversarial objects in the environment, ranging from specially fabricated shapes to common everyday items, so that the perception model misrecognizes the surrounding scene [17, 16]. Laser-based attacks, in contrast, use a directed external laser to interfere with the LiDAR's signal-acquisition process itself, injecting or suppressing points at the measurement

level [6, 10, 13, 14]. This paper focuses on laser-based attacks because they constitute a particularly stealthy and general threat. As they leave no visible physical object in the scene, they are difficult for a driver or observer to notice and can be carried out from a distance. Moreover, because they operate directly on the raw sensor measurements, they require no optimization against a specific perception model, which adversarial-object attacks typically depend on.

The Physical Removal Attack (PRA) [6] injects an external laser to prevent a legitimate object from being perceived. Most LiDAR sensors apply a minimum operational threshold (MOT) filter that automatically discards points below a certain range, because at very close range the ToF becomes too short to distinguish a genuine measurement from calibration error. PRA exploits this filter: it shifts the points corresponding to a target object to within the MOT so that they are automatically discarded, thereby preventing the LiDAR from perceiving the object.

PLA [10] precisely synchronizes an external laser with the scanning order of the LiDAR to insert spoofed points at locations chosen by the attacker. Through this, it can create objects that do not actually exist or distort the shape of existing objects, thereby inducing misclassification in the object-recognition model. The HFR attack [13], on the other hand, uses a high-frequency pulsed laser to continuously disrupt many of the LiDAR's laser-firing events. While PLA inserts spoofed points precisely at specific locations, HFR interferes with the reception of the legitimate reflected signals themselves, randomly removing or corrupting the points of an object. As a result, the shape of an existing object partially disappears or is entirely omitted, which can prevent the object-recognition model from detecting that object.

2.3 Countermeasures against Laser-Based Attacks

Several countermeasures have been proposed to defend against LiDAR attacks. Shadow-Catcher [9] focuses on the shadow region behind an object, where few points should exist since the laser cannot reach it, and analyzes the number and distribution of points there to determine whether an attack has occurred. In addition, ADoPT [7] exploits the temporal consistency between consecutive LiDAR frames. It aligns and merges multiple past frames to construct an accumulated frame, and then compares this with the current input frame to detect temporally inconsistent points as outliers. In other words, it determines whether an attack has occurred by leveraging fine-grained, point-level temporal correspondences across frames. Although these techniques have demonstrated the feasibility of LiDAR attack detection by leveraging physical properties or temporal consistency, each has its own limitations. Shadow-Catcher operates on the bounding boxes reported by a 3D object detector and validates each detected object individually, so a perturbation that the detector does not report as an object never enters its analysis. ADoPT, by contrast, decides at the frame level. It aligns multiple historical frames and compares the result with the incoming frame, which makes it burdensome to apply directly in a real-time environment.

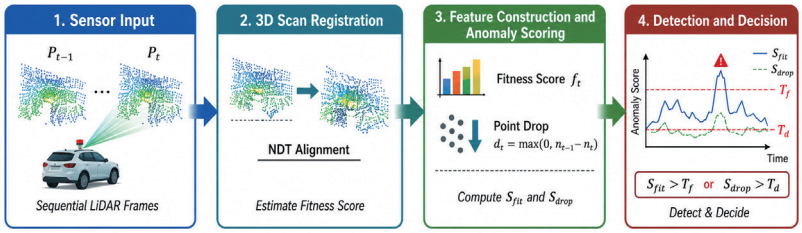


Fig. 1. Overview of the proposed LiDAR attack detection method.

2.4 NDT-based Inter-frame Registration

The Normal Distributions Transform (NDT) [5] is a registration technique used to estimate the relative position and pose between consecutive 3D point cloud frames. In this paper, it is utilized to analyze the structural similarity and variation between frames. NDT first divides the 3D space into a voxel grid and approximates the distribution of points within each voxel as a 3D Gaussian distribution, represented by a mean and a covariance. It then estimates the transformation between the two frames by matching the input point cloud to the probability distributions of these voxels. Unlike the Iterative Closest Point (ICP) algorithm, which requires explicit point-to-point correspondence calculations, NDT performs registration based on the statistical distribution of the point cloud, thereby achieving high computational efficiency. Furthermore, because it relies on the voxel-wise distribution of points, it demonstrates relatively stable and robust performance even in the presence of noise.

3 Threat Model

This paper considers a threat model in which an autonomous vehicle detects road objects using a LiDAR-based perception system, while an adversary aims to distort these perceptual outputs through an external laser-based spoofing attack. The attacker can carry out two types of attacks. An injection attack adds spoofed points that do not exist in the real environment, such as a wall or fog, inducing unnecessary deceleration or sudden braking. A removal attack deletes the points of real objects such as pedestrians or vehicles, so that the victim vehicle fails to perceive a real obstacle, which can lead to a collision. Both attacks abruptly corrupt the point cloud, so the temporal consistency between consecutive frames deviates significantly from that of a normal driving environment. Focusing on this difference, this work aims to detect attacks using only a LiDAR sensor, without any prior information at runtime on whether the input point cloud is benign or contains an attack, and without additional sensors or learning-based models.

4 Methodology

The overall process of the proposed LiDAR attack detection method consists of four steps, as illustrated in Fig. 1. The method analyzes the temporal variation of both the registration result and the point count between consecutive LiDAR point cloud frames to determine whether an attack has occurred. We refer to the resulting fitness and point-drop deviations collectively as inter-frame anomalies. Here, P_t denotes the LiDAR point cloud frame acquired at time t.

The proposed method takes the entire LiDAR frame as input without predefining any specific azimuth or region of interest. Therefore, the same detection procedure applies not only to injection and removal attacks that affect the entire field of view, but also to localized attacks targeting a partial forward region, although its sensitivity depends on the fraction of the frame that the attack occupies. Additionally, it aims for a lightweight structure that uses a single LiDAR sensor without any additional sensors or learning-based models.

Step 1: Consecutive Sensor Data Input

Two consecutive frames P_{t-1} and P_t are used as input. In a normal driving environment, the vehicle's motion and the changes in surrounding objects occur continuously, so the spatial structure and point count between two adjacent frames do not change abruptly. In contrast, a laser-based injection attack adds points or structures that do not exist in the real environment, while a removal attack removes the real points of a specific region. Both cases induce abnormal variations between adjacent frames.

Step 2: 3D Scan Registration

For the input frame pair (P_{t-1}, P_t), NDT-based registration is performed to estimate the optimal transformation between the two frames. Among the metrics the registration yields, this work uses the fitness score f_t, which directly reflects the registration quality, as the primary feature. Since the fitness score quantifies the degree of mismatch between the two registered point clouds, any disruption of the inter-frame spatial structure caused by an attack is directly reflected in the fitness score. In addition, we use the total number of points n_t in frame P_t as a complementary feature, since a removal attack reduces the points available for registration.

Step 3: Feature Construction and Anomaly Scoring

The proposed method estimates the normal variation range of each feature from normal driving data, and scores how far a newly arriving frame deviates from that range in the attack-relevant direction. Since features occasionally spike even during normal driving, the mean and standard deviation are prone to distortion; we therefore adopt a *robust* z-score using the median $\widetilde{x}$ and a scale $\hat{\sigma}$ derived from the median absolute deviation (MAD) [11, 12]:

$$\hat{\sigma} = 1.4826 \times \text{median}(|x - \tilde{x}|) \tag{1}$$

The factor 1.4826 calibrates the MAD to the scale of the standard deviation under a normal distribution, so $\hat{\sigma}$ gives a comparable margin of variation while remaining robust to outliers.

When formulating the final anomaly scores, a small constant ϵ is added to the denominator to prevent division by zero. Furthermore, attacks manifest strictly as abnormal *increases* in the fitness score or per-frame point drop. To filter out the irrelevant negative deviations caused by normal fluctuations, we clip these values to zero using a $\max(0, \cdot)$ function.

Fitness anomaly score (injection attacks). An injection attack adds spoofed points or structures that do not exist in the real environment. This disrupts the inter-frame spatial consistency, which is consequently reflected as a significant rise in the fitness score. Let $\tilde{f}$ and $\hat{\sigma}_f$ be the median and robust scale of the fitness score over normal driving data. The fitness anomaly score is defined as the upper deviation of the current fitness score f_t from its normal level:

$$S_{fit}(t) = \max\left(0,\ \frac{f_t - \tilde{f}}{\hat{\sigma}_f + \epsilon}\right) \tag{2}$$

Point-drop anomaly score (removal attacks). A removal attack eliminates the points used for registration, rendering the registration process itself impossible. In such cases, utilizing registration metrics becomes unfeasible. Therefore, the proposed method detects this attack directly through the massive loss of points. Because the attack causes the point count to drop sharply relative to the previous frame, we first define the per-frame point-count decrease as

$$d_t = \max(0,\ n_{t-1} - n_t) \tag{3}$$

where n_t is the total number of points in frame P_t. Since the point count also fluctuates slightly during normal driving, we score d_t relative to its normal level rather than using it directly. Let $\tilde{d}$ and $\hat{\sigma}_d$ be the median and robust scale of d_t over normal driving data. The point-drop anomaly score is then defined as

$$S_{drop}(t) = \max\left(0,\ \frac{d_t - \tilde{d}}{\hat{\sigma}_d + \epsilon}\right) \tag{4}$$

Step 4: Anomaly Detection and Decision

The fitness anomaly score $S_{fit}(t)$ and the point-drop anomaly score $S_{drop}(t)$ obtained in Step 3 are each compared against their respective thresholds. If either of the two scores exceeds its threshold, the corresponding frame is determined to be under attack:

$$\textit{Attack} \quad \text{if} \quad S_{fit}(t) > T_f \ \text{ or } \ S_{drop}(t) > T_d, \tag{5}$$

Table 1. Summary of NDT performance on the normal driving sequence.

Metric	Value
Number of frames	4,540
Average Runtime (ms)	99.01
Translation MAE / RMSE	0.0134 / 0.0560
Rotation Mean / Std.	0.7347 / 0.9852
Correlation with GT	0.9789
Fitness Mean / Std.	0.0969 / 0.3305

where T_f and T_d are the thresholds for the fitness anomaly score and the point-drop anomaly score, respectively. Injection attacks are detected primarily by the fitness anomaly score, whereas removal attacks are detected primarily by the point-drop score.

Threshold calibration. The robust location and scale of each feature $(\tilde{f}, \hat{\sigma}_f, \tilde{d}, \hat{\sigma}_d)$ are estimated from normal driving data alone, so scoring itself requires no attack data. Given a target false-positive constraint, each threshold is taken as the corresponding upper quantile of the normal score distribution. In the experiments a separate tuning set was used only to confirm that the resulting operating point is reasonable, by reporting the F1-score attained under each constraint.

5 Experiments and Evaluation

5.1 Experimental Setup and NDT Performance on Normal Driving

All experiments were conducted on a laptop with an Intel Core i7-11800H CPU and 32 GB of RAM running Ubuntu 22.04.5 LTS. The experiments were conducted using the KITTI Odometry dataset [8], collected in real-world driving environments. We performed NDT-based registration on consecutive frames of a normal driving sequence, with the results summarized in Table 1. The runtime, fitness, translation, and rotation statistics characterize the registration stability and the normal variation range under attack-free conditions. These statistics are subsequently used as reference values for attack detection.

5.2 Attack Scenario Setup

To evaluate the proposed method, we constructed three attack scenarios that reflect the point injection and point removal that can be induced by external laser interference: Random Noise, Constant-range Wall, and Blackout. The Random Noise attack inserts points at random locations within the target region, independent of the surrounding normal point structure. The Constant-range Wall attack densely places points at a fixed range of approximately 35 m from the sensor, forming a structure at a constant distance that does not exist in the real environment. The Blackout attack removes the points in the target region,

Table 2. Detection results for global (360°) attacks.

Attack	Recall	F1-score	Average Runtime (ms)
Random Noise	100.00%	88.89%	495.7
Constant-range Wall	100.00%	88.89%	95.8
Blackout	100.00%	88.89%	25.67

Table 3. Detection results for localized (front 30°) attacks.

Attack	Recall	F1-score	Average Runtime (ms)
Random Noise	100.00%	88.89%	128.0
Constant-range Wall	89.06%	83.21%	110.8
Blackout	62.50%	66.67%	88.0

reducing the observable structural information. Each attack was applied at two spatial scopes. A global attack affects the entire 360° field of view of the LiDAR. A localized attack is confined to a 30° sector in front of the vehicle. This setup allows us to evaluate whether the same detection procedure applies both when the entire field of view is perturbed and when only a partial forward region is affected.

Frames contained 120,321 points on average. Globally, Random Noise injected a comparable number of synthetic points, the Constant-range Wall 36,096 (30% of the frame), and Blackout removed 117,915 (98%). The front-30° sector held 9,734 points (8.1%), into which 29,201 noise or 19,467 wall points were injected.

The dataset consists of 4,412 normal frames and 128 frames per attack type. The 128 frames of each attack type were split evenly into a tuning set, used to select the decision thresholds, and a held-out test set, on which all reported results were obtained. The normal frames were likewise partitioned into calibration, tuning, and test subsets.

5.3 Detection Performance

The decision thresholds were selected on the tuning set under four candidate false-positive-rate constraints of 0.5%, 1%, 2%, and 5%. As the constraint was relaxed from 0.5% to 2%, the tuning F1-score increased from 89.30% to 91.33% and then to 93.39%. Further relaxing it to 5% yielded only a marginal improvement to 93.47%, indicating that detection performance had largely saturated. We therefore adopt the 2% constraint as the operating point, which yields $T_f = 17.94$ and $T_d = 323.81$. The detection results for global and localized attacks under this setting are reported in Table 2 and Table 3, respectively.

Under global attacks, all three scenarios, Random Noise, Constant-range Wall, and Blackout, achieved a 100% recall. The injection-based attacks, namely Random Noise and Constant-range Wall, add abnormal points across the entire

Table 4. Sensitivity to the FPR constraint. Values are measured on the held-out test set.

Constraint	T_f	T_d	Realized FPR	Precision	Recall	F1
0.5%	72.70	323.81	0.54%	98.11%	81.25%	88.89%
1.0%	32.07	323.81	0.82%	97.38%	86.98%	91.88%
2.0%	17.94	323.81	1.45%	95.66%	91.93%	93.76%
5.0%	13.24	323.81	2.27%	93.49%	93.49%	93.49%

frame. This significantly increases the registration mismatch, allowing them to be reliably detected by the fitness anomaly score. Conversely, the Blackout attack causes massive point loss across the global view. This results in an abrupt decrease in the point count relative to the previous frame, which is successfully captured by the point-drop anomaly score.

For localized attacks confined to the front 30° sector, the detection performance differed across attack types. Random Noise was still detected with 100% recall, as the randomly injected points substantially perturb the normal distribution even within a limited region. However, the recall decreased to 89.06% for Constant-range Wall and to 62.50% for Blackout. For Constant-range Wall, the injected points form a structure aligned at a fixed range, so the perturbation is smaller than for random noise; confined to the front 30° sector, these points account for only a small fraction of those used for registration, which limits the rise in the fitness score and makes the attack harder to separate from normal registration variation.

Table 4 reports the test-set performance under each constraint. The constraint affects the two channels asymmetrically: relaxing it lowers the fitness threshold while the point-drop threshold remains at $T_d = 323.81$ throughout, indicating that recall in this regime is gained mainly through the fitness channel. The localized Constant-range Wall responds directly to this, with its recall rising from 28.12% to 59.38%, 89.06%, and 98.44% as the constraint is relaxed from 0.5% to 5%. Its fitness scores therefore lie close to the decision boundary, indicating that a localized injection attack sits at the margin between attack and normal variation under whole-frame registration. The localized Blackout, which is detected almost entirely through the point-drop channel, therefore remains at 62.50% under every constraint. Under normal driving, registration runs at about 100 ms per frame, and the detection runtime under attack stays at a comparable level in most scenarios, remaining well below 130 ms. The exception is the global Random Noise scenario, whose elevated runtime reflects the increased computational cost when the entire field of view is filled with injected points. However, successfully injecting external laser interference across a full 360° field of view is physically highly challenging, making this an extreme attack scenario.

Table 5. Comparison with ADoPT under frame-level detection.

	ADoPT [7]	Proposed
Frames required	10	2
Injection detection	Yes	Yes
Point-removal detection	Future work	Yes
Reported latency (per frame)	2.1 s	∼130 ms

6 Discussion

6.1 NDT Registration as a Security Signal

Under attack-free driving, consecutive frames are temporally consistent, so the fitness and point-count statistics remain within a narrow band. An injection attack introduces points that do not match the preceding frame and inflates the fitness residual, whereas a removal attack causes an abrupt drop in the point count. The inter-frame variation ordinarily treated as registration noise thus becomes a usable detection feature once read from a security perspective. Since the vehicle already performs this registration for localization, detection adds only two scalar statistics and a threshold comparison per frame, i.e. constant overhead on top of the existing pipeline.

6.2 Comparison under Frame-Level Detection

Defenses differ in the granularity at which they decide. Shadow-Catcher [9] validates individual objects reported by a detector, whereas the proposed method decides per frame; we therefore compare it against ADoPT [7], which likewise consumes raw point clouds and issues a verdict for each incoming frame. Table 5 summarizes the comparison. ADoPT accumulates ten historical frames and reports a latency of about 2.1 s per frame, whereas the proposed method requires only the preceding frame and stays near 130 ms per frame across most scenarios. The sole exception is the global Random Noise case, in which injected points fill the entire 360° field of view; even this worst case is detected within 495.7 ms, and injecting laser interference across a full 360° field of view is in itself physically difficult, making it an extreme rather than a practical attack. The methods also differ in coverage. ADoPT identifies fabricated points as clusters that lack temporal support, and its authors state removal attacks as future work. Laser-based removal erases measurements rather than displacing them, leaving no cluster to flag; the point-drop channel of the proposed method captures this case within the same procedure, as the Blackout results confirm.

6.3 Detection Scope and Extensions

Because the detection features are aggregated over the whole frame, attacks confined to a narrow sector are diluted among the remaining points, though

still detected at a meaningful rate. Rather than registering the whole frame, restricting the NDT to a region of interest (RoI) or applying it per object would prevent localized perturbations from being averaged out in both the fitness and the point-count statistics, allowing such attacks to be detected more effectively. Furthermore, because registration would then be performed over a limited set of points, the computational cost is expected to drop substantially. In deployment, the point-drop channel may raise false alarms under conditions that thin the returns, such as adverse weather, and the calibration statistics require re-estimation when the sensor or environment changes.

7 Conclusion

In this paper, we proposed a LiDAR spoofing detection method that repurposes NDT registration, conventionally used for scan matching and localization, from a security perspective. By treating the inter-frame variation induced by an attack as a detection feature, the method identifies injection attacks through a fitness anomaly score and removal attacks through a point-drop anomaly score, without additional sensors or learning-based models. On the KITTI Odometry dataset, all global attacks were detected with 100% recall, while the recall for localized attacks varied with the attack type; the experiments showed that NDT registration can detect the loss of temporal consistency in the point cloud caused by an attack, and the detection runtime remained low. At the same time, the evaluation delineated a limitation: detection sensitivity decreases for attacks confined to a small forward sector, such as the localized Constant-range Wall and Blackout, because the manipulated points occupy only a small fraction of those used for whole-frame registration. As future work, rather than registering the whole frame, we plan to restrict NDT to a region of interest or estimate it at the object level, which we expect to improve robustness against spatially localized attacks while further reducing the computational cost.

ACKNOWLEDGEMENT

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00341722, Development of Cyber Resilience Method for Intelligent Service Robots, 34%), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00345275, Enhancing Cyber Resilience of AI-enabled Cyber-Physical Systems against Physical Layer Security Threats, 33%), and Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIR) (RS-2023-KI002685, HRD Program for Industrial Innovation, 33%).

References

1. Motional. https://motional.com/

2. Nuro. https://www.nuro.ai/
3. Zoox. https://zoox.com/
4. Waymo has launched its commercial self-driving service in phoenix. https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12 (2018)
5. Biber, P., Straßer, W.: The normal distributions transform: A new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). vol. 3, pp. 2743–2748. IEEE (2003)
6. Cao, Y., Bhupathiraju, S.H., Naghavi, P., Sugawara, T., Mao, Z.M., Rampazzi, S.: You can't see me: Physical removal attacks on {lidar-based} autonomous vehicles driving frameworks. In: 32nd USENIX security symposium (USENIX Security 23). pp. 2993–3010 (2023)
7. Cho, M., Cao, Y., Zhou, Z., Mao, Z.M.: Adopt: Lidar spoofing attack detection based on point-level temporal consistency. arXiv preprint arXiv:2310.14504 (2023)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
9. Hau, Z., Demetriou, S., Muñoz-González, L., Lupu, E.C.: Shadow-catcher: Looking into shadows to detect ghost objects in autonomous vehicle 3d sensing. In: European Symposium on Research in Computer Security. pp. 691–711. Springer (2021)
10. Jin, Z., Ji, X., Cheng, Y., Yang, B., Yan, C., Xu, W.: Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 1822–1839. IEEE (2023)
11. Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L.: Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of experimental social psychology **49**(4), 764–766 (2013)
12. Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. Journal of the American Statistical association **88**(424), 1273–1283 (1993)
13. Sato, T., Hayakawa, Y., Suzuki, R., Shiiki, Y., Yoshioka, K., Chen, Q.A.: Lidar spoofing meets the new-gen: Capability improvements, broken assumptions, and new attack strategies. arXiv preprint arXiv:2303.10555 (2023)
14. Sato, T., Suzuki, R., Hayakawa, Y., Ikeda, K., Sako, O., Nagata, R., Yoshida, R., Chen, Q.A., Yoshioka, K.: On the realism of lidar spoofing attacks against autonomous driving vehicle at high speed and long distance. In: NDSS (2025)
15. Urmson, C., Bagnell, J.A., Baker, C., Hebert, M., Kelly, A., Rajkumar, R., Rybski, P.E., Scherer, S., Simmons, R., Singh, S., et al.: Tartan racing: A multi-modal approach to the darpa urban challenge (2007)
16. Zhu, S., Zhao, Y., Chen, K., Wang, B., Ma, H., Wei, C.: AE-Morpher: Improve physical robustness of adversarial objects against LiDAR-based detectors via object reconstruction. In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 7339–7356. USENIX Association, Philadelphia, PA (Aug 2024), https://www.usenix.org/conference/usenixsecurity24/presentation/zhu-shenchen
17. Zhu, Y., Miao, C., Zheng, T., Hajiaghajani, F., Su, L., Qiao, C.: Can we use arbitrary objects to attack lidar perception in autonomous driving? In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. p. 1945–1960. CCS '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3460120.3485377, https://doi.org/10.1145/3460120.3485377

Understanding the U.S. RG 5.96 Three-Tier Cybersecurity Analysis: Procedural Review and Considerations for Advanced Reactor Design

Heewon Choi[1], Soo Min Lim[1], Su Bong Lee[1] and Kook Heui Kwon[1*]

[1] Korea Institute of Nuclear Nonproliferation and Control (KINAC), Daejeon, Republic of Korea

Abstract. As advanced reactor (AR) technologies including Small Modular Reactors (SMRs) and non-light water reactors (non-LWRs) continue to emerge, cybersecurity frameworks developed for conventional large light water reactors (LWRs) face increasing limitations in addressing their distinct design characteristics and operational environments. In March 2026, the U.S. Nuclear Regulatory Commission (NRC) finalized 10 CFR Part 53 and issued Regulatory Guide (RG) 5.96, establishing a risk-informed, performance-based, and technology-inclusive cybersecurity framework centered on the Three-Tier Analysis methodology. This paper systematically analyzes the Three-Tier Analysis procedures defined in RG 5.96 on a tier-by-tier basis and derives three key cybersecurity considerations for advanced reactor design through comparative analysis and reference to Sandia National Laboratories' (SNL) Tiered Cybersecurity Analysis (TCA) application cases: (1) integration of physical security assessments into cybersecurity analysis; (2) adversary-oriented attack path modeling linked to Defensive Cybersecurity Architecture (DCSA) design; and (3) a consequence-based graded approach to asset categorization. These findings suggest that the Three-Tier Analysis framework provides a structured methodology applicable to the diverse design characteristics of advanced reactors.

Keywords: Nuclear Regulatory Cybersecurity Framework, Advanced Reactor, Consequence-based, Risk-informed, Performance-based, Three-Tier Analysis

1 Introduction

The cybersecurity paradigm is undergoing a fundamental shift. Conventional security frameworks were structured around prescriptive compliance, where fulfilling explicitly mandated obligations was considered sufficient. This shift is particularly pronounced in the financial sector, where regulations now establish only minimum principles and objectives, leaving organizations to design and operate their own controls based on internal risk assessments[1].

The nuclear industry faces a similar inflection point. Advanced reactors, including SMRs, introduce new technical characteristics such as remote operation, autonomous control, and wireless communication that existing regulations — designed around the air-gapped environments of large LWRs — are increasingly insufficient to address[2]. These designs also differ

widely in their safety architectures, consequence potentials, and attack surfaces, which a uniform control catalog derived from a single large-LWR reference plant cannot accommodate — motivating a risk-informed, consequence-based approach scaled to each design. In response, the NRC finalized 10 CFR Part 53 and §73.110, and issued Regulatory Guide (RG) 5.96 as implementation guidance. Part 53 adopts risk-informed, consequence-based, and performance-based principles, representing a shift from uniform prescriptive control application toward protection levels determined by the severity of potential cyberattack consequences.

As development of advanced reactors, including the innovative i-SMR, is actively pursued in Korea, a thorough understanding of the U.S. regulatory framework and its core analytical methodology is increasingly necessary to proactively identify key cybersecurity considerations for future advanced reactor licensing. This paper systematically analyzes the Three-Tier Analysis procedures defined in RG 5.96 on a tier-by-tier basis and derives key cybersecurity considerations for advanced reactor design through structural comparison with the existing U.S. regulatory framework. The contributions of this paper are as follows.

- We systematically analyzed the Three-Tier Analysis procedures defined in RG 5.96 on a tier-by-tier basis.
- We identified key structural characteristics of RG 5.96 framework by comparing it with existing cybersecurity regulatory analysis approaches, referencing Sandia National Laboratories' Tiered Cybersecurity Analysis (TCA) application cases.
- We derived cybersecurity considerations for advanced reactor design based on the procedural analysis of RG 5.96.

The remainder of this paper is organized as follows. Section 2 examines the evolving U.S. nuclear cybersecurity regulatory landscape. Section 3 analyzes the Three-Tier Analysis procedures of RG 5.96 on a tier-by-tier basis. Section 4 compares the framework against existing regulatory approaches and presents key cybersecurity considerations for advanced reactor design. Section 5 summarizes the findings and outlines future research directions.

2 Background

2.1 U.S. Nuclear Cybersecurity Regulations: From Prescriptive to Consequence

U.S. nuclear cybersecurity requirements comprise two frameworks: the existing framework, applicable to reactors licensed under 10 CFR Parts 50/52 (the conventional construction-permit and operating-license pathway), which consists of §73.54 and RG 5.71; and the new framework, applicable to reactors licensed under 10 CFR Part 53 (the technology-inclusive, risk-informed pathway established for advanced reactors), which consists of §73.110 and RG 5.96.

§73.54 aims to protect identified Critical Digital Assets (CDAs) against cyber threats up to the Design Basis Threat (DBT) level and requires high assurance of protection. RG 5.71 provides a catalog of technical, operational, and managerial security controls for CDA protection. Although it permits alternative controls — and supplementary guidance such as NEI 13-10 introduces consequence-based grading of CDAs — its structure remains control-catalog-centered, applying a common baseline of controls to identified CDAs. This distinguishes it from RG 5.96, which derives the required protection directly from a consequence-based analysis at each tier.

§73.110 aims to ensure that cyberattacks do not result in radiological sabotage or physical intrusion, requiring reasonable assurance of protection. To implement this, RG 5.96 introduces

the Three-Tier Analysis as its core methodology. This framework is designed to realize the Security by Design (SeBD) principle, which structurally eliminates or reduces the potential for cyberattacks from the design stage. It sequentially analyzes risks at the facility, function, and system levels, evaluating the applicability of SeBD elements at each stage. This represents a shift away from reactive security control application toward an approach in which design characteristics themselves serve as the foundation of cybersecurity.

2.2 Finalization of 10 CFR Part 53: Key Changes

The NRC finalized the 10 CFR Part 53 Final Rule in March 2026, with an effective date of April 2026[3]. Two notable changes were introduced in the cybersecurity regulatory structure relative to the Proposed Rule.

First, the cybersecurity regulatory guide was finalized. The draft DG-5075 was formally issued as RG 5.96, in which the Three-Tier Analysis, previously presented in draft form, was officially established as the core methodology for meeting §73.110 requirements. A notable procedural change was also identified in the Adversary Technical Sequences (ATS) process: the control application criterion for Most Critical Systems was revised from "All System Cybersecurity Controls" in DG-5075 to "Necessary System Cybersecurity Controls" in RG 5.96, marking a shift away from applying the full control catalog to Most Critical Systems by default toward selecting only the controls necessary to mitigate each system's identified attack sequences [4].

Second, RG 5.96 also introduced requirements not present in DG-5075. As some SMRs adopt a factory-fabricated, fuel-loaded delivery model, cybersecurity during manufacturing and transportation emerged as a new regulatory challenge. Accordingly, RG 5.96 established cybersecurity requirements for such scenarios. Specifically, if no digital assets are involved, only simplified administrative requirements apply; if digital assets are present, Three-Tier Analysis is provided as an acceptable implementation method, with the analysis criteria modified to focus on the protection of functions performing radiation monitoring and criticality requirements, and physical security functions, respectively. The Three-Tier Analysis procedures finalized in RG 5.96 are analyzed in detail in Section 3.

2.3 ROK Context for Advanced Reactor Cybersecurity Review

The domestic nuclear facility licensing framework in Korea is structured around the mandatory construction-permit and operating-license steps, undertaken for a specific site and facility, together with the optional, site-independent standard design approval for power reactors and associated facilities under the Nuclear Safety Act. The direct regulatory basis for cybersecurity is found in the Act on Physical Protection and Radiological Emergency Preparedness for Nuclear Facilities and its subordinate regulations. This Act defines nuclear facility computer and information systems as electronic control, management systems, and information and communication networks of nuclear facilities, and defines electronic intrusion as any attack on such systems intended to cause illegal transfer of nuclear material or nuclear sabotage.

The current physical protection plan framework requires the inclusion of computer and information system security plans, system analysis, tiered security strategies, technical, operational, and managerial security controls, and security audits and vulnerability assessments[5]. These elements represent potential institutional touchpoints through which the Three-Tier Analysis of RG 5.96 could be incorporated into the domestic regulatory review process. However, as the current requirements are primarily structured around security planning, asset analysis, and control documentation, supplementary provisions at the level of subordinate regulations, detailed drafting criteria, or review guidelines would be necessary to explicitly incorporate consequence-based and tiered analytical methodologies that reflect the design characteristics of advanced reactors.

3 RG 5.96 Three-Tier Analysis: A Tier-by-Tier Procedural Review

RG 5.96 presents the Three-Tier Analysis as the core methodology for meeting the requirements of §73.110. This section first examines the theoretical basis and overall structure of the Three-Tier Analysis, followed by a detailed tier-by-tier analysis of its procedural implementation.

3.1 Overview: Risk Management Framework & Three-Tier Structure

RG 5.96 adopts a tiered risk management framework analogous to that of NIST SP 800-37. NIST SP 800-37 structures risk management across three levels — Organization (Tier 1), Mission/Business Process (Tier 2), and Information System (Tier 3) — progressing from strategic risk perspectives at higher tiers to tactical risk perspectives at lower tiers[6]. RG 5.96 applies this structure to the nuclear facility context, implementing a three-tier framework at the facility, function, and system levels. By conducting independent risk assessments at each tier, the framework realizes defense-in-depth and prevents failures at a single tier from compromising the overall cybersecurity program.

The core principle of this structure is to sequentially perform analyses at each tier until it is demonstrated that cyberattacks cannot result in radiological sabotage or physical intrusion. Accordingly, the Three-Tier Analysis may conclude at Tier 1, extend through Tier 2, or require all three tiers. At Tier 1, existing safety and security assessments are leveraged to determine whether the facility's design-basis features are sufficient to prevent the consequences of radiological sabotage and physical intrusion. If sufficiency is demonstrated, no further analysis is required; otherwise, the analysis proceeds to Tier 2.

At Tier 2, functional-level attack pathways are eliminated through the application of prohibitive Cyber Security Plan (CSP) measures (e.g., prohibition of wireless communications) and passive or deterministic DCSA elements. Only scenarios that remain unmitigated advance to Tier 3, where ATSs are developed for residual scenarios and active CSP and DCSA measures (e.g., detection and response systems) are specified to establish system-level protections. Both Tier 2 and Tier 3 apply a graded approach, in which the level of cybersecurity protection is commensurate with the potential consequences of a cyberattack.

3.2 Tier 1: Facility-Level Analysis

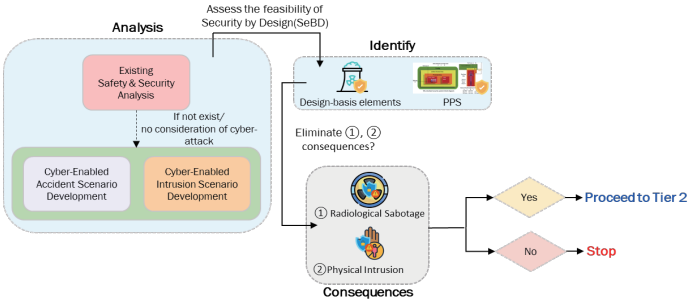


Fig. 1. Tier 1(Facility-Level) analysis flow

Figure 1 illustrates the overall procedure of the Tier 1 facility-level analysis. Tier 1 determines whether existing design-basis features and the Physical Protection System (PPS) are already sufficient to prevent radiological sabotage and physical intrusion. This tier leverages existing safety and security assessments as its starting point; Cyber-Enabled Accident Scenarios (CEAS) or Cyber-Enabled Intrusion Scenarios (CEIS) are newly developed only when such assessments are absent or do not account for cyberattacks. A key assumption at Tier 1 is that a cyberattack succeeds immediately upon adversary access. This assumption is intended not to assess whether an attack can be prevented, but rather to evaluate whether the design-basis features and PPS can prevent the attack from resulting in radiological sabotage or physical intrusion even if the attack succeeds. If the analysis demonstrates that such consequences cannot occur, no further tier analysis is required.

Tier 1 conducts CEAS and CEIS analyses corresponding to the consequences of radiological sabotage and physical intrusion, respectively. CEAS evaluates whether a cyberattack could lead to radiological sabotage by identifying accident scenarios in which cyber compromise of reactor safety functions results in radioactive release. During this analysis, the applicability of SeBD elements — such as passive safety systems and analog control devices — that structurally prevent cyberattack consequences is assessed. If the design-basis is demonstrated to be sufficient to prevent radiological sabotage, the analysis for that scenario is concluded and a periodic review mechanism is established. Otherwise, the scenario advances to Tier 2.

CEIS identifies intrusion scenarios in which cyberattacks compromise digital assets within the PPS, potentially leading to physical intrusion or unauthorized removal of nuclear material. While CEIS follows the same procedural structure as CEAS, it utilizes existing physical security analysis rather than safety analysis. This is intended to evaluate whether a cyberattack could compromise the PPS design and enable physical intrusion. If the PPS is demonstrated to be sufficient to prevent physical intrusion, the analysis for that scenario concludes; otherwise, it advances to Tier 2.

Once both CEAS and CEIS analyses are complete, if both consequences are demonstrated to be preventable by the existing design-basis features and PPS, the Tier 1 analysis concludes and the results are documented in the CSP. If either consequence cannot be sufficiently prevented, the unmitigated scenarios advance to the Tier 2 function-level analysis.

3.3 Tier 2: Function-Level Analysis

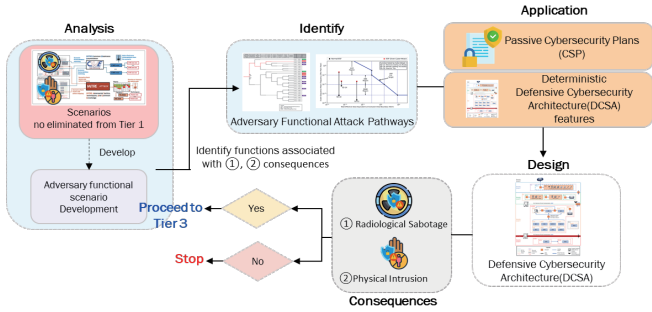


Fig. 2. Tier 2(Function-Level) analysis flow

Figure 2 illustrates the overall procedure of the Tier 2 function-level analysis. As the second tier of the Three-Tier Analysis, Tier 2 analyzes functional-level attack pathways for CEAS and

CEIS scenarios identified in Tier 1 as insufficiently mitigated by existing design-basis features and PPS alone. The objective of Tier 2 is to identify attack pathways through which an adversary could access Safety, Security, and Emergency Preparedness (SSEP) functions and cause radiological sabotage or physical intrusion, and to derive the CSP implementation measures and passive or deterministic DCSA elements necessary to eliminate those pathways.

The core analytical tool of Tier 2 is the Adversary Functional Scenario Analysis (AFSA). AFSA aims to identify cybersecurity risks associated with the consequences of radiological sabotage and physical intrusion, and is performed for scenarios identified in Tier 1 CEAS and CEIS analyses as potentially leading to such consequences. Scenarios fully mitigated by SeBD elements are excluded from AFSA. AFSA begins by developing adversary functional scenarios to identify attack pathways through which an adversary could access SSEP functions. For each CEAS and CEIS, four attack pathways are considered: physical access, wired communications, wireless communications, and portable media/device connectivity. The identified pathways are then addressed through two approaches: elimination and mitigation. Elimination is achieved by applying DCSA elements that structurally block external network access through deterministic and unidirectional communication paths such as data diodes, or through prohibitive CSP measures such as wireless communication bans. Where no unmitigated scenarios remain, Tier 3 is optional rather than mandatory, and licensees may elect to perform it to further enhance defense-in-depth.

If attack pathways remain unmitigated despite the application of CSP implementation measures and passive or deterministic DCSA elements, the corresponding scenarios advance to the Tier 3 system-level analysis.

3.4 Tier 3: System-Level Analysis

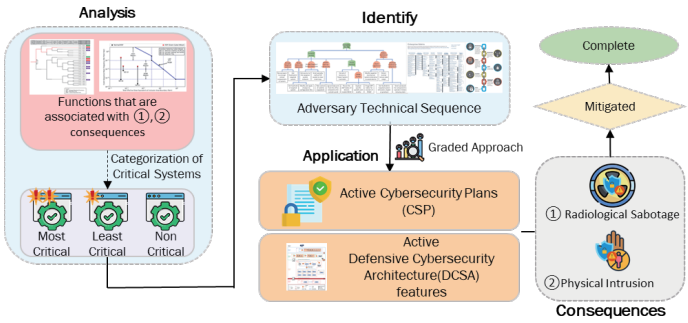


Fig. 3. Tier 3(System-Level) analysis flow

Figure 3 illustrates the overall procedure of the Tier 3 system-level analysis. As the third tier of the Three-Tier Analysis, Tier 3 performs detailed system-level analysis for scenarios identified in Tier 2 AFSA as having the potential for an adversary to complete a CEIS or CEAS. Tier 3 consists of three steps: identification of Critical Functions, system categorization, and ATS analysis. Updates are required upon discovery of new vulnerabilities, advances in attack techniques, or identification of new accident or intrusion scenarios.

Critical Function identification is the first step of Tier 3, involving the identification of digital assets associated with the consequences of radiological sabotage and physical intrusion and the derivation of Critical Functions. Specifically, all functions performed or supported by a system

are identified and compared against the Critical Functions associated with CEAS and CEIS, to verify that the system's design and functions are consistent with the assumptions made in the Tier 1 and Tier 2 analyses. For example, if a system was assumed to be fully isolated from external networks in Tier 1 and Tier 2, but an external connection for maintenance purposes is identified in Tier 3, the prior assumptions are invalidated. In such cases, the licensee may take one of three actions: re-perform the facility-level analysis (Tier 1) for the Critical Functions of that system, update the assumptions of the existing analysis, or incorporate alternative compensating measures into the CSP and DCSA.

Categorization of Critical Systems is the step for applying a graded approach, determining whether each system performs or supports Critical Functions, or adversely affects them if compromised. Systems are classified into three categories: Noncritical, Least Critical, and Most Critical. Noncritical Systems do not require security controls, though their inclusion is recommended to enhance DCSA defense-in-depth. Least Critical Systems are afforded flexibility in control selection, while Most Critical Systems require a review of the entire control catalog to identify necessary controls — a process of evaluating each control's applicability rather than applying all controls uniformly. Each Critical System must be located within a computer security zone with defined physical and logical boundaries, and the level of security protection is graded according to system categorization.

ATS analysis examines the step-by-step technical attack chain through which an adversary would complete a CEIS or CEAS. For scenarios that remain unmitigated despite the application of passive or deterministic DCSA and prohibitive CSP measures in Tier 2, Tier 3 applies active CSP and DCSA elements with prevention, detection, response, and recovery capabilities to prevent the adversary from completing the attack chain. ATS analysis assumes that the passive or deterministic DCSA and prohibitive CSP measures derived in Tier 2 are already in place, and applies active CSP and DCSA to address residual risks through which an adversary could still complete the attack chain. The outputs of ATS analysis — including identification of additional controls and assessment of detection and response sufficiency — are documented in the CSP. Once all ATSs have been mitigated, the analysis is complete.

4 Cybersecurity Considerations for Advanced Reactor Design Based on RG 5.96 Three-Tier Analysis

This section compares the Three-Tier Analysis procedures of RG 5.96, analyzed in Section 3, against the conventional control-catalog-based regulatory approach (RG 5.71), and presents three key characteristics of the Three-Tier Analysis and their implications for advanced reactor cybersecurity design. These three characteristics were derived by contrasting each tier of RG 5.96 with the CDA-and-control-catalog approach of RG 5.71, cross-checked where applicable against SNL's TCA application case. Unlike the RG 5.71-based approach, which applies a uniform set of security controls to all identified CDAs, the Three-Tier Analysis progressively narrows the scope of analysis and control application as consequence prevention is demonstrated at each tier. The sequential operation of SeBD verification at Tier 1, attack pathway elimination at Tier 2, and consequence-based graded categorization at Tier 3 results in a progressively reduced analytical burden — an effect that becomes more pronounced as the physical design characteristics of advanced reactors are strengthened.

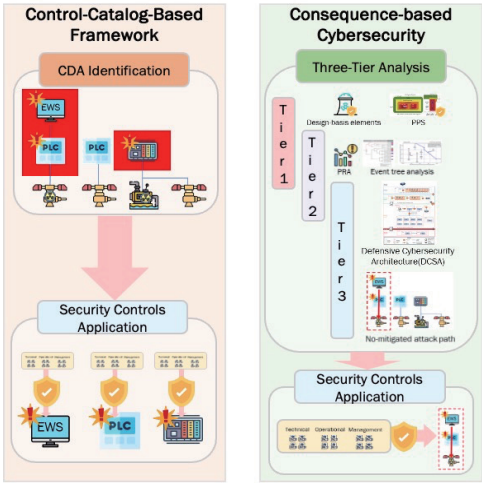


Fig. 4. Comparison of Cybersecurity Control Application: Control-Catalog-Based Framework (RG 5.71) vs. Consequence-Based Framework (RG 5.96)

Figure 4 illustrates the difference between these two approaches. As shown, while RG 5.71 applies a uniform control set to all identified CDAs, the Three-Tier Analysis of RG 5.96 continues analysis only for unmitigated scenarios at each tier, ultimately applying controls that are graded according to consequence severity. Sections 4.1 through 4.3 examine each of these characteristics in detail.

4.1 Integration of Physical Security and Cybersecurity Analysis

Conventional cybersecurity regulation has focused primarily on applying security controls to CDAs identified at the system level. In contrast, the Three-Tier Analysis of RG 5.96 first evaluates, at the facility and function levels, whether physical design characteristics and the physical protection system can already prevent the consequences of cyberattacks before proceeding to system-level analysis. This process examines the applicability of SeBD elements — such as passive safety systems and data diodes — that structurally prevent cyberattack consequences by design. Where SeBD is demonstrated to prevent such consequences, further analysis at lower tiers becomes unnecessary. In this way, the Three-Tier Analysis integrates physical design characteristics with cybersecurity analysis, realizing an approach in which cybersecurity is structurally embedded from the design stage.

This approach is particularly well-suited to advanced reactors with enhanced physical design characteristics, such as passive safety systems. For advanced reactors including SMRs, structurally embedding cybersecurity from the design stage is essential, and the integration of physical design characteristics with cybersecurity analysis provides the structural basis to achieve this. Reactors with stronger passive safety features are more likely to demonstrate consequence prevention at the facility level, thereby reducing the need for lower-tier analyses and improving licensing efficiency.

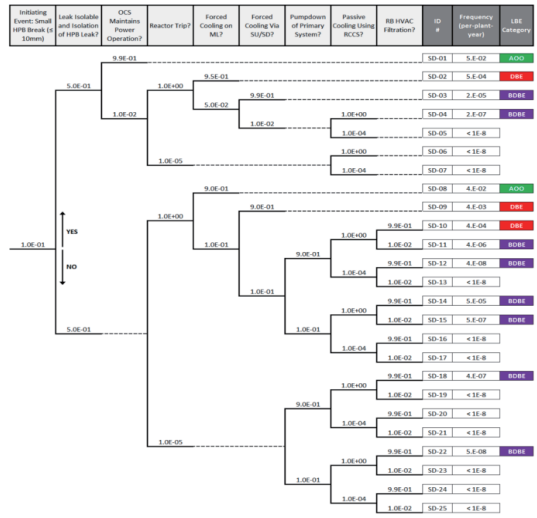


Fig. 5. High Temperature Gas-Cooled Reactors(HTGR) Event Tree Analysis example [4]

Sandia National Laboratories (SNL) applied the TCA framework to an HTGR design, utilizing event tree analysis from probabilistic risk assessment (PRA), as illustrated in Figure 5[7]. PRA event sequences were used to model accident scenarios in which cyber-induced functional failures lead to radioactive release. Specifically, combinations of simultaneous or sequential compromise of multiple functions — such as cooling and emergency response — were analyzed to quantitatively assess whether each combination results in an increased risk of radioactive release. This enabled a distinction between scenarios already mitigated by physical design characteristics and those requiring additional security controls. Systems identified as requiring controls were classified into four security levels based on risk, with graded passive DCSA applied accordingly. This case demonstrates that the integration of physical design characteristics and cybersecurity analysis is a practically implementable approach for establishing cybersecurity programs for advanced reactors.

4.2 Adversary-Oriented Attack Path Modeling and DCSA Design

Unlike conventional cybersecurity regulation, which focuses on the protection of individual CDAs, the Three-Tier Analysis of RG 5.96 adopts an adversary-oriented approach that systematically analyzes attack pathways at the functional and system levels. At Tier 2, AFSA identifies functional-level attack pathways through which an adversary could access SSEP functions, considering four attack vectors: physical access, wired communications, wireless communications, and portable media/device connectivity. Identified pathways are structurally blocked through passive DCSA elements and prohibitive CSP measures. Scenarios that remain unmitigated advance to the Tier 3 ATS analysis, where the step-by-step technical attack chain through which an adversary would complete an attack is analyzed and addressed through active CSP and DCSA measures. Through this progressive structure — from functional-level pathway elimination to

system-level attack chain mitigation — RG 5.96 enables a systematic, adversary-oriented threat analysis that moves beyond the protection of individual assets.

This approach is particularly well-suited to advanced reactors characterized by new technical features such as remote operation, autonomous control, and wireless communication. Unlike conventional large LWRs, advanced reactors may present a broader range of attack pathways, and individual asset protection alone is insufficient to adequately block complex functional-level attack vectors.

In SNL's TCA application study, functional-level adversary access pathways were identified based on risk assessments conducted through PRA, and the results were directly linked to DCSA design. Specifically, systems were assigned Security Levels 1 through 4 according to the degree of risk increase, with the highest-level systems permitted only unidirectional communication via data diodes, structurally blocking communication with lower-level systems. This demonstrates a framework in which adversary-oriented functional-level threat analysis directly informs DCSA design, confirming the practical applicability of the Three-Tier Analysis for establishing cybersecurity programs for advanced reactors.

4.3 Consequence-Based Graded Approach to Asset Categorization

The Three-Tier Analysis of RG 5.96 implements a graded approach in which systems are classified based on the potential severity of cyberattack consequences, and security controls are applied proportionally. A key characteristic of this approach is that consequence-based classification extends beyond asset identification to directly determine the selection and application of security controls. As consequence severity increases, the nature of required controls shifts from policy- and procedure-based administrative measures toward structural elimination of attack surfaces — a distinction that sets this approach apart from simple layering of security stringency.

This graded approach is particularly well-suited to the diverse design landscape of advanced reactors. Advanced reactor types, including SMRs and non-LWRs, vary significantly in safety system configuration, automation level, and operational mode. Rather than applying uniform security requirements across all designs, a consequence-proportionate graded approach provides a foundation for achieving consistent security objectives while accommodating the distinct design characteristics of each reactor type. This graded categorization, together with the physical-security integration and adversary-oriented pathway analysis, provides the structural basis for scaling cybersecurity to the specific consequence profile of each advanced reactor design.

4.4 Applicability to ROK Advanced Reactor Cybersecurity Review

The Three-Tier Analysis of RG 5.96 can serve as a reference model to complement the existing CDA identification and security control approach in the domestic cybersecurity review of advanced reactors. Given the unique design and operational characteristics of advanced reactors, including passive safety systems, modular construction, remote monitoring, and diverse communication pathways, a structured approach that progressively evaluates cyberattack consequences at the facility, function, and system levels is warranted. Table 1 summarizes the potential domestic application of each tier of the Three-Tier Analysis.

Table 1. Tier-by-Tier Implementation Approach for Three-Tier Analysis

Tier	Applicability to Korean Regulatory Review	Current Regulatory Basis	Supplementary Consideration

Tier 1	Assessable via design-basis safety features & PPS	Nuclear Safety Act & Radiation Disaster Prevention Act (Sabotage & Electronic intrusion)	Integration guidance needed across safety & physical protection & cybersecurity
Tier 2	Assessible via identification & elimination of attack pathways targeting SSEP	Computer/information system analysis, tiered security strategy, and security control documentation	Review & inspection criteria needed for each attack pathway
Tier 3	Assessible via critical system categorization & graded controls for residual risks	Technical, operational, and management security controls with monitoring & evaluation	Criteria needed for consequence-based asset grading & DCSA application

From a domestic regulatory review perspective, Tier 1 can be interpreted as a step that verifies whether existing design-basis safety features and physical protection systems can prevent the consequences of radiological sabotage, illegal transfer of nuclear material, or physical intrusion, even in the event of a successful cyberattack. This suggests that advanced reactor cybersecurity reviews should extend beyond digital asset identification and be linked with design-stage safety analysis and physical protection assessments.

Tier 2 can serve as a basis for evaluating whether attack pathways capable of accessing Safety, Security, and Emergency Preparedness functions have been systematically identified and eliminated. For advanced reactors with potential use of remote monitoring, autonomous control, or wireless communication, functional-level attack pathway analysis may be more important than conventional CDA-list-based reviews. Tier 3 provides a structural basis for applying security controls in a graded manner according to the criticality and potential consequence severity of each system. Rather than applying uniform security controls to all digital systems, systems can be classified according to their association with severe consequences such as radiological sabotage, physical intrusion, or illegal transfer of nuclear material, with detection, response, and recovery measures focused on high-risk residual scenarios.

In terms of domestic implementation, it is more practical to reflect the Three-Tier Analysis of RG 5.96 through targeted supplementation of the existing physical protection plan drafting framework, detailed drafting criteria, and review guidelines rather than pursuing comprehensive legislative revision. Specifically, this could involve incorporating facility-, function-, and system-level analysis concepts into the "computer and information system analysis" element, attack pathway elimination and DCSA design review into the "tiered security strategy" element, and consequence-based asset classification and residual-risk security measures into the relevant security control and monitoring elements.

5 Conclusion

This study confirmed that the Three-Tier Analysis of RG 5.96 offers an effective approach to establishing cybersecurity programs for advanced reactors in three key respects: (1) integrating physical design characteristics and physical protection assessments as the starting point for cybersecurity analysis; (2) conducting adversary-oriented threat analysis at the functional and

system levels and linking the results to DCSA design; and (3) applying consequence-based graded controls that accommodate the diverse design characteristics of advanced reactors.

The transition of U.S. nuclear cybersecurity regulation toward a consequence-based, performance-based framework through RG 5.96 reflects a broader convergence of the nuclear sector toward risk-informed security paradigms. The analytical findings of this paper are expected to serve as a foundation for examining key cybersecurity considerations in the development of advanced reactors of various types. While the domestic regulatory framework already provides relevant statutory bases, further operationalization into a consequence-based and tiered cybersecurity review structure will be necessary to address the distinct design and operational characteristics of advanced reactors. Future research will apply the full Three-Tier Analysis — CEAS/CEIS, AFSA, system categorization, and ATS analysis — to a specific advanced reactor design, such as an SMR or non-LWR, as a worked example to validate the considerations derived here and to identify the practical gaps that emerge in implementation.

Acknowledgments. This work was supported by the Nuclear Safety Research Program through the Regulatory Research Management Agency for SMRs (RMAS) and the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (No. RS-2022-KN073810, 50%) and (No. RS-2026-25540913, 50%).

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. Sejin Jung, Lawtimes, “Financial Security Paradigm: From Regulation to Autonomy”, “https://www.lawtimes.co.kr/news/articleView.html?idxno=210813”, Accessed in June, 2026.
2. GARCIA, Nuclear Regulatory Commission (NRC), “U.S.A Regulatory efforts for cyber security of advanced reactors”, 2022.
3. Nuclear Regulatory Commission (NRC), “10 CFR Part 53”, 2026.
4. Nuclear Regulatory Commission (NRC), “RG 5.96”, 2026.
5. Nuclear Safety and Security Commission (NSSC), “Act on Protection of Nuclear Facilities, etc. and Radiation Disaster Prevention Measures”, 2026.
6. NIST, “NIST SP 800-37 rev.2”, 2018.
7. Sandia National Laboratories, “Advanced Reactor Safeguards & Security Design of Defensive Cybersecurity Architectures for High Temperature, Gas-Cooled Reactors”, 2024.
8. Korea Institute of Nuclear nonproliferation and Control, “KINAC/RS-015”, 2023.

LLM-Assisted Command Syntax Extraction for Flight Software Fuzzing

Geonwoo Park[1], Junhak Lee[2], and Sang Kil Cha[1]

[1] KAIST, Daejeon, Republic of Korea
{rjsdn, sangkilc}@kaist.ac.kr
[2] Raon Secure, Seoul, Republic of Korea
jhlee0010@raon.com

Abstract. Flight software (FSW) manages the mission-critical tasks of spacecraft, from control and communication to data processing. Finding its vulnerabilities is critical, since even a denial-of-service flaw can interrupt a mission, sever communication, or push a spacecraft off its intended orbit. Fuzzing is a natural fit for this task, but FSW accepts only inputs that obey a strict command syntax, and traditional coverage-based fuzzers rarely produce them. Deriving that syntax, in turn, has required manual analysis of command structures, which is inefficient and hard to scale to complex systems. We address this challenge with SatFuzz, a fuzzing framework that automatically extracts FSW command syntax by providing an LLM with call graphs, source code, and manuals. Our evaluation on NASA cFS, SUCHAI, and C2A shows that SatFuzz achieves higher coverage and discovers more vulnerabilities than traditional coverage-based fuzzing.

1 Introduction

Once a spacecraft leaves the ground, software is all that keeps the mission alive. Flight software (FSW) controls spacecraft such as satellites, orchestrating mission-critical functions including attitude control, orbit maintenance, payload processing, and telecommand/telemetry exchange with the ground segment [5, 11]. On a remotely operated cyber-physical system, an FSW fault is rarely just a crashed program. A single command-handler crash, reboot loop, resource exhaustion, or denial-of-service condition can sever the link to the ground and strand the spacecraft in an unsafe state, with no operator able to step in and reset it [14, 15].

These risks make pre-deployment bug discovery especially important. To address this need, recent studies have explored fuzz testing (fuzzing in short) [9] as a practical technique for exposing latent bugs and robustness violations in satellite FSW and firmware [5, 12, 14].

However, traditional byte-level mutation is limited when fuzzing FSW because many relevant inputs are highly structured. This is especially problematic at the telecommand interface through which ground operators control spacecraft behavior. Telecommands typically identify a target application or module, specify an operation, and carry command-specific parameters. For example,

NASA cFS routes Software Bus packets by message identifier, while applications use command or function codes to select command handlers [11]. Because command dispatchers and handlers validate these fields before executing the requested functionality, inputs with invalid message identifiers, function codes, or parameter layouts are rejected early [5]. Consequently, byte-level fuzzers such as AFL [16] often corrupt command syntax and remain confined to shallow rejection paths.

This reflects the broader structured-input fuzzing problem: grammar-blind mutation often stalls in parser or validation logic, whereas grammar- or syntax-aware fuzzers preserve input validity and reach deeper program states [2, 7, 13]. Applying such fuzzers to FSW telecommands, however, requires a target-specific command-dispatch specification, not just a generic grammar: it must capture which message identifiers and function codes select each handler, how the payload is interpreted, and what parameter constraints must hold. Such a specification is rarely available as a standalone schema. Dispatch logic is spread across source files, command tables, callbacks, and call relationships, while parameter meanings and constraints are documented separately in comments, manuals, or other mission artifacts. Recovering them therefore requires manually correlating code and documentation, which is time-consuming and scales poorly with the number of applications and commands.

To address this challenge, this paper proposes SatFuzz, a novel fuzzer that leverages FSW command syntax specifications extracted from source code and documentation. The key idea is to guide an LLM with program-analysis context: traditional static analysis first localizes command-relevant code and call relationships, and the LLM then combines this context with source-code comments, documentation, and developer manuals. From these artifacts, SatFuzz infers message identifiers, function codes, parameter schemas, and field constraints. The resulting specification guides command-syntax-aware mutation, preserving command validity while reducing the manual effort required from analysts.

We evaluate SatFuzz on three representative FSW systems: NASA's core Flight System (cFS), SUCHAI, and the Command Centric Architecture (C2A) [5, 10, 11]. Their diverse implementations and command-processing architectures allow us to assess whether the extraction method generalizes across codebases. Our evaluation shows that the extracted specifications preserve command validity, improve access to command-handling logic, reduce manual input construction effort, and lead to the discovery of 18 previously unknown bugs.

We summarize our contributions as follows:

- We propose SatFuzz, a fuzzing framework that extracts FSW command syntax specifications and uses them for command-sequence fuzzing.
- We evaluate SatFuzz on three representative FSW systems, showing that it preserves command validity, improves command-handler coverage, and reduces manual input-construction effort.
- We report 18 previously unknown bugs in NASA cFS, SUCHAI, and C2A.
- We make SatFuzz publicly available: `https://github.com/SoftSec-KAIST/SatFuzz`

2 Motivating Command-Syntax-Aware Fuzzing

We first explain why telecommand[3] syntax is central to fuzzing satellite flight software (FSW), then introduce the FSW systems used in our evaluation.

2.1 Why Telecommand Syntax Matters

Satellite flight software (FSW) is command-driven embedded software that operates spacecraft functions through ground-issued telecommands. In many FSW architectures, a telecommand identifies the target application or module, specifies the requested operation, and carries command-specific parameters [5, 10, 11]. This syntax is essential for FSW fuzzing: inputs with invalid message identifiers, function codes, or parameter layouts are rejected before they reach mission-relevant command handlers. As a result, byte-level mutation often exercises only shallow validation paths instead of deeper command-processing logic. This paper therefore focuses on extracting valid telecommand syntax so that fuzzing can preserve command validity and reach command handlers more effectively.

2.2 Command Syntax in Real FSW Systems

NASA Core Flight System (cFS). The Core Flight System (cFS) is a reusable satellite flight software framework developed by NASA to support diverse mission environments through modularity and portability [11]. cFS delivers ground commands to applications through a message-based interface: message identifiers route packets to applications, and command codes select application-specific handlers. Thus, fuzzing cFS requires inputs that preserve both message-level routing syntax and application-level command syntax.

SUCHAI. The Satellite of the University of Chile for Aerospace Investigation (SUCHAI) flight software is CubeSat flight software developed at the University of Chile and deployed on real satellite missions [5]. SUCHAI exposes system functions through predefined command formats and parameter structures. Its centralized command dispatch rejects invalid syntax before commands reach functional modules, making SUCHAI a useful subject for studying command-syntax extraction beyond a large framework such as cFS.

Command-Centric Architecture (C2A). Command-Centric Architecture (C2A), developed by the Intelligent Space Systems Laboratory at the University of Tokyo, is an onboard software architecture designed for reusability and flexible in-orbit reconfiguration [10]. C2A describes satellite behavior through command definitions and command sequences. It also supports block commands that group multiple commands for mode processing, mode transitions, and event responses.

[3] We use the terms "telecommand" and "command" interchangeably throughout this paper.

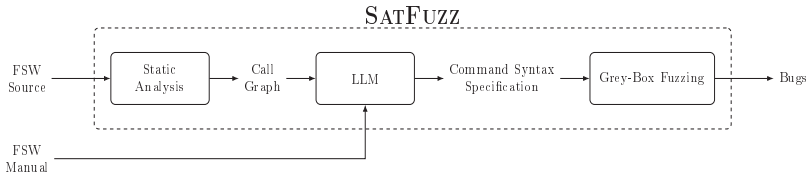


Fig. 1. Overview of SatFuzz, our command-aware fuzzing system.

This design makes command syntax central to fuzzing C2A: valid inputs must respect command definitions as well as the structure of command sequences.

Together, these systems let us evaluate command-syntax extraction across different FSW command-processing designs.

3 System Design

3.1 Overview

Figure 1 shows the overall design of SatFuzz. SatFuzz takes the source code of a target FSW and its manual—which describes commands and their formats—as input, and outputs bug-triggering command sequences. It first statically analyzes the source code to construct a call graph, which captures command dispatch and handler reachability. An LLM combines this information with the manual's command descriptions to derive a command syntax specification, which initializes and guides the seed pool for coverage-based grey-box fuzzing.

The grey-box fuzzing module generates and mutates command sequences according to this specification, executes them on the target FSW, and collects coverage and failure feedback. Sequences that increase coverage are retained in the seed pool for subsequent mutations, while failure feedback identifies bug-triggering sequences and prevents further exploration from failed executions. Through this feedback loop, SatFuzz explores state-dependent command interactions and reports the sequences that expose bugs.

The following subsections detail each component of SatFuzz. §3.2 explains how SatFuzz derives the command syntax specification from the FSW manual and the static-analysis call graph. §3.3 describes how SatFuzz generates and mutates test cases as command sequences. Finally, §3.4 explains how it selects seeds during a fuzzing campaign.

3.2 Recovering Command Syntax from Code and Documentation

SatFuzz's LLM-assisted preprocessing stage derives a command syntax specification, which the fuzzer uses to generate valid telecommands that the FSW accepts. Satellite flight software rarely defines its command interface in a single place. Instead, the relevant details are scattered across the manual and the

source code, where they appear as constant literals, dispatch logic, and command handler implementations. Gathering these scattered details by hand is time-consuming and error-prone. SatFuzz therefore uses an LLM to collect and organize them into the information needed to generate syntactically valid commands.

This stage draws on two complementary sources: the FSW manual and the static-analysis call graph. The manual provides high-level descriptions of command formats, parameters, and constraints, whereas the source code contains the concrete implementation of command parsing, dispatch, and validation. The call graph, which SatFuzz extracts from the source code, captures the structural relationships between command handlers and internal routines. By combining the manual with this call-graph context, the LLM identifies command types, command fields, parameter constraints, and candidate argument values that are likely to pass input validation.

The preprocessing stage outputs the command syntax specification as a JSON document. The specification describes the structure of valid commands: module identifiers, function codes, argument layouts, parameter types, and the constraints that SatFuzz infers from the manual and the source code. It may also list candidate values, such as file names, application names, numeric constants, and mode values, that the command interface is likely to accept. Because the specification is machine-readable, the fuzzer can programmatically generate valid command variants and mutate their parameters without violating the expected command format.

3.3 Test Case Generation and Mutation

In SatFuzz, a test case is an ordered sequence of commands rather than a single command. This choice reflects the state-dependent nature of satellite flight software: the effect of a command depends on the internal state established by the commands that precede it. A command issued in isolation typically exercises only shallow input validation, whereas the same command can reach much deeper logic once an earlier command has initialized a module, changed a configuration value, or allocated a resource. Many FSW behaviors are therefore reachable only through a specific combination of commands, and a fuzzer that mutates a single command at a time cannot expose them.

SatFuzz generates each test case from the command syntax specification derived in §3.2. For every command in a sequence, it selects a command type and fills in the required fields with values drawn from the specification, so that each command is individually well-formed and likely to pass input validation. It then assembles these commands into a sequence, allowing later commands to build on the state produced by earlier ones. By treating the entire sequence as the unit of generation and mutation, SatFuzz can construct the state-building prefixes needed to drive the target into deep, state-dependent execution paths.

SatFuzz produces new test cases by mutating existing seeds at two levels, always preserving the command syntax so that mutated sequences remain valid. At the parameter level, it replaces a command's arguments with other

values allowed by the specification while keeping the command type unchanged. At the sequence level, it concatenates coverage-increasing seeds, appends new commands after a useful prefix, and reorders or interleaves commands to explore order-dependent behavior. Because these mutations operate on structured commands rather than raw bytes, every generated sequence stays well-formed, focusing the search on state-dependent behavior instead of on inputs the FSW rejects outright.

3.4 Seed Selection and Prioritization

SatFuzz follows the standard coverage-guided loop [9]: it maintains a seed pool of command sequences and retains a sequence whenever it discovers new coverage, measured by comparing the current execution against the accumulated global coverage. Two aspects are specific to our setting. First, SatFuzz also retains sequences that trigger abnormal behavior, such as crashes, hangs, or assertion failures, and gives them higher scheduling priority, since such behavior is the primary target of fuzzing. Second, when two sequences reach equivalent coverage, SatFuzz prefers the shorter one, which lowers execution cost and makes subsequent mutations more efficient. Because the effect of a command depends on the commands that precede it, each seed stores its full sequence, letting the fuzzer reuse the state-building prefix in later mutations.

4 Implementation

We implemented SatFuzz in approximately 12K SLoC of Python. Rather than building on an existing grey-box fuzzer such as AFL [16] or AFL++ [4], we wrote our fuzzer from scratch to retain full control over its mutation strategy, which lets us isolate and measure the impact of each fuzzing component. The fuzzer generates and mutates command sequences, schedules seeds, sends commands to the target FSW, and collects coverage and failure feedback. In our setup, cFS and C2A run in containers, while SUCHAI runs natively; in all cases, the fuzzer drives each target through its existing command-delivery interface. For each execution, it sends a generated command sequence and collects the resulting coverage and failure feedback to update the seed pool.

During preprocessing, we used GPT-5.5 to analyze the target command interface and extract values for grammar-preserving command generation. We provided the CodeQL-generated call graph, command manual, and relevant code, and instructed the LLM to emit a predefined JSON format pairing each command with its parameters. The fuzzer then uses this output to construct valid commands and mutate their parameters without breaking the command format.

5 Evaluation

In this section, we evaluate SatFuzz to answer the following research questions.

Table 1. Numbers of commands extracted by SatFuzz, commands used for fuzzing, and their parameter counts.

			# parameters										
Target	Extracted	Used	0	1	2	3	4	5	6	7	8	9	10
cFS	101	**100** *	30	36	24	8	–	1	1	–	–	–	–
SUCHAI	64	**64**	19	22	16	2	–	3	1	–	–	–	1
C2A	182	**182**	54	69	33	16	6	2	2	–	–	–	–

* One cFS command was excluded because it was marked as deprecated.

- **RQ1.** What command specifications does SatFuzz generate from the target FSW, and what are their basic characteristics? (§5.2)
- **RQ2.** How effectively does SatFuzz improve code coverage compared with a state-of-the-art fuzzer? (§5.3)
- **RQ3.** How effectively does SatFuzz expose bugs compared with a state-of-the-art fuzzer? (§5.4)
- **RQ4.** Are any of the discovered bugs caused by deep state-dependent behavior? (§5.5)

5.1 Experimental Setup

We evaluated SatFuzz on cFS v7.0.0, SUCHAI v3.0.0-rc.2,[4] and C2A v4.5.1. Each target was built and executed in a Docker-based environment, and SatFuzz delivered generated command sequences to the target. All experiments were conducted on a workstation with Ubuntu 22.04.5 LTS, an AMD Ryzen Threadripper PRO 3955WX 16-core processor, and 128 GB of RAM. We used Docker 29.5.3 for target execution and CodeQL 2.25.5 to extract static call graphs from the target FSW source code. For each target and fuzzer, we ran a 12-hour fuzzing campaign, measured code coverage, and collected bug-triggering inputs.

5.2 RQ1. Command Specification

The command specification bounds what SatFuzz can reach: missing commands leave handlers unreachable, while missing parameters force unstructured mutation. We therefore first characterize what SatFuzz extracts across the three targets before evaluating its impact on coverage and bug discovery.

Table 1 shows the command specifications extracted by SatFuzz. Across the three targets, SatFuzz extracted all defined commands: 101 commands from cFS, 64 from SUCHAI, and 182 from C2A. We validated the extracted specifications with scripts and confirmed that the command names, parameter counts, and parameter types match the target command definitions. For cFS, we

[4] The exact SUCHAI revision was `3.0.0.rc-2-25-g1c59222`.

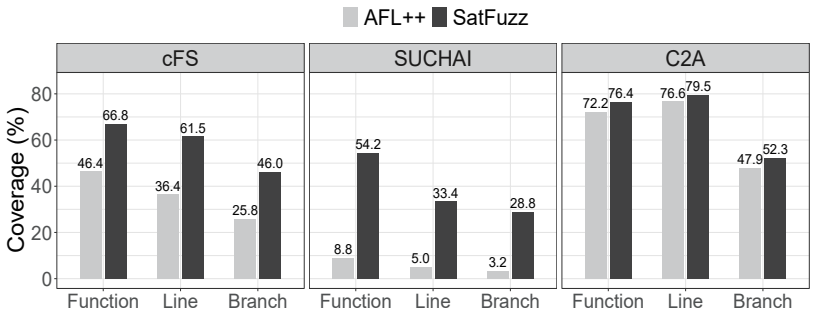


Fig. 2. Coverage comparison between AFL++ and SatFuzz.

used 100 of the 101 extracted commands, excluding one command after confirming that it was marked as deprecated in the cFS command list. Most extracted commands take zero to two parameters, while a smaller number expose more complex multi-field interfaces. These results show that SatFuzz recovers not only command identifiers but also the parameter structures needed to generate fuzzing inputs across diverse FSW command interfaces.

5.3 RQ2. Coverage Improvement

We compare SatFuzz against AFL++ [4], a state-of-the-art byte-level fuzzer, in terms of function, line, and branch coverage on the three FSW targets. We ran both tools for five independent 12-hour sessions on each target with an empty seed corpus. As Figure 2 shows, SatFuzz achieves significantly higher coverage than AFL++ across all targets and all three metrics, reaching 1.09–9.06× higher branch coverage in particular. This result shows that the command syntax specification extracted by SatFuzz helps the fuzzer reach command-processing logic more effectively than byte-level mutation.

The improvement is consistent across all evaluated FSW targets. On cFS, SatFuzz covered 4,105 additional lines, 1,463 additional branches, and 284 additional functions on average compared with AFL++. On SUCHAI, SatFuzz covered 1,370 additional lines, 593 additional branches, and 148 additional functions on average. On C2A, SatFuzz covered 355 additional lines, 119 additional branches, and 45 additional functions on average. These results show that SatFuzz improves not only a single coverage metric, but all measured coverage metrics across all targets.

Inspecting where the additional coverage came from confirms that it stems from inferring command syntax. On cFS, the gains span application execution, event handling, and table management, along with the common service APIs these modules call; many of the newly covered paths guard their logic with state checks and branches that execute only for commands carrying valid identifiers, parameters, and execution contexts. On SUCHAI, the gains concentrate in

Table 2. Bugs discovered by AFL++ and SatFuzz.

	Total		Discovery overlap		
Target FSW	AFL++	SatFuzz	Both	AFL++ only	SatFuzz only
cFS	0	**2**	0	0	2
SUCHAI	0	**2**	0	0	2
C2A	6	**11**	3	3	8

telemetry-related code, where invalid inputs are otherwise rejected before they can update state or reach payload handling. On C2A, they appear in communication modules such as driver supervision, telemetry management, and CCSDS telemetry-command handling, whose paths require valid command structures to exercise. In every case, the extra coverage lies behind validation that only syntactically valid commands can pass, showing that inferring command syntax is what lets SatFuzz reach this logic where byte-level mutation cannot.

Overall, RQ2 shows that SatFuzz explores satellite FSW more effectively than AFL++ by generating valid command sequences and retaining coverage-increasing sequences as seeds.

5.4 RQ3. Bug Discovery

We next compare the bugs that SatFuzz and AFL++ discover on the three FSW targets. We triaged every reported bug manually and counted each distinct externally observable fault, namely crashes, hangs, abnormal termination, and loss of housekeeping telemetry. Table 2 reports, for each target, the number of bugs found by each fuzzer (columns 2–3) and how they split into shared bugs and bugs unique to one fuzzer (columns 4–6). Overall, SatFuzz finds more than twice as many bugs as AFL++ across the three targets. Crucially, the extra bugs are not simply more of the same: as we detail in §5.5, many are state-dependent bugs that surface only after a specific command sequence is issued, which AFL++ cannot construct. These results show that command-aware fuzzing exposes faults that byte-level fuzzing leaves undiscovered in FSW.

Columns 4–6 of Table 2 show how many bugs each fuzzer found that the other did not. The result shows a clear advantage for SatFuzz. Out of 18 total bugs, 12 were found only by SatFuzz, whereas AFL++ found only 3 bugs that SatFuzz did not. Thus, the bug-discovery gap is not merely a difference in total counts: SatFuzz also dominates AFL++ in the number of unique bugs discovered.

We further analyzed bugs found by only one fuzzer to understand this gap. For bugs found only by AFL++, SatFuzz had extracted and executed the corresponding commands but used valid parameter values, so it did not trigger crashes requiring malformed or out-of-range values. AFL++ could reach these cases by corrupting individual fields through byte-level mutation. In contrast, bugs found only by SatFuzz required inputs much harder to obtain this way,

including specific strings, valid file or application names, and commands with multiple well-formed parameters. These bugs were hidden behind command syntax and handler-level validation, making byte-level mutation unlikely to reach the vulnerable code. By generating commands from the extracted specification, SatFuzz preserves valid syntax, produces meaningful parameter values, reaches the corresponding handlers, and exposes bugs AFL++ misses.

Overall, RQ3 shows that SatFuzz is more effective at finding bugs in satellite FSW in our evaluation: it finds more bugs overall and more unique bugs, including those requiring valid command construction rather than accidental byte-level corruption. We reported the discovered bugs to the upstream project maintainers.

5.5 RQ4. Case Study: State-Dependent Bugs

We now ask why SatFuzz finds bugs that AFL++ misses. To answer this, we divide the discovered bugs into two classes. A *single-command bug* is triggered by one command regardless of prior execution, such as unintended memory access or divide-by-zero caused directly by command parameters. A *state-dependent bug*, in contrast, requires preceding commands to place the FSW in a particular internal state before the triggering command executes.

This distinction explains the gap between the two fuzzers. AFL++ found only single-command bugs, all six in C2A. In contrast, every state-dependent bug was found only by SatFuzz. This is because such bugs require a valid command prefix followed by a valid triggering command, whereas byte-level mutation is unlikely to preserve both the order and syntax of the sequence.

One cFS bug illustrates this behavior. The first command enables a state in which events are routed through the telemetry path, and the second command changes the execution context of the telemetry consumer. Only this combined state drives the target into an error-handling path that eventually segfaults; neither command crashes the target alone. On C2A, one bug has a similar two-command structure: an earlier command sets an internal variable to zero, and a later command uses it as a modulo divisor without a zero check, causing a SIGFPE.

These cases show why sequence preservation matters for FSW fuzzing. By retaining coverage-increasing command sequences as seeds, SatFuzz can reuse state-building prefixes and expose failures that emerge only from interactions across multiple valid commands.

6 Related Work

Stateful fuzzing has been widely studied in the context of protocol testing [1, 8]. However, FSW has been largely overlooked, despite its unique characteristics and critical role in satellite operations. Prior FSW fuzzing work shows that command interfaces can be tested, but existing approaches do not combine automatic command-syntax extraction with coverage-guided exploration of state-dependent command interactions.

Broberg *et al.* [3] applied byte-level mutation to cFS, but byte-level mutation does not preserve FSW command syntax and often stalls before deep command-handling logic. Gutierrez *et al.* [5,6] proposed command-aware random sequence fuzzing for SUCHAI, but their approach manually extracts command information and generates random sequences with fixed lengths. It mainly targets individual command-handler failures rather than coverage-guided exploration of state-dependent behavior. To our knowledge, our work is the first to combine LLM-assisted command syntax extraction with grey-box fuzzing to systematically explore state-dependent command interactions in FSW.

7 Limitations

SatFuzz relies on the quality of the command syntax specification extracted by the LLM. If the manual is incomplete, the source code is difficult to analyze, or the LLM misses important parameter constraints, the generated commands may fail to exercise some valid behaviors. Moreover, because the extracted specification tends to describe values that preserve command validity, SatFuzz may miss robustness bugs that require malformed or out-of-range parameter values.

Improving the precision and diversity of LLM-extracted specifications, for example by adding boundary and invalid-value candidates, is orthogonal to our main goal. Our focus is to show that once useful command syntax information is available, it can guide grey-box fuzzing toward deeper command-processing and state-dependent execution paths.

8 Conclusion

In this paper, we introduced SatFuzz, a command-syntax-aware grey-box fuzzing framework for satellite flight software. SatFuzz takes FSW manuals and source code as input and produces bug-triggering command sequences as output. It uses LLM-assisted preprocessing to extract command syntax specifications from these inputs. Using the extracted command syntax specification, SatFuzz generates and mutates valid command sequences while using coverage feedback to retain sequences that explore new execution paths. We evaluated SatFuzz on three satellite flight software targets: NASA cFS, SUCHAI, and C2A. SatFuzz achieved higher coverage than AFL++ on all targets. Across both fuzzers, our evaluation discovered 18 distinct previously unknown bugs, 15 of which were found by SatFuzz. We reported the bugs to the developers.

Acknowledgements

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2025-02263143, Development of Cybersecurity Threat Response Technologies for Satellite Ground Stations, and No.2021-0-01332, Developing Next-Generation Binary Decompiler).

References

1. Ba, J., Böhme, M., Mirzamomen, Z., Roychoudhury, A.: Stateful greybox fuzzing. In: Proceedings of the USENIX Security Symposium. pp. 3255–3272. Boston, MA (2022)
2. Blazytko, T., Aschermann, C., Schlögel, M., Abbasi, A., Schumilo, S., Wörner, S., Holz, T.: GRIMOIRE: Synthesizing structure while fuzzing. In: Proceedings of the USENIX Security Symposium. pp. 1985–2002 (2019)
3. Broberg, R.: Fuzzing NASA core flight system software. `https://www.aerospacevillage.org/defcon-29`
4. Fioraldi, A., Maier, D., Eißfeldt, H., Heuse, M.: AFL++ : Combining incremental steps of fuzzing research. In: Proceedings of the USENIX Workshop on Offensive Technologies (2020)
5. Gutierrez, T., Bergel, A., Gonzalez, C.E., Rojas, C.J., Diaz, M.A.: Toward applying fuzz testing techniques on the SUCHAI nanosatellites flight software. In: Proceedings of the IEEE Biennial Congress of Argentina. pp. 1–4 (2020)
6. Gutierrez, T., Bergel, A., Gonzalez, C.E., Rojas, C.J., Diaz, M.A.: Systematic fuzz testing techniques on a nanosatellite flight software for agile mission development. IEEE Access **9**, 114008–114021 (2021)
7. Kim, H., Kim, S., Lee, J., Cha, S.K.: AsFuzzer: Differential testing of assemblers with error-driven grammar inference. In: Proceedings of the International Symposium on Software Testing and Analysis. pp. 1099–1111 (2024)
8. Kim, W., Nam, H., Tran, M., Jalilov, A., Liang, Z., Cha, S.K., Kang, M.S.: Fork state-aware differential fuzzing for blockchain consensus implementations. In: Proceedings of the International Conference on Software Engineering. pp. 3048–3059 (2025)
9. Manès, V.J.M., Han, H., Han, C., Cha, S.K., Egele, M., Schwartz, E.J., Woo, M.: The art, science, and engineering of fuzzing: A survey. IEEE Transactions on Software Engineering **47**(11), 2312–2331 (2021). `https://doi.org/10.1109/TSE.2019.2946563`
10. Nakajima, S., Takisawa, J., Ikari, S., Tomooka, M., Aoyanagi, Y., Funase, R., Nakasuka, S.: Command-centric architecture C2A: Satellite software architecture with a flexible reconfiguration capability. Acta Astronautica **171**, 208–214 (2020)
11. NASA: Core flight system. `https://github.com/nasa/cfs`
12. Scharnowski, T., Buchmann, F., Wörner, S., Holz, T.: A case study on fuzzing satellite firmware. In: Proceedings of the Workshop on the Security of Space and Satellite Systems (SpaceSec) (2023)
13. Wang, J., Chen, B., Wei, L., Liu, Y.: Superion: Grammar-aware greybox fuzzing. In: Proceedings of the International Conference on Software Engineering. pp. 724–735 (2019)
14. Willbold, J., Schloegel, M., Göhler, F., Scharnowski, T., Bars, N., Wörner, S., Schiller, N., Holz, T.: Scaling software security analysis to satellites: Automated fuzz testing and its unique challenges. In: Proceedings of the IEEE Aerospace Conference. pp. 1–12 (2024)
15. Willbold, J., Schloegel, M., Vögele, M., Gerhardt, M., Holz, T., Abbasi, A.: Space odyssey: An experimental software security analysis of satellites. In: Proceedings of the IEEE Symposium on Security and Privacy. pp. 1–19 (2023)
16. Zalewski, M.: American Fuzzy Lop. `http://lcamtuf.coredump.cx/afl/`

Practical Fuzzing Saturation Detection with Benford's Law

Jungwoo Lee, Haeun Lee, Sangjun Park, and Sang Kil Cha

KAIST, Daejeon, Korea
jwlee2217@kaist.ac.kr, haeunlee@softsec.kaist.ac.kr,
{sangjuns,sangkilc}@kaist.ac.kr

Abstract. Practitioners of fuzzing often face a dilemma: can we stop the fuzzing campaign now, or is the fuzzer still making meaningful progress? Developing a reliable stopping criterion is challenging because existing approaches either ignore the fuzzer's internal state, require users to choose arbitrary thresholds for saturation, or depend on metrics whose behavior varies across programs and fuzzing campaigns. A prior registered report [15] proposed a promising way out of this dilemma by using Benford's law (the first-digit law): given a grey-box fuzzer's inherent randomness, the numerical data it produces should follow the patterns of naturally occurring data once coverage has saturated, so deviation from Benford's law signals that the fuzzer is still making progress. That report demonstrated the idea through a small-scale proof-of-concept analysis, but left open whether Benford's law can serve as a practical stopping criterion in large, real-world fuzzing campaigns. In this paper, we provide the first large-scale empirical evaluation of this criterion: we run the state-of-the-art fuzzer AFL++ on 30 programs from the UniFuzz and FuzzBench benchmarks for two weeks each. Our results show that the Benford's-law-based criterion detects saturation effectively and compares favorably with existing stopping criteria, providing the first empirical evidence that Benford's law can be a practical solution to threshold-free fuzzing saturation detection.

1 Introduction

Fuzzing is one of the most widely used techniques for discovering software vulnerabilities [21], but practitioners still lack a reliable way to decide when a fuzzing campaign has run long enough. This decision is essential for using fuzzing resources effectively: analysts can stop campaigns that have already saturated and redirect their effort to targets that may still yield more vulnerabilities. Stopping too early, on the other hand, wastes the opportunity to find more vulnerabilities in the current target. A reliable stopping criterion is therefore important in practice.

Unfortunately, existing stopping criteria remain insufficient. A hard time limit—e.g., the 24 hours suggested by Klees *et al.* [13]—is blind to the fuzzer's internal state (**L1**), so a fixed budget may stop some fuzzers prematurely even

after a week [18] while running others unnecessarily long. Metric-based criteria, such as stopping when code coverage saturates, instead require a threshold t to decide how much stagnation is enough to declare saturation (**L2**), and results can vary significantly with t, even for state-of-the-art approaches [17]. They also depend on the metric itself (**L3**): coverage granularity such as line versus branch changes the outcome [28], and counting unique bugs requires precise but costly triage, where even a lightweight tool like AddressSanitizer (ASAN) [26] slows fuzzing and shifts its outcome [12].

To address these limitations (**L1**–**L3**), a prior registered report [15] introduced a stopping criterion based on Benford's law [3], a leading-digit pattern used to detect anomalies in domains such as finance and elections [9, 25]. The key idea is that, as a coverage-based grey-box fuzzer saturates and its adaptive bias [4] fades, the number of bytes mutated per test case—the *muta-bytes*—steadily conforms to Benford's law. This makes the criterion aware of the fuzzer's internal state, free from user-defined thresholds, and cheap to compute.

While this prior work introduced an interesting idea, it evaluated the criterion only on a small, toy-like benchmark, leaving its practical impact unclear. This paper turns the Benford-based approach into a practical stopping rule with a novel *warm-up filtering* strategy and provides the first systematic evaluation on real-world fuzzing campaigns. Specifically, we run AFL++ on 30 programs from the UniFuzz and FuzzBench benchmarks for two weeks each, simulate several stopping criteria including the Benford's law-based one, and find that it detects saturation effectively and compares favorably with the existing criteria in both code coverage and the number of unique bugs found, without any user-defined threshold.

In summary, our contributions are as follows:

1. We introduce *warm-up filtering*, a strategy for turning Benford-based saturation detection into a practical stopping rule.
2. We conduct the first large-scale evaluation of this stopping criterion on real-world fuzzing campaigns, running AFL++ on 30 programs from UniFuzz and FuzzBench for two weeks each.
3. We assess the criterion's practicality and effectiveness, showing its negligible logging overhead and comparing it with common coverage- and crash-based stopping criteria.
4. We publicly release our implementation and the data collected during our evaluation at `https://github.com/SoftSec-KAIST/Benford-criterion-artifact` to facilitate future research on fuzzing stopping criteria.

2 Benford's Law and Conformity Testing

Benford's law observes that the leading digits of many naturally occurring numbers are *not* uniformly distributed: "1" appears far more often than "9" [3]. Because fabricated data often violate this pattern, the law has been widely used to detect anomalies and fraud [9, 25]. A prior registered report [15] conjectures that random test case generation similarly becomes an unbiased sampling process

once coverage feedback is no longer informative, causing the generated numbers to follow Benford's law.

We measure conformity to Benford's law using Mean Absolute Deviation (MAD), a common distance metric for testing the *Benfordness* of a dataset [7]. MAD averages the absolute differences between the observed and expected frequencies of the nine leading digits, without depending on the sample size as chi-square testing does, making it suitable for the large datasets produced by fuzzing. Following Nigrini [24], we treat a dataset as conforming to Benford's law when its MAD is at most 0.012. Existing works use a threshold between 0.006 and 0.012, and a MAD of 0.012 is commonly considered acceptable conformity to Benford's law, which is enough to relate Benfordness to fuzzing saturation. Unlike the program-specific thresholds required by existing stopping criteria (**L2**), this threshold is domain-independent given enough unbiased data, which makes the approach generalizable across programs and fuzzing setups.

3 The Benford-Based Stopping Criterion

A prior registered report [15] proposed a Benford's law-based stopping criterion for fuzzing, arguing that once a fuzzer stops making progress, its internal numerical quantities should resemble naturally sampled data and conform to Benford's law. However, the report provided only preliminary observational evidence on a small set of controlled programs, leaving the criterion unproven in large-scale, real-world fuzzing campaigns. We recap the criterion below, present a novel *warm-up filtering* strategy for turning it into a practical stopping rule, and systematically evaluate it at scale in §5.

3.1 Muta-bytes and the Hypothesis

A fuzzer maintains many numerical quantities during its execution, but not all of them conform to Benford's law. The registered report [15] identifies the *muta-byte*—the total number of bytes mutated to generate a test case from a seed—as a quantity that does. Unlike the size of a test case, the muta-byte is *not* artificially constrained by program-specific input formats nor biased by the sizes of the initial seeds, and it spans multiple orders of magnitude while reflecting the fuzzer's internal state, e.g., its seed pool.

The criterion rests on the hypothesis that the first digits of muta-bytes conform to Benford's law when a fuzzer reaches saturation. Intuitively, a saturated fuzzer searches with little adaptive bias, whereas a still-exploring fuzzer remains guided by coverage feedback. The registered report [15] supported this hypothesis through controlled experiments whose saturation behavior was known in advance: a no-op program, a synthetic maze generated by Fuzzle [14], and `ffmpeg`. In these cases, muta-bytes conformed to Benford's law only when the campaign saturated, and their MAD was strongly negatively correlated with edge coverage. However, the hypothesis has not yet been confirmed on a large real-world dataset, where programs, input formats, and fuzzing dynamics are substantially more diverse.

3.2 Warm-up Filtering

With the muta-byte identified as a suitable quantity, the registered report proposed a stopping criterion that terminates a fuzzing campaign once the MAD of the first-digit distribution of muta-bytes falls below the threshold of 0.012 (§2), which is the standard threshold for Benford conformity [24]. However, they also found that this criterion is not directly applicable to real-world fuzzing campaigns, where the first-digit distribution of muta-bytes fluctuates significantly at the beginning of a campaign due to the small sample size and the mutation heuristic that AFL++ employs.

To address this, we propose a simple yet effective *warm-up filtering* strategy that ignores the MAD values for the first 24 hours of a campaign, which is the minimum fuzzing time recommended by Klees *et al.* [13]. While the warm-up period can be adjusted, we find that 24 hours is sufficient to stabilize the first-digit distribution of muta-bytes in our evaluation.

3.3 Implementation

We extended AFL++'s introspection mode to compute and log the muta-byte of *every* test case generated (not merely those saved to the corpus), using a write buffer so that the logging overhead remains negligible, as we confirm in §5.3. Concretely, we sum the bytes mutated by every mutation operation applied to a seed; e.g., inserting a 4-byte interesting value, flipping one byte, and deleting two bytes yields a muta-byte of $4 + 1 + 2 = 7$. Computing the MAD on the fly is cheap, as it only requires nine buckets for the nine possible first digits. In our evaluation, however, we compute the MAD offline from the collected muta-byte data in order to minimize the impact of logging on fuzzing performance.

4 Experimental Setup

Benchmark. We evaluate 30 programs from two real-world fuzzing benchmarks: UniFuzz [8, 16] and FuzzBench [11, 22]. To cover diverse input formats, we randomly select two programs from each UniFuzz format, except for the network format, which contains only one program, for a total of 11 UniFuzz programs. We also include 19 FuzzBench programs to broaden the evaluation.

Fuzzer Configuration. We used the modified version of AFL++ [1] (as described in §3.3) to run fuzzing experiments and collect muta-byte data to test for conformance to Benford's law. Specifically, we made our modifications on AFL++ (0c054f5). When running AFL++, we used the command-line arguments and initial seeds provided by UniFuzz [8] and FuzzBench [11].

Environments. We conducted our experiments on a server equipped with 88 Intel Xeon E5-2699 v4 CPU cores with 2.20 GHz and 128 GB of memory. We ran each fuzzing campaign in an isolated Docker container, each assigned with one CPU core and 8GB of memory.

Fuzzing Hours. For the main experiment, we run each campaign for two weeks and repeat it ten times, totaling 4,200 CPU days across 30 programs. Unless stated otherwise, we report averages over the ten runs.

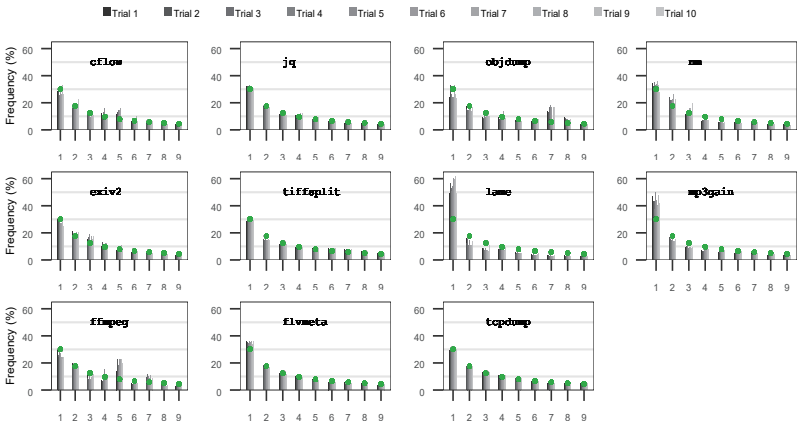


Fig. 1: Distributions of leading digits of muta-bytes obtained from 2 weeks of fuzzing on 11 UniFuzz programs with 10 repeated trials. In each panel, the x-axis is the leading digit and the y-axis is its frequency; the green dots represent the ideal Benford distribution.

5 Evaluation

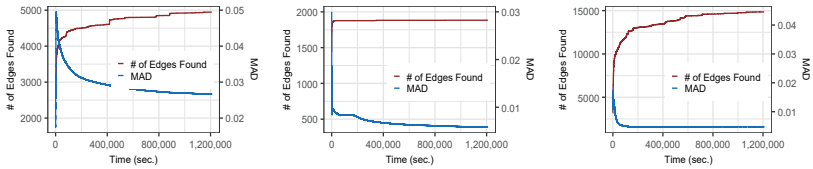
In this section, we answer the following research questions.

RQ1. Do muta-bytes conform to Benford's law for real-world programs? (§5.1)
RQ2. How does the Benford-based criterion compare to existing criteria? (§5.2)
RQ3. Is the Benford-based criterion practical given the overhead of logging muta-bytes? (§5.3)

5.1 Conformance to Benford's Law

First-Digit Distribution of Muta-bytes. Does Benfordness of muta-bytes hold for a various set of real-world programs? To answer the question, we performed a first-digit distribution analysis of muta-bytes on the 11 UniFuzz programs that we ran for 2 weeks in §4. Figure 1 shows the distribution of leading digits of muta-bytes at the end of each fuzzing campaign. We can clearly see non-uniform distributions of leading digits for all programs. Indeed, most of them closely follow Benford's law in that they exhibit the expected distribution of first digits where "1" is the most common leading digit, followed by "2", "3", and so on. Nevertheless, we observed some deviations from the expected distribution in a few cases such as `objdump` and `ffmpeg`.

MAD Values and Saturation in Fuzzing. To quantitatively measure the conformity of the distributions to Benford's law, we computed the MAD between the observed distribution and the expected Benford distribution. The last column of Table 1 presents the MAD values for each program after 2 weeks of fuzzing.



(a) `objdump`. (b) `jq`. (c) `tcpdump`.

Fig. 2: Coverage and MAD over time for three representative UniFuzz programs: `objdump`, whose coverage keeps increasing (high MAD, not saturated); `jq`, which saturates early (low MAD); and `tcpdump`, which keeps gaining coverage yet reaches a low MAD owing to its high throughput.

Only four out of the eleven programs have an average MAD value below the acceptable threshold of 0.012. This does not contradict the visual distributions in Figure 1: most programs already exhibit a Benford-like shape, but the MAD values show that seven of them have not yet reached strict quantitative conformance after two weeks of fuzzing. Under our hypothesis, this residual deviation is informative because it suggests that the fuzzer may still have unexplored code or input subspaces to reach. We therefore further analyzed each program to see whether its coverage had reached saturation.

Four Programs with Low MAD. Among the four programs with low MAD (`jq`, `tiffsplit`, `flvmeta`, and `tcpdump`), `jq` and `flvmeta` are the clearest cases of strict coverage saturation: they are simple command-line tools whose invocation arguments tightly limit the input space, and their coverage plateaued early, as Figure 2 shows for `jq`. For `jq`, we manually verified this by confirming that the fuzzer could no longer reach any new code in the program. Such full saturation is unlikely for most real-world programs, where large input spaces and complex path constraints leave some code unexplored even after long campaigns. Indeed, `tiffsplit` and `tcpdump` kept gaining coverage throughout the campaign, so their low MAD values do not mean that every possible line was covered. Rather, they show the main practical implication of Benfordness: for these high-throughput programs, fuzzing had become ineffective at discovering new coverage at a meaningful rate, even though additional code remained to explore.

Seven Programs with High MAD. For the other seven programs, the high MAD values should not be read as evidence that the fuzzer was stuck. Rather, they indicate that the campaign was still discovering new code or paths often enough to keep the muta-byte distribution from stabilizing into a good-enough Benford distribution within two weeks. The cases of `objdump` and `ffmpeg` make this clear from their still-increasing coverage, with `objdump` shown in Figure 2; `nm`, which shares code with `objdump` but has a higher throughput and 4× more test cases, is expected to become more conformant as more test cases accumulate.

The remaining cases show slower or more irregular progress rather than a complete lack of progress. `exiv2` occasionally reaches image formats beyond the initial jpeg seed (only one of ten trials recovered a raw Minolta format),

Table 1: Average test cases generated and final MAD for each of the 30 UniFuzz and FuzzBench programs over 10 two-week fuzzing runs. Test-case counts are in millions; grey cells mark MAD ≤ 0.012.

UniFuzz

Program	SLoC	# Test Cases	Final MAD
cflow	29K	234.6M	0.0173
jq	17K	251.8M	0.0070
objdump	160K	334.3M	0.0280
nm	160K	1,385M	0.0220
exiv2	80K	591.5M	0.0134
tiffsplit	59K	1,304M	0.0108
lame	41K	302.6M	0.0557
mp3gain	8K	432.0M	0.0302
ffmpeg	1,015K	13.38M	0.0375
flvmeta	14K	1,656M	0.0118
tcpdump	112K	1,175M	0.0048

FuzzBench

Program	SLoC	# Test Cases	Final MAD
bloaty	626K	106.9M	0.0188
curl	154K	77.94M	0.0079
freetype2	119K	816.4M	0.0154
harfbuzz	118K	504.0M	0.0111
jsoncpp	11K	589.4M	0.0065
lcms	45K	611.0M	0.0087
libjpeg-turbo	59K	2,753M	0.0072
libpcap	57K	1,843M	0.0074
libpng	63K	4,675M	0.0080
libxml2	212K	142.7M	0.0100
mbedtls	226K	2,020M	0.0093
openssl	428K	1,186M	0.0077
openthread	284K	1,007M	0.0092
php	1379K	806.7M	0.0094
proj4	409K	81.02M	0.0070
re2	28K	541.1M	0.0052
sqlite3	443K	230.8M	0.0148
systemd	541K	275.6M	0.0094
woff2	3K	3,927M	0.0103

`lame` and `mp3gain` require specific floating-point inputs to trigger some new code and also exhibit low throughput (Table 1), and `cflow` needs grammar-aware inputs to explore its input space more systematically. These constraints make new coverage discoveries sparse and uneven, which keeps the distribution away from the Benford threshold even though the fuzzer still occasionally finds new behavior.

In summary, our results confirm our hypothesis. When fuzzing continues to find new coverage, even sparsely, the first-digit distributions of muta-bytes may not strictly conform to Benford's law. On the other hand, when new coverage discovery becomes ineffective, the muta-byte distribution tends to conform to Benford's law, which suggests that the degree of conformity can be used as a practical stopping criterion for fuzzing. `tiffsplit` and `tcpdump` are exceptions: their high throughput lets the fuzzer saturate an input subspace quickly, a limitation we discuss in §6. **Overall, the degree of conformity to Benford's law serves as a practical measure for detecting fuzzing coverage saturation.**

FuzzBench provides additional support for this observation. We obtain the same measurements for the 19 FuzzBench programs, reported alongside the UniFuzz programs in Table 1. The pattern is even clearer on FuzzBench: 16 of the 19 programs conform to Benford's law, yielding a conformance rate of about 84%, much higher than on UniFuzz. This difference is expected because FuzzBench targets are library harnesses with narrow, single-API input spaces, so AFL++ reaches coverage saturation sooner than it does on UniFuzz's full applications. The coverage and MAD trends show the same behavior. For conforming programs such as `libpng`, the MAD drops below the threshold as coverage gains plateau;

Table 2: Average number of edges covered and unique bugs found when using the Benford's law-based criterion and the existing coverage- and crash-based stopping criteria.

Program	Edge Coverage			Unique Bugs		
	Benford's Law-Based	Coverage-Based	Crash-Based	Benford's Law-Based	Coverage-Based	Crash-Based
`cflow`	**1128.6**	1127.5	1128.1	**1.2**	1.0	1.0
`jq`	1874.3	**1875.5**	1873.7	**0.0**	**0.0**	**0.0**
`objdump`	**4849.0**	4748.8	4532.1	**4.9**	4.2	3.0
`nm`	**3232.4**	3088.2	3097.2	**9.3**	7.8	7.9
`exiv2`	4216.3	**4354.8**	4293.3	2.8	**3.5**	2.8
`tiffsplit`	**1668.0**	1651.4	1667.4	2.6	2.5	**2.7**
`lame`	**3295.9**	3285.8	3286.6	**4.0**	**4.0**	**4.0**
`mp3gain`	**917.6**	906.7	911.1	**6.8**	6.0	6.3
`ffmpeg`	**24160.6**	23480.0	11853.9	**2.5**	2.4	0.0
`flvmeta`	**230.0**	**230.0**	**230.0**	**1.0**	**1.0**	**1.0**
`tcpdump`	11793.4	**14264.6**	11839.4	0.2	**0.4**	0.2

for the few non-conforming programs such as `sqlite3`, continued coverage gains keep the MAD high.

5.2 Comparison with Existing Criteria

To evaluate the effectiveness of the Benford's law-based criterion, we compared it against two stopping criteria commonly used in practice: coverage-based and crash-based criteria [17]. The coverage-based criterion stops a campaign when no new coverage is achieved for a fixed time window t, while the crash-based criterion stops when no new crash is triggered for t. Thus, both criteria require a threshold t to be chosen.

For each criterion, we determined the time at which the fuzzing campaign would stop, which we refer to as *stop-time*. For the Benford's law-based criterion, the stop-time is the first time at which the MAD reaches the acceptable threshold of 0.012. Here, we applied the warm-up filtering described in §3.2, ignoring the MAD values for the first 24 hours of a campaign to minimize the influence of the mutation heuristic that AFL++ employs. For the coverage- and crash-based criteria, we used 12 different thresholds for t, ranging from 2 hours to 24 hours in 2-hour increments, to reduce the effect of any single arbitrary threshold. This threshold set is broader than the state-of-the-art method, which used only 2, 4, 6, and 8 hours [17].

By simulating the termination of 2-week-long fuzzing campaigns based on the obtained stop-times, we compared the three criteria in both the coverage achieved (§5.2) and the number of unique bugs found (§5.2).

Code Coverage Achieved For all fuzz runs, we obtained the total code coverage achieved at the stop-times determined based on each of the three criteria. Note, we used AFL++'s edge coverage as the measure of code coverage. For coverage-

based and crash-based criteria, the average coverage is computed by taking the mean of coverage achieved by all 12 thresholds.

The *Edge Coverage* columns of Table 2 present the results. **The Benford-based criterion achieves the best coverage on most of the 11 programs**, surpassing both the coverage- and crash-based criteria. Its two clear shortfalls are `exiv2` and `tcpdump`, which fall short for different reasons. `exiv2` only rarely discovers new image formats, so its MAD drops below the threshold before these late coverage gains, causing premature termination in 7 of 10 runs. For `tcpdump`, the high throughput pushes the MAD below the threshold before the coverage plateaus. In both cases the criterion stops early and misses some edges.

Number of Unique Bugs Found For each fuzz run, we determined the total number of unique bugs found by triaging all crashes that were reported until the respective stop-times. Note, we followed the triage procedure of UniFuzz [8], where the GDB stacktraces [2] and the ASAN reports [26] are used to match the crashes to one of the CVEs in their database. The *Unique Bugs* columns of Table 2 present the results. The same holds for bug finding. **The Benford-based criterion finds the most unique bugs overall**, with the highest total across the 11 programs and the best or matching count on most of them.

5.3 Performance Overhead

To quantify the cost of logging muta-bytes, we measured the fuzzing throughput (executions per second) of AFL++ in three configurations—without the introspection mode, with AFL++'s original introspection mode, and with the modified mode that logs muta-bytes (§3.3)—averaged over three runs on `cflow` and `objdump`. The modified mode is on average only 15.33% slower than running AFL++ without the introspection mode and 5.09% slower than AFL++'s original introspection mode. This small additional cost comes from computing and logging the muta-byte of *every* generated test case, whereas AFL++'s introspection mode records data only for the test cases saved to the corpus—a tiny fraction of all test cases generated. Importantly, this cost stems from *logging* every muta-byte for our offline analysis; rather than logging, a deployed criterion could compute the MAD on the fly inside AFL++ (§3.3), which would reduce the overhead much further. **The performance overhead of logging muta-bytes is therefore small and does not significantly affect the fuzzing throughput.**

6 Discussion and Future Work

This paper does not claim the Benford's law-based stopping criterion as a definitive solution for all fuzzing scenarios; rather, it assesses how well the criterion holds in practice. Our large-scale evaluation confirms that the muta-bytes derived from fuzzing exhibit a *Benford-like* distribution across a wide range of real-world programs.

Although the Benford's law-based stopping criterion addresses many limitations of existing solutions, such as the dependence on thresholds and detection metrics of the coverage-based and the crash-based stopping criteria, it is not without its own limitations. Notably, the throughput of the program under test can affect the reliability of the Benford's law-based approach, as demonstrated with the results on `tiffsplit` and `tcpdump` in §5.1. For some programs with large input spaces, exceptionally high throughput lets the fuzzer quickly saturate the input subspace reachable with the current seed pool. When this happens, the internal state of the fuzzer can behave as though it has reached saturation, even when the entire input space has not been fully explored. This causes the data derived from fuzzing, such as the muta-bytes, to conform well to Benford's law, resulting in misleading conclusions. Therefore, future studies that apply Benford's law in fuzzing should consider the throughput of the fuzzed programs when drawing conclusions based on the degree of conformity to Benford's law. Another option is to pair the MAD with a coverage signal and stop only when both indicate saturation, though this reintroduces the threshold dependence (**L2**) that the criterion is meant to avoid.

The criterion assumes AFL-like fuzzers, where the mutation operators are applied randomly, as is typical of mutation-based grey-box fuzzers. However, some fuzzers use different mutation scheduling schemes, such as MOpt [20], which uses Particle Swarm Optimization to select the mutation operator during fuzzing. Since each mutation operator alters a different range of bytes, varying the selection probabilities can yield different muta-byte distributions, so we leave extending the evaluation to such fuzzers as future work.

The analysis can also be extended beyond the most significant digit, as Benford's law predicts the distribution of the second, third, and subsequent digits as well [23]; evaluating the first several digits and their combinations could give a more complete picture of how closely a fuzzing state complies with Benford's law. We believe these are promising directions for future research.

7 Related Work

Coverage-based fuzzing inherently suffers from resource limitations—as we spend more resources on fuzzing a program, the marginal coverage increase diminishes [10]. Therefore, estimating a good stopping point is crucial for fuzzing, and there have been numerous attempts to address this problem.

Finding a good termination criterion is largely relevant to the problem of predicting the effectiveness of a fuzzer because knowing the future effectiveness of a fuzzer can help decide when to stop. Woo *et al.* [27] cast the problem of estimating the effectiveness of a fuzzer as a multi-armed bandit problem and propose a novel method to estimate the probability of discovering a bug. Böhme *et al.* [4] propose a method to estimate the discovery probability of a fuzzer in terms of code coverage. Liyanage *et al.* present an empirical study that evaluates several well-known statistical estimators of reachable coverage and discusses their limitations [18], and later propose an extrapolation method

that leverages bias-free black-box fuzzing to predict a grey-box fuzzer's future coverage rate [19]. Lipp *et al.* [17] propose to stop a fuzzing campaign when the fuzzer has not covered any new potentially vulnerable functions for a certain amount of time.

Seed selection requires evaluating the effectiveness of a seed, which is similar to estimating the effectiveness of a fuzzer; there is abundant work on it [5,6], and we refer the reader to the survey [21] for a broader review.

8 Conclusion

In this paper, we extended the prior work on the Benford's law-based stopping criterion for fuzzing [15] in two ways: we introduced a practical filtering mechanism to reduce noise in the fuzzing data, and we provided a thorough real-world evaluation of the criterion. Specifically, we ran a large-scale, two-week fuzzing campaign with AFL++ on 30 real-world programs from the UniFuzz and FuzzBench benchmarks, and showed that the Benford-based criterion can serve as a practical indicator of fuzzing saturation.

Acknowledgements

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2025-02263143, Development of Cybersecurity Threat Response Technologies for Satellite Ground Stations, and No.RS-2024-00440780, Development of Automated SBOM and VEX Verification Technologies for Securing Software Supply Chains).

References

1. American Fuzzy Lop plus plus (AFL++). `https://github.com/AFLplusplus/AFLplusplus`
2. GDB: The GNU Project Debugger. `https://www.sourceware.org/gdb`
3. Benford, F.: The law of anomalous numbers. Proc. Am. Philos. Soc. **78**(4), 551–572 (1938)
4. Böhme, M., Liyanage, D., Wüstholz, V.: Estimating residual risk in greybox fuzzing. In: ACM FSE. pp. 230–241 (2021). `https://doi.org/10.1145/3468264.3468570`
5. Böhme, M., Manès, V.J.M., Cha, S.K.: Boosting fuzzer efficiency: An information theoretic perspective. In: ACM FSE. pp. 678–689 (2020)
6. Böhme, M., Pham, V.T., Roychoudhury, A.: Coverage-based greybox fuzzing as markov chain. In: ACM CCS. pp. 1032–1043 (2016)
7. Cerqueti, R., Maggi, M.: Data validity and statistical conformity with benford's law. Chaos Solitons Fractals **144**, 110740 (2021). `https://doi.org/https://doi.org/10.1016/j.chaos.2021.110740`
8. Chen, Y.: unifuzz. `https://github.com/unifuzz`

9. Deckert, J., Myagkov, M., Ordeshook, P.C.: Benford's law and the detection of election fraud. Polit. Anal. **19**(3), 245–268 (2011). https://doi.org/10.1093/pan/mpr014
10. Gao, W., Pham, V.T., Liu, D., Chang, O., Murray, T., Rubinstein, B.I.: Beyond the coverage plateau: A comprehensive study of fuzz blockers (registered report). In: FUZZING. pp. 47–55 (2023). https://doi.org/10.1145/3605157.3605177
11. Inc., G.: FuzzBench. https://github.com/google/fuzzbench
12. Kim, T.E., Choi, J., Im, S., Heo, K., Cha, S.K.: Evaluating directed fuzzers: Are we heading in the right direction? In: ACM FSE. pp. 316–337 (2024). https://doi.org/10.1145/3643741
13. Klees, G., Ruef, A., Cooper, B., Wei, S., Hicks, M.: Evaluating fuzz testing. In: ACM CCS. pp. 2123–2138 (2018)
14. Lee, H., Kim, S., Cha, S.K.: Fuzzle: Making a puzzle for fuzzers. In: ASE. pp. 1–12 (2022)
15. Lee, J., Lee, H., Park, S., Cha, S.K.: On the applicability of Benford's law to detect saturation in fuzzing (registered report). In: FUZZING (ISSTA Companion) (2025). https://doi.org/10.1145/3713081.3731723
16. Li, Y., Ji, S., Chen, Y., Liang, S., Lee, W.H., Chen, Y., Lyu, C., Wu, C., Beyah, R., Cheng, P., Lu, K., Wang, T.: UniFuzz: A holistic and pragmatic metrics-driven platform for evaluating fuzzers. In: USENIX Security (2020)
17. Lipp, S., Elsner, D., Kacianka, S., Pretschner, A., Böhme, M., Banescu, S.: Green fuzzing: A saturation-based stopping criterion using vulnerability prediction. In: ACM ISSTA. pp. 127–139 (2023). https://doi.org/10.1145/3597926.3598043
18. Liyanage, D., Böhme, M., Tantithamthavorn, C., Lipp, S.: Reachable coverage: Estimating saturation in fuzzing. In: ICSE. pp. 371–383 (2023). https://doi.org/10.1109/ICSE48619.2023.00042
19. Liyanage, D., Lee, S., Tantithamthavorn, C., Böhme, M.: Extrapolating coverage rate in greybox fuzzing. In: ICSE. pp. 1–12 (2024). https://doi.org/10.1145/3597503.3639198
20. Lyu, C., Ji, S., Zhang, C., Li, Y., Lee, W.H., Song, Y., Beyah, R.: MOpt: Optimized mutation scheduling for fuzzers. In: USENIX Security. pp. 1949–1966 (2019)
21. Manès, V.J.M., Han, H., Han, C., Cha, S.K., Egele, M., Schwartz, E.J., Woo, M.: The art, science, and engineering of fuzzing: A survey. IEEE Trans. Softw. Eng. **47**(11), 2312–2331 (2021). https://doi.org/10.1109/TSE.2019.2946563
22. Metzman, J., Szekeres, L., Simon, L., Sprabery, R., Arya, A.: FuzzBench: An open fuzzer benchmarking platform and service. In: ACM FSE. pp. 1393–1403 (2021)
23. Miller, S.J.: Benford's Law: Theory and Applications. Princeton University Press, 1 edn. (2015)
24. Nigrini, M.J.: Benford's Law: Applications for Forensic Accounting, Auditing, and Fraud Detection. Wiley, 1 edn. (2012)
25. Nigrini, M.J.: The patterns of the numbers used in occupational fraud schemes. Manag. Audit. J. **34**(5), 606–626 (2019). https://doi.org/10.1108/MAJ-11-2017-1717
26. Serebryany, K., Bruening, D., Potapenko, A., Vyukov, D.: AddressSanitizer: A fast address sanity checker. In: USENIX ATC. pp. 309–318 (2012)
27. Woo, M., Cha, S.K., Gottlieb, S., Brumley, D.: Scheduling black-box mutational fuzzing. In: ACM CCS. pp. 511–522 (2013)
28. Wu, W.C., Nongpoh, B., Nour, M., Marcozzi, M., Bardin, S., Hauser, C.: Fine-grained coverage-based fuzzing. ACM Trans. Softw. Eng. Methodol. **33**(5) (jun 2024). https://doi.org/10.1145/3587158, https://doi.org/10.1145/3587158

LLM-Based Mutant Selection for Improving AFL++ Coverage

Dongeui Ko[0009-0007-3613-9775], Ilhwan Ji[0009-0003-9027-1305] and Jung Taek Seo* [0000-0003-0971-8548]

Gachon University, Seongnam-daero 1342, Seongnam-si, Republic of Korea
{eastmean, ilhwan1013, seojt}@gachon.ac.kr

Abstract. Existing AFL++ QEMU-based fuzzing for closed-source binaries suffers from high instrumentation overhead and slow execution speed, and because it executes generated mutants indiscriminately, coverage expansion and vulnerability discovery are delayed. Meanwhile, approaches that use large language models (LLMs) alone for fuzzing face the problems of excessive token consumption and difficulty in incorporating real-time feedback. To address these limitations, this paper proposes an LLM-based mutant scoring model in which the LLM scores mutants and determines their execution priority. The proposed model consists of an LLM-based mutant selector, plateau monitor, selector daemon, and ranking backend, and reduces unnecessary overhead and token consumption by invoking the LLM only when fuzzing enters a stagnation (plateau) state. In five independent 4-hour fuzzing for five times campaigns on four 1-Day vulnerabilities in commercial software, the proposed model achieved mean final edge coverage of 18.98%, 9.38%, 34.99%, and 35.25%, respectively, outperforming MOpt, QASan, and Fuzz4All, while consuming on average 95.49% fewer tokens than Fuzz4All, the LLM-based baseline. This shows that limiting the role of the LLM to mutant prioritization rather than input generation is effective in terms of balancing fuzzing efficiency and cost.

Keywords: Fuzzing, Mutant, Large-Language-Model

1 Introduction

Fuzzing is a dynamic analysis technique that discovers vulnerabilities by automatically generating and executing malformed or unexpected inputs to trigger abnormal program behavior. A fuzzer automates this process by repeatedly generating inputs, executing the target, collecting feedback, and saving crashes [1]. Among existing fuzzers, AFL++ is a representative coverage-guided framework that supports both open-source and closed-source targets [2]. For closed-source software, AFL++ commonly relies on QEMU mode because direct coverage instrumentation is difficult [3,4]. However, QEMU-based fuzzing introduces substantial runtime overhead and relatively low execution speed due to dynamic binary instrumentation. In addition, because generated mutants are executed without explicit prioritization, low-value inputs may be consumed

before those more likely to trigger new edges, reducing both coverage growth efficiency and vulnerability discovery speed.

To address these limitations, recent studies have increasingly explored the use of large language models (LLMs) for program and vulnerability analysis. However, using an LLM as a standalone fuzzing engine has several drawbacks: it incurs high token costs during mutation, input generation, and feedback handling; it tends to shift attention across regions rather than persistently exploring a promising fault-prone area; and it is not well suited to incorporating fuzzing feedback in real time.

This paper proposes an approach in which an LLM scores mutants and guides their execution priority during fuzzing. The framework consists of an LLM-based selector mutator, a plateau monitor, a selector daemon, and a ranking backend. Mutants generated by AFL++ are ranked using either LLM-based or heuristic scores and then fed back into the fuzzing loop. The plateau monitor detects stagnation, the selector daemon forwards re-ranking requests when needed, and the ranking backend invokes the LLM to recompute mutant priorities and return the result. In 4-hour fuzzing for five times experiments on four 1-day vulnerabilities from commercial software targets, the proposed method achieved the highest final coverage in all cases, reaching 18.98%, 9.38%, 34.99%, and 35.25%, while reducing token usage by an average of 95.49% compared with the LLM-based baseline. The main contributions of this paper are as follows.

- By proposing a scheme in which the AFL++ fuzzer invokes the model only when it enters a plateau state, unnecessary overhead and LLM token consumption are reduced.
- In 4-hour fuzzing for five times experiments on four 1-Day vulnerabilities, the proposed approach achieved higher coverage than MOpt [5], QASan [6], and Fuzz4All [7], while consuming 95.49% fewer tokens than Fuzz4All.

The remainder of this paper is organized as follows. Section 2 analyzes related work and its limitations. Section 3 describes the architecture of the proposed LLM mutant selector. Section 4 compares the proposed model with related approaches through fuzzing experiments on 1-Day vulnerabilities. Finally, Section 5 presents the conclusion and future research directions.

2 Related Works

In AFL++, fuzzing Linux closed-source binaries is commonly performed in QEMU mode [4]. Representative studies related to this setting include MOpt [5] and QASan [6], while Fuzz4All [7] represents an LLM-based fuzzing approach. These studies focus on different goals, namely mutation scheduling, shadow-memory-based error detection, and vulnerability discovery driven primarily by AI.

QEMU mode enables coverage-guided fuzzing in environments where only binaries are available, and source code is not accessible. It executes the fuzzing target through user-mode emulation and dynamically collects instrumentation data at runtime, which is then fed back into AFL++ as coverage information. MOpt improves fuzzing efficiency through an adaptive mutation scheduler. Rather than using all mutation operators uniformly, it applies Particle Swarm Optimization (PSO) to dynamically adjust the

selection probability of each operator so that operators more likely to produce interesting inputs are chosen more frequently. QASan is a sanitizer designed for QEMU-based binary fuzzing with a focus on detecting memory errors. Its design extends QEMU to instrument memory accesses and maintain shadow memory inside the emulator, while a runtime library hooks allocation and deallocation functions to apply redzones and quarantine. This allows memory-related bugs to be detected more effectively in closed-source targets. Fuzz4All, in contrast, is a generative fuzzer in which the LLM directly produces inputs. It automatically analyzes the user-provided target, generates test cases, and produces LLM-optimized prompts that are used in subsequent fuzzing iterations. In this sense, the LLM is responsible for most of the fuzzing workflow.

Despite their strengths, these approaches still have limitations. QEMU mode is suitable for closed-source targets, but it incurs substantial runtime overhead and focuses mainly on coverage collection and execution tracing, without prioritizing mutants in advance. MOpt only adjusts the probability distribution of low-level mutation operators and does not consider input structure or contextual consistency when prioritizing individual mutants. QASan uses the same mutation strategy as QEMU mode and focuses on memory bug detection, so its overall fuzzing efficiency is not fundamentally different from standard QEMU-based fuzzing. Fuzz4All relies on the LLM for core fuzzing functions, resulting in high overhead and substantial token consumption.

Accordingly, the goal of this paper is to address both the limited coverage expansion achieved relative to the high overhead of existing fuzzing approaches and the excessive token cost of LLM-based fuzzing.

3 Proposed LLM-Based Mutant Selection

This section presents the architecture of an LLM-based mutant selector for effective coverage expansion in closed-source software targets.

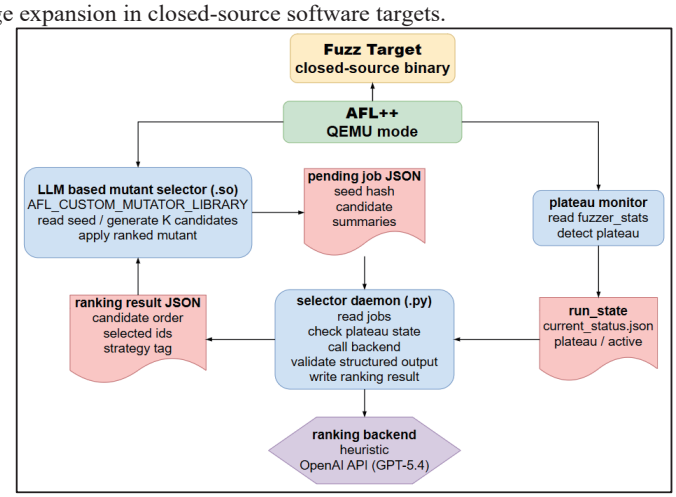


Fig. 1. Overview of the Proposed Model Architecture

As shown in Fig. 1, the proposed model consists of four complementary components: the LLM-based mutant selector, selector daemon, plateau monitor, and ranking backend. The LLM-based mutant selector applies either LLM-based or heuristic scores to AFL++ mutant candidates. The selector daemon forwards re-ranking requests to the ranking backend and shares the results. The plateau monitor tracks the fuzzing state and determines whether the process has entered a stagnation phase. When scoring is needed, the ranking backend invokes the LLM, ranks the candidates, and returns the result. To integrate this mechanism reliably without disrupting the fuzzing workflow, communication between the fuzzer and the LLM is handled through a structured JSON format.

3.1 LLM-Based Mutant Selector

The LLM-Based Mutant Selector is a library that operates inside AFL++. It generates multiple mutant candidates based on the current test case and, when the fuzzer enters a plateau state, forwards those candidates to the Ranking backend and requests re-ranking. Each candidate is assigned metadata including family, preview, len, emit_count, success_count, and score, and the candidates are executed according to either the ranking returned by the LLM or an internal heuristic ranking. If no valid LLM ranking is returned, or if the confidence of the response is low, the internal heuristic ranking is applied instead. The resulting data are delivered to the selector daemon in JSON format and are then reflected in subsequent fuzzing.

3.2 Selector Daemon

The selector daemon is a re-ranking relay module that operates independently outside AFL++. Directly calling the LLM inside the fuzzer or re-ranking all candidates at every mutation stage would incur high cost and overhead, which could degrade the performance of the entire fuzzing process. To avoid this, this study separates the scoring function into an independent external background process. The selector daemon requests re-ranking from the ranking backend based on the candidate information and input structure metadata generated by the LLM-based mutant selector and stores the result in a form that can be read again by the fuzzer. In the proposed model, the selector daemon is designed to invoke the ranking backend. It also records and shares separate state and job information in JSON format so that fuzzing can continue from the previous state even if the fuzzing process is interrupted. As a result, the selector daemon acts as an asynchronous external module that does not directly intervene in the internal flow of the fuzzer but invokes the ranking backend only when necessary.

3.3 Plateau Monitor

The plateau monitor is a module that monitors the fuzzing state of AFL++ and determines whether new edge exploration is still active or whether the process has entered a stagnation region. This module is responsible for informing the selector daemon when mutant candidates should be re-ranked. It reads the fuzzer state values and execution statistics, determines whether the fuzzing process is in a plateau state, and records the result in a state file so that the selector daemon can determine whether re-ranking is necessary. In other words, it identifies the point at which re-ranking of mutant

candidates are required and provides the corresponding state information to the selector daemon. Through this mechanism, the default mutation flow is maintained while fuzzing is actively exploring new edges, and the ranking backend is invoked only when the fuzzing process enters a plateau region where exploration efficiency has decreased.

3.4 Ranking Backend

The ranking backend is a module that enables the LLM to evaluate the set of mutant candidates delivered by the selector daemon and returns the re-ranking result in a form that can be reflected into the fuzzer. When the AFL++ fuzzer enters a plateau state, the mutator generates a JSON job file containing candidate information, and the Selector Daemon reads this file and forwards it to the ranking backend. The LLM then determines the ranking based on the metadata assigned to each mutant candidate so that candidates with a higher likelihood of generating new corpus entries are executed first. The resulting ranking is stored in JSON format, passed through the selector daemon to the LLM-based mutant selector, and then reflected in the subsequent fuzzing process.

The ranking backend is invoked only when a re-ranking request is triggered. Under normal conditions, candidates are first ordered according to internal heuristic scores, and the LLM is called for re-ranking only when a plateau state is detected. In addition, each time a candidate belonging to the same family is executed, its emit_count is increased, and if one of those candidates contributes to corpus growth, its success_count is also increased. This execution history is reflected in the family score, so families that are more effective at generating new corpus entries receive higher scores. Even when the superiority between candidates is not clearly distinguishable, the ranking result is still returned, but with a lower confidence value, allowing the system to fall back to the heuristic order when necessary.

4 Experiments and Validation

4.1 Experimental Setup

In this paper, fuzzing experiments were conducted to evaluate the performance of the AFL++-based mutant selector using four 1-day vulnerabilities from four commercial software targets. The hardware environment consisted of a Ryzen5 5600 CPU running Windows 11 Education 25H2, and the operating system was Ubuntu 24.04.3 LTS on WSL [8]. The fuzzer was built with AFL++ version 4.36a, and all targets were compiled and configured as closed-source software. The LLM-Based Mutant Selector, MOpt, and QASan were evaluated through five independent four-hour fuzzing campaigns in a parallel fuzzing environment with one master node and two slave nodes. Fuzz4All does not support parallel fuzzing, so it was evaluated on a single node, and the LLM was unified to ChatGPT 5.4 [9], the same model used in the proposed approach. The main parameters and descriptions of the four components in the proposed model are summarized in Table 1.

Table 1. Proposed model parameters and value

Module	Parameter	Value
LLM-Based Mutant Selector	family	Mutation strategy/type
	preview	Header preview metadata
	len	Length of the candidate
	emit_count	Executed same-family candidates
	success_count	Corpus-added same-family candidates
	score	Family history-based heuristic score
Selector Daemon	pending_job	Monitored in the runtime for reranking
	result_json	Returned JSON ranking result
Plateau Monitor	plateau_active	Stagnation-state indicator
	seconds_since_last_path	Time since last new edge
Ranking backend	model	ChatGPT 5.4 codex

4.2 Method for measuring coverage

In the experiments that evaluate fuzzing performance by comparing the LLM-based mutant selector applied to AFL++ with the baseline models, coverage is measured using edge coverage (1).

$$Edge\ Coverage(\%) = \frac{Number\ of\ Executed\ Edges}{Shared\ Memory\ bit\ map\ size(64KB)} \times 100 \quad (1)$$

In the proposed model, coverage is represented as the percentage of how many new edges were executed during fuzzing iterations within the entire 64 KB shared bitmap memory used by AFL++

4.3 Prompt composition

The prompt consists of a system prompt and a user prompt. The system prompt is a fixed instruction given to the LLM, requesting it to comprehensively evaluate the mutant candidates provided through the user prompt and determine their ranking. The user prompt converts the mutant candidates that require ranking into JSON format and sends them to the LLM for scoring, and the result is also returned in JSON format.

Table 2. System and User prompt for reranking

Prompt	Description
System prompt	You rank AFL++ mutant candidates for a closed-source binary. Prefer candidates that are more likely to unlock new edges while keeping the input close to a plausible valid structure. When the metadata suggests XML Schema / XSD input, prefer candidates that keep the file close to a plausible schema and diversify the declaration shapes. Candidate metadata may include historical emit_count, success_count, and score from the local fuzzing run. Treat these as noisy priors that can help break ties, not as hard rules. Prefer rankings that preserve structural diversity. Return JSON only. Do not explain your reasoning outside the schema.
User prompt	You are ranking mutant candidates for a closed-source AFL++ QEMU fuzzing run. Return only JSON matching the provided schema.

4.4 Validation and Comparative Evaluation of 1-Day Vulnerability Fuzzing

In this section, we evaluate the proposed LLM-based mutant selector through five independent four-hour fuzzing campaigns and compare its performance with that of the baseline models. For four 1-day vulnerabilities from four software targets, we compare mean final edge coverage, average execution speed, and crash detection results to assess the effectiveness of the proposed model.

4.5 Case 1: CVE-2017-12997

CVE-2017-12997 [10] is a 1-day vulnerability in TCPdump version 34cec72 where the LLDP parser enters an infinite loop while processing an invalid TLV value, eventually leading to an out-of-bounds read. Fig. 2 shows edge coverage across five times for four-hour fuzzing runs. Red denotes the proposed model, green MOpt, yellow QASan, and black Fuzz4All.

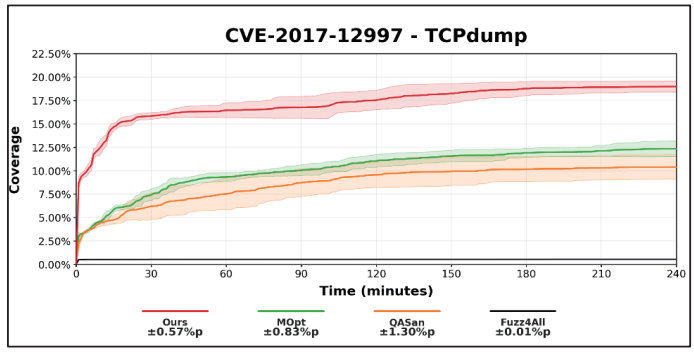


Fig. 2. Edge Coverage Graph During Fuzzing of CVE-2017-12997

As shown in Fig. 2, the proposed model rapidly increased mean edge coverage during the early stage of fuzzing and maintained the highest coverage throughout the four-hour campaign. MOpt and QASan showed similar coverage growth trends, whereas Fuzz4All exhibited almost no meaningful improvement. At the end of the five times for four-hour runs, the proposed model achieved a mean final edge coverage of 18.98 ± 0.57 %, compared with 12.34 ± 0.83 % for MOpt, 10.39 ± 1.30 % for QASan, and 0.54 ± 0.01 % for Fuzz4All. These results indicate that the proposed model consistently maintained a clear coverage advantage across the repeated runs.

Table 3. CVE-2017-12997 Average Execution Speed and Crash Discovery Time

Model	exec/sec	crash(mm:ss) / success
MOpt[5]	480.01 ± 44.70	N/A / (0/5)
QASan[6]	174.56 ± 23.40	12:03 ± 01:02 / (4/5)
Fuzz4All[7]	0.10 ± 0.01	N/A / (0/5)
Ours	201.64 ± 43.64	N/A / (0/5)

As shown in Table 3, the proposed method achieved an average execution speed of 201.64 ± 43.64 exec/sec, which was 278.37 exec/sec lower than MOpt, but 27.08 exec/sec and 201.54 exec/sec higher than QASan and Fuzz4All, respectively. In terms of vulnerability triggering, success denotes the number of runs in which the vulnerability was successfully

triggered out of five independent fuzzing campaigns. QASan was the only method that successfully triggered CVE-2017-12997, doing so in four of five runs with a mean trigger time of 12:03 ± 01:02. By contrast, MOpt, Fuzz4All, and the proposed method failed to trigger the vulnerability in all five runs. Nevertheless, the proposed method achieved the highest mean final edge coverage.

4.6 Case 2: CVE-2024-8176

CVE-2024-8176 [11] is a 1-day vulnerability in libexpat version 2.7.0. During the processing of recursive entity expansion in an XML document, infinite recursive calls occur, eventually exhausting the stack and leading to a stack overflow. Fig. 3 shows the variation in edge coverage across five times for four-hour fuzzing for this vulnerability.

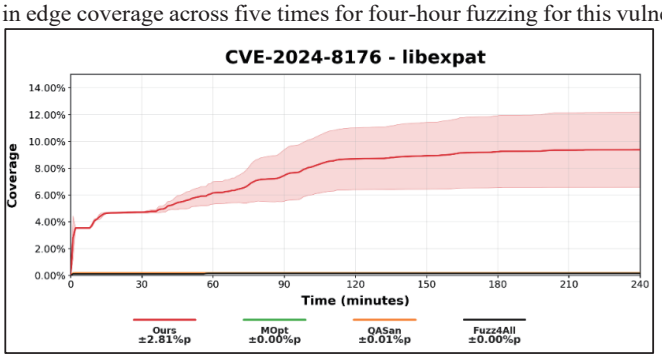


Fig. 3. Edge Coverage Graph During Fuzzing of CVE-2024-8176

As shown in Fig. 3, the proposed model exhibited a relatively rapid increase in mean edge coverage from the early stage of fuzzing and showed a clearly larger gain than the other comparison models. In contrast, MOpt, QASan, and Fuzz4All showed almost no meaningful coverage growth throughout the campaign. At the end of the five times for four-hour runs, the proposed model achieved the highest mean final edge coverage at 9.38 ± 2.81 %, whereas MOpt, QASan, and Fuzz4All recorded 0.17 ± 0.00 %, 0.16 ± 0.01 %, and 0.13 ± 0.00 %, respectively.

Table 4. CVE-2024-8176 Average Execution Speed and Crash Discovery Time

Model	exec/sec	crash(mm:ss) / success
MOpt[5]	932.00 ± 23.77	07:28 ± 00:10 / (4/5)
QASan[6]	829.76 ± 625.96	N/A / (0/5)
Fuzz4All[7]	0.53 ± 0.15	03:01 ± 00:05 / (5/5)
Ours	392.61 ± 173.44	40:12 ± 01:05 / (5/5)

As shown in Table 4, the proposed method achieved an average execution speed of 392.61 ± 173.44 exec/sec, which was 539.39 exec/sec lower than MOpt and 437.15 exec/sec lower than QASan, but 392.08 exec/sec higher than Fuzz4All. In terms of vulnerability triggering, both Fuzz4All and the proposed method succeeded in all five runs, while MOpt succeeded in four of five runs and QASan failed in all five runs. The proposed method showed a mean trigger time of 40:12 ± 01:05, which was 37:11 slower than Fuzz4All and 32:44 slower than MOpt. Despite this slower trigger speed, the proposed method achieved the highest mean edge coverage..

4.7 Case 3: CVE-2025-3154

CVE-2025-3154 [12] is an out-of-bounds array write vulnerability in Xpdf 4.05 and earlier. It is triggered when an invalid VerticesPerRow value is processed in a PDF shading dictionary. Fig. 4 shows the variation in edge coverage across five times for four-hour fuzzing runs for this vulnerability.

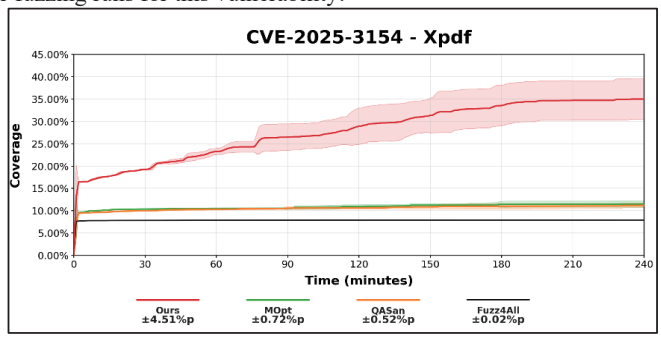


Fig. 4. Edge Coverage Graph During Fuzzing of CVE-2025-3154

As shown in Fig. 4, the proposed method rapidly increased mean edge coverage during the early stage of fuzzing and continued to expand coverage in a stepwise manner throughout the four-hour campaign. MOpt and QASan showed similar coverage growth, whereas Fuzz4All reached approximately 7% coverage early in the campaign but exhibited little subsequent improvement. Across the five repeated four-hour runs, the proposed method achieved the highest mean final edge coverage of 34.99 ± 4.51 %, compared with 11.47 ± 0.72 % for MOpt, 11.16 ± 0.52 % for QASan, and 7.89 ± 0.02 % for Fuzz4All. These results indicate that the proposed method maintained a substantial coverage advantage across the repeated runs.

Table 5. CVE-2025-3154 Average Execution Speed and Crash Discovery Time

Model	exec/sec	crash(min:ss) / success
MOpt[5]	250.28 ± 21.21	N/A / (0/5)
QASan[6]	119.85 ± 64.01	N/A / (0/5)
Fuzz4All[7]	0.43 ± 0.01	N/A / (0/5)
Ours	95.16 ± 14.40	N/A / (0/5)

As shown in Table 5, the proposed method achieved an average execution speed of 95.16 ± 14.40 exec/sec, which was 155.12 exec/sec lower than MOpt and 24.69 exec/sec lower than QASan, but 94.73 exec/sec higher than Fuzz4All. Across five independent four-hour fuzzing runs, none of the four methods successfully triggered CVE-2025-3154. Nevertheless, the proposed method achieved the highest mean final edge coverage of 34.99 ± 4.51%, outperforming MOpt, QASan, and Fuzz4All. These results indicate that, despite its lower execution speed than MOpt and QASan, the proposed method was more effective at broad path exploration for the Xpdf target.

4.8 Case 4: CVE-2025-49795

CVE-2025-49795 [13] is a vulnerability found in libxml2 version 6a6a46f0, in which a NULL pointer dereference occurs while processing an XPath expression in XML.

Fig. 5 shows the variation in edge coverage across five four-hour fuzzing runs for this vulnerability.

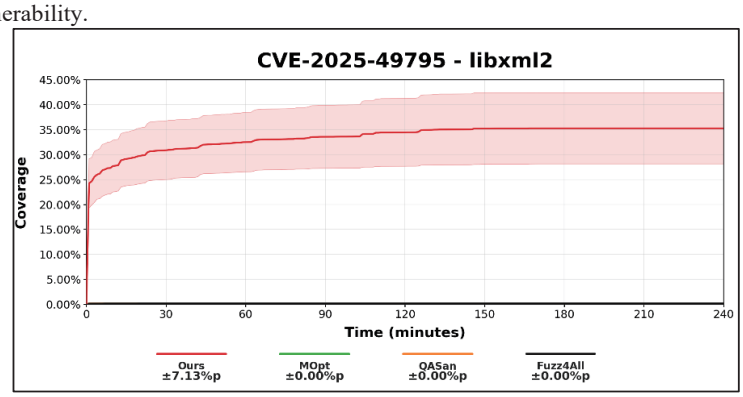


Fig. 5. Edge Coverage Graph During Fuzzing of CVE-2025-49795

As shown in Fig. 5, the proposed method rapidly increased mean edge coverage in the early stage of fuzzing and continued to expand coverage throughout the four-hour campaign, maintaining a substantial gap over the baseline methods. In contrast, MOpt, QASan, and Fuzz4All showed negligible coverage growth across the five repeated runs. The proposed method achieved a mean final edge coverage of 35.25 ± 7.13 %, whereas MOpt and QASan both reached 0.19 ± 0.02 %, and Fuzz4All reached 0.16 ± 0.00 %. Although the proposed method showed a larger standard deviation than the baseline methods, it consistently maintained a clear coverage advantage across the repeated runs.

Table 6. CVE-2025-49795 Average Execution Speed and Crash Discovery Time

Model	exec/sec	crash(min:ss) / success
MOpt[5]	2128.52 ± 103.63	66:32 ± 21:36 / (5/5)
QASan[6]	1420.49 ± 461.01	20:15 ± 01:20 / (5/5)
Fuzz4All[7]	0.20 ± 0.11	N/A / (0/5)
Ours	1121.26 ± 12.42	05:49 ± 02:05 / (5/5)

As shown in Table 6, the proposed method achieved an average execution speed of 1121.26 ± 12.42 exec/sec, which was 1007.26 exec/sec lower than MOpt and 299.23 exec/sec lower than QASan, but 1121.06 exec/sec higher than Fuzz4All. In terms of crash detection, the proposed method successfully triggered the vulnerability in all five runs and recorded the earliest mean trigger time of 05:49 ± 02:05. QASan and MOpt also triggered the vulnerability in all five runs, but their mean trigger times were 14:26 and 60:43 later than that of the proposed method, respectively, whereas Fuzz4All failed to trigger the vulnerability in any run. In the CVE-2025-49795 fuzzing experiment, the proposed method showed the best overall performance by achieving both the highest edge coverage and the fastest crash detection.

4.9 Comparison of LLM token cost

The comparison of token usage between Fuzz4All, which relies on an LLM, and the proposed LLM-based mutant selector model in this experiment is as follows.

Table 7. Average LLM token usage

CVE	Ours	Fuzz4All[7]
CVE-2017-12997[10]	53,114	2,606,296
CVE-2024-8176[11]	193,257	1,862,744
CVE-2025-3154[12]	26,654	1,651,718
CVE-2025-49795[13]	77,238	1,930,618

As shown in Table 7, the proposed model consistently consumed fewer tokens than Fuzz4All across all four targets. The token reductions were 97.96% for CVE-2017-12997, 89.63% for CVE-2024-8176, 98.39% for CVE-2025-3154, and 96.00% for CVE-2025-49795. These results indicate that invoking the LLM only when mutant selection is required can substantially reduce token consumption while still supporting effective fuzzing.

5 Discussion

Across five independent four-hour fuzzing campaigns on four 1-day vulnerability targets, the proposed LLM-based mutant selector achieved the highest mean final edge coverage in all cases. Although the proposed method was generally slower than MOpt and QASan due to the overhead of selector-daemon communication and LLM-based reranking, it still outperformed Fuzz4All in both execution efficiency and coverage while using 95.49% fewer LLM tokens on average. In terms of crash detection, it triggered crashes in two of the four targets, matching the target-level success rate of MOpt and QASan and outperforming Fuzz4All. In particular, it detected crashes earlier than all comparison methods on CVE-2025-49795, whereas some baselines were faster on CVE-2024-8176. Overall, these results indicate that the proposed method favors broader edge exploration and is effective for closed-source fuzzing despite the additional execution overhead.

6 Conclusion and future works

This paper proposes an LLM-based mutant selection model that prioritizes and dynamically reorders the execution of AFL++-generated mutants during fuzzing. The proposed selector is integrated into the fuzzing loop and, when a plateau state is detected, invokes a selector daemon to obtain LLM-based scoring and reflect the result in the subsequent execution order. This study evaluated the proposed design on four 1-day vulnerability targets through five independent four-hour fuzzing campaigns and compared it with MOpt, QASan, and Fuzz4All. The results show that the proposed model achieved the highest mean final edge coverage on all targets, demonstrating that prioritizing mutants that are more likely to exercise new edges is more effective for coverage expansion than simply executing a large number of mutants. Although the proposed method was generally slower than MOpt and QASan due to the overhead of daemon communication and LLM-based reranking, it still outperformed Fuzz4All in both execution efficiency and coverage while using 95.49% fewer LLM tokens on average.
In terms of crash detection, the proposed method triggered crashes on two of the four targets, matching the target-level detection success rate of MOpt and QASan and

outperforming Fuzz4All. In particular, it detected crashes earlier than all comparison methods on CVE-2025-49795, whereas some baselines detected crashes earlier on CVE-2024-8176. Overall, the proposed LLM-based mutant selector expanded coverage most effectively in the closed-source fuzzing setting while requiring substantially fewer tokens than an existing LLM-based fuzzer.

As future work, we plan to use the LLM API [14] to integrate the selector more tightly with the fuzzer and enable immediate feedback-driven scoring. Through this, we aim to reduce overhead to a level closer to LLVM clang [15] and improve the system so that 0-day vulnerabilities can be discovered in a shorter amount of time.

Acknowledgements. This work was partly supported by the Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00400955, Technology Development of Threat Model/XAI-based Network Abnormality Detection, Response and Cyber Threat Prediction, 50%) and Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT)(RS-2026-25585348, Graduate School of Security Convergence(AI)(Gachon University), 50%)

References

1. Fioraldi, A., Mantovani, A., Maier, D., & Balzarotti, D. (2023). Dissecting american fuzzy lop: a fuzzbench evaluation. ACM transactions on software engineering and methodology, 32(2), 1-26.
2. Fioraldi, A., Maier, D., Eißfeldt, H., & Heuse, M. (2020). AFL++: Combining incremental steps of fuzzing research. In 14th USENIX workshop on offensive technologies (WOOT 20).
3. Bellard, F. (2005, April). QEMU, a fast and portable dynamic translator. In Usenix ATC, Freenix Track (pp. 41-46).
4. AFL++ QEMU mode, https://aflplus.plus/docs/fuzzing_binary-only_targets/#qemu-mode, [accessed: 2026.06.08]
5. Lyu, C., Ji, S., Zhang, C., Li, Y., Lee, W. H., Song, Y., & Beyah, R. (2019). MOPT: Optimized mutation scheduling for fuzzers. In 28th USENIX security symposium (USENIX security 19) (pp. 1949-1966).
6. Fioraldi, A., D'Elia, D. C., & Querzoni, L. (2020, September). Fuzzing binaries for memory safety errors with QASan. In 2020 IEEE Secure Development (SecDev) (pp. 23-30). IEEE.
7. Xia, C. S., Paltenghi, M., Le Tian, J., Pradel, M., & Zhang, L. (2024, April). Fuzz4all: Universal fuzzing with large language models. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (pp. 1-13).
8. Barnes, H. (2021). Pro Windows Subsystem for Linux (WSL) (pp. 1-321). Apress.
9. An, J., Ding, W., & Lin, C. (2023). Chatgpt. tackle the growing carbon footprint of generative AI, 615, 586.
10. CVE-2017-12997. https://nvd.nist.gov/vuln/detail/cve-2017-12997
11. CVE-2024-8176. https://nvd.nist.gov/vuln/detail/cve-2024-8176
12. CVE-2025-3154. https://nvd.nist.gov/vuln/detail/cve-2025-3154
13. CVE-2025-49795. https://nvd.nist.gov/vuln/detail/cve-2025-49795
14. Fei, X., Zheng, X., & Feng, H. (2025). Mcp-zero: Active tool discovery for autonomous llm agents. arXiv preprint arXiv:2506.01056.
15. Lattner, C. (2008, May). LLVM and Clang: Next generation compiler technology. In The BSD conference (Vol. 5, pp. 1-20).

MIKI-Fuzz: End-to-End Black-Box Fuzzing of Introspection-Disabled GraphQL APIs

Miri Park[0009-0000-0151-0449], Ilhwan Ji[0009-0003-9027-1305] and Jung Taek Seo*[0000-0003-0971-8548]

Gachon University, Seongnam-daero 1342, Seongnam-si, Republic of Korea
{qkralfl02, ilhwan1013, seojt}@gachon.ac.kr

Abstract GraphQL's flexible query model and nested data dependencies expose APIs to context-dependent vulnerabilities, such as access-control bypass, denial of service (DoS), and injection, that conventional testing tools struggle to uncover. Existing GraphQL fuzzers require schema information to build valid queries, and many rely on introspection, which is often disabled in production. Even when a partial schema is recovered from error responses, detection fails unless that schema drives query generation and vulnerability detection. This paper proposes **MIKI-Fuzz**, a response-based black-box fuzzer for introspection-disabled GraphQL APIs. **MIKI-Fuzz** recovers a partial schema from validation errors and successful responses and connects it to query generation, structural mutation, and vulnerability detection. During fuzzing it feeds newly observed responses back into recovery, expanding the schema and reaching operations that static recovery alone would miss. On DVGA and poc-graphql, **MIKI-Fuzz** confirmed detection signals for all public vulnerability scenarios under disabled introspection. Against a locally deployed Saleor 3.23.12 with all published CVE patches applied, it confirmed three previously undisclosed zero-day vulnerabilities, all reported privately to the vendor.

Keywords: GraphQL Security, Black-box Fuzzing, Vulnerability Detection

1 Introduction

Modern software increasingly consists of independent services that exchange data through APIs, and GraphQL is a widely adopted choice for building them. Although REST, SOAP, and gRPC remain common, GraphQL adoption is growing because it lets clients express complex data relationships in a single request and retrieve only the required fields [1,2]. Unlike those APIs, the client query itself determines the request scope and execution path. This flexibility improves data retrieval but complicates security testing, because server behavior depends on how each query is composed. Malformed or deeply nested queries can cause denial of service (DoS), verbose validation errors can leak clues to the hidden schema, and inconsistent authorization at the object or field level can lead to access-control bypass or information disclosure.

To address these risks, a range of GraphQL security testing and fuzzing approaches has emerged. Tools such as EvoMaster [3] and GraphQLer [4] generate requests from the schema, recent work applies large language models to query generation [5], and

Clairvoyance [6] instead reconstructs a hidden schema from server error responses. Most obtain the schema through introspection, which production servers often disable to hide internal structure, leaving those tools unable to proceed. The few that recover part of the schema from error responses stop there. That recovered schema is rarely carried through to query generation and detection, and connecting the two is difficult, because a partial schema yields only partial queries and each response must refine both the schema and the next request. Closing this gap between recovering a schema and using it to detect vulnerabilities is the problem this paper addresses.

This paper proposes **MIKI-Fuzz**, a response-based black-box fuzzer for introspection-disabled GraphQL APIs. **MIKI-Fuzz** does not presuppose a complete schema. It connects the partial schema recovered from validation errors and successful responses to query generation, structural mutation, and vulnerability detection. On three GraphQL servers, **MIKI-Fuzz** recovers more schema than introspection-dependent tools, and on two of them a usable schema where those tools recover nothing at all. On a fully patched deployment of the Saleor e-commerce platform it discovered three previously undisclosed zero-day vulnerabilities. The main contributions are as follows:

- **Response-based schema recovery under disabled introspection: MIKI-Fuzz** reconstructs a partial schema from validation errors and successful responses alone, without introspection, a ground-truth schema, or source-level access. Newly observed responses are fed back to expand the recovered schema during fuzzing.
- **An end-to-end recovery-to-detection pipeline: MIKI-Fuzz** connects the recovered schema to query generation, AST-based structural mutation, and vulnerability detection, so that fuzzing proceeds even from a partial schema. This connection, absent in prior tools, is what turns recovery into detection.
- **Evaluation and zero-day discovery: MIKI-Fuzz** is evaluated on DVGA, pocgraphql, and Saleor under disabled introspection, confirming three previously undisclosed zero-day vulnerabilities in a fully patched Saleor deployment.

2 Background & Related Work

This section covers GraphQL APIs, their vulnerabilities, and prior work.

2.1 GraphQL Fundamentals

GraphQL is an API query language, open-sourced by Facebook in 2015 [7], that exposes all operations through a single endpoint. A GraphQL API is organized around a schema, and clients request the exact fields they need, which reduces over-fetching and under-fetching. The schema is built from scalar types such as Int, String, and Boolean, and from object types whose fields may be scalars, objects, or lists, so object references make the schema an interconnected graph. Clients read data through Query operations and modify it through Mutation operations. A specialized introspection query returns type and operation information, but production servers often disable it to limit exposure. Before execution, a request is validated against the schema and resolved field by field, so non-existent fields or invalid arguments produce validation errors.

2.2 GraphQL Vulnerabilities

GraphQL's flexibility gives clients fine-grained control over each query, and that freedom opens new ways to attack it. Some resemble familiar web problems such as injection or broken access control. Others arise from GraphQL itself, where a single endpoint runs whatever the client sends. This paper sorts them into four groups. Transport-layer misconfiguration, such as a permissive CORS header or a Subscription over plain HTTP, is not a query-level class but is evaluated with them in Section 4.3.

- **Injection**: If the API passes user input straight to a database or system command without checking it, an attacker can hide malicious code in that input. This is how SQL injection, OS command injection, and cross-site scripting (XSS) happen, each able to leak or alter protected data.
- **Information disclosure**: Sometimes the server reveals too much. Error messages with stack traces or file paths, or fields that return more than they should, let an attacker map out the system, often as the starting point for a sharper attack.
- **Access-control bypass**: This happens when the API does not properly check who may see what. An attacker can change an object's ID to reach data belonging to someone else, an insecure direct object reference (IDOR), or bundle many actions into one request to slip past checks that run one operation at a time.
- **Denial of service**: A single query can force the server into heavy work. Deeply nested fields, repeated or aliased fields, and batched queries all pile on cost and can slow the server or take it offline.

2.3 Related Works

Schema-driven testing. Most API testing assumes a machine-readable description of the interface. In the REST domain, RESTler [8] infers dependencies between requests from prior responses and builds stateful test sequences, but it relies on an OpenAPI specification. For GraphQL, EvoMaster [3] applies evolutionary and random testing, and GraphQLer [4] analyzes the schema to infer producer-consumer dependencies between queries and mutations, while recent work adds large language model reasoning to query generation [5]. All of these drive test generation from a specification or an introspected schema, so their testing cannot start once introspection is disabled.
Schema recovery. Closer to our setting, Clairvoyance [6] reconstructs type and field names from server error messages without introspection. However, it stops at the recovered schema and does not use it for testing, leaving recovery disconnected from detection, and it depends on the server suggesting the correct name. **MIKI-Fuzz** instead infers fields from the shape of error responses and expands them with word rules, recovering a schema even from terse-error servers that offer no suggestions.
Vulnerability-specific analysis. A separate line of work targets individual vulnerability classes. For denial of service, Cha et al. [9] analyze query cost statically, WENDIGO [10] finds resource-exhausting queries through reinforcement learning, and other work detects malicious queries with language models [11]. For access control, Barabanov et al. [12] process API specifications and Lambers et al. [13] apply taint analysis. Each

assumes a schema, a specification, or source-level access, none of which is available once introspection is disabled.

Across these lines, no GraphQL tester recovers the schema from server responses under disabled introspection and carries it through to detection. **MIKI-Fuzz** fills this gap, integrating response-based schema recovery, query generation, structural mutation, and detection into a single end-to-end black-box pipeline.

3 Proposed MIKI-Fuzz Framework

This section presents the **MIKI-Fuzz** methodology. The pipeline is shown in Fig. 1.

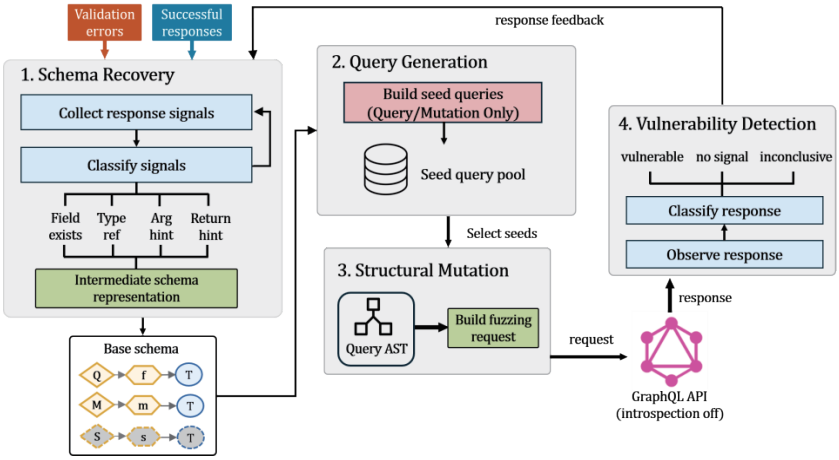


Fig. 1. Overview of **MIKI-Fuzz**

The framework proceeds in four stages. It recovers a partial schema from server responses, generates queries, applies structural mutation, and detects vulnerabilities. Responses observed during fuzzing are fed back to expand the schema.

3.1 Response-based Schema Recovery

The pipeline begins by recovering the schema, that is, which fields, arguments, and types the target API allows, from the server's responses alone, with no ground-truth schema, source code, or configuration file.

Recovery starts from a small fixed list of common GraphQL names, such as id, name, and user, reused for every target. **MIKI-Fuzz** then grows the list by watching how the server replies. A wrong field name sometimes draws a suggestion for the correct one, a malformed query leaks type and argument names in its error, and a successful query reveals real field names in its data. Each becomes a name to try next. **MIKI-Fuzz** also derives related names with simple word rules such as pluralization,

create/read/update/delete verb combination, and camelCase composition. A name enters the schema only after a later request confirms it, so guesses never stand alone.

Each candidate is checked from the root type by sending it and reading the server's reaction. If the server replies "*Cannot query field x on type Query*" for *query { x }*, the field does not exist. But if it complains about something that assumes the field is there, such as a missing argument, a demand to select subfields, a type mismatch, or a permission error, the field is real and is kept. Because this check happens before any resolver runs, **MIKI-Fuzz** never needs a working query to start, avoiding a chicken-and-egg problem. Once a field is confirmed to be an object, **MIKI-Fuzz** inserts wrong subfields to trigger more errors, and the suggestions or type errors that come back reveal the object's inner fields and type relationships. Servers that give no suggestions are handled with argument errors, subfield-selection demands, and type mismatches instead, though the less an error says, the less can be recovered.

The Subscription type is recovered the same way, by sending an invalid Subscription operation over HTTP and reading the validation errors, without ever running it. All recovered elements form an intermediate schema, each tagged with its source and confirmation status. Confirmed elements are used first in query generation, and unconfirmed ones remain candidates for later checks.

3.2 Schema-based Query Generation

The next step turns the recovered schema into seed queries. **MIKI-Fuzz** picks a root field from the Query or Mutation type and fills in its subfields based on the return type. A scalar is used as is, while an object or list is expanded until it reaches a scalar or a set depth limit. When a field needs arguments, **MIKI-Fuzz** prefers values already seen in earlier responses, and otherwise draws from enum definitions, input object structures, argument names, and finally the GraphQL scalar type. A query that runs without a validation error is saved to the seed set. One that triggers an error is sent back to recovery, where it may reveal something new.

Recovered Subscription operations are left out of seed generation. A Subscription runs over a WebSocket stream rather than an ordinary HTTP POST, so this HTTP-only fuzzer cannot execute one. They serve only to identify attack surface and for the transport-layer check in Section 3.4.

3.3 AST-based Structural Mutation

Seed queries are then reshaped into fuzzing requests. Instead of editing the query as raw text, **MIKI-Fuzz** edits its abstract syntax tree (AST), the parsed tree form of the query, so the document stays structurally valid while its fields, arguments, and subfields are rearranged. The mutations add aliases, duplicate fields, insert fragments, expand nesting within limits, add the non-introspection meta-field __typename, insert directives, and substitute boundary values for arguments. Each has a purpose. Aliases and duplicated fields make the same resolver run repeatedly, bounded nesting follows the recovered object relationships to push execution deeper, and boundary values stress input handling to surface validation errors and injections.

Before it is sent, a mutated query passes a basic syntax check. A request is dropped only if its structure is broken, never just for using something not yet in the recovered

schema. Since that schema may be incomplete, any syntactically valid request is sent and the server's response decides it. If the server accepts it, it joins the seed set. If a validation error comes back, that response returns to recovery as a fresh clue.

3.4 Response Handling and Vulnerability Detection

Every response feeds vulnerability determination, which relies only on externally observable signals such as status codes, GraphQL error messages, returned data, response time, and shifts in server availability. From successful responses, **MIKI-Fuzz** extracts object identifiers, scalar values, returned fields, and object counts, then reuses them when re-querying the same object or changing the user role, authentication context, or argument values. From error responses, it reads the GraphQL errors field with the HTTP status code. Validation errors refine later requests, while database errors, command-execution traces, stack traces, internal paths, and exposed sensitive fields are detection signals.

Access-control problems appear as response differences across permission levels. If a lower-privilege request returns the same object, sensitive fields, or a wider data range than it should, **MIKI-Fuzz** flags an access-control bypass. For state-changing requests, it re-queries the affected objects before and after execution to check whether limited resources such as stock, coupons, or refunds slip past their allowed range.

The remaining vulnerability types each have their own tell. For server-side request forgery (SSRF), **MIKI-Fuzz** injects a URL with a unique marker and confirms a server-side request through an external receiver's log, a reflected response, or repeated timeout-level delays. A deny-list bypass appears when ordinary requests are blocked but named or mutated operations still go through. Resource-consumption issues appear when nested, repeated, aliased, or batched queries push latency far above normal or leave the server unresponsive. This latency reflects the target’s cost handling, not the load the fuzzer places on the server. Aggregate load and post-fuzzing recovery are reported in Section 4.5. Transport-layer misconfiguration is judged from the HTTP response itself. Subscription exposure is recorded when the server validates an HTTP-sent Subscription operation against the Subscription type instead of rejecting it, which does not mean a subscription resolver actually runs. A cross-origin resource sharing (CORS) misconfiguration is recorded when the Access-Control-Allow-Origin header is a wildcard or echoes back the requesting origin.

Race conditions take a different test, running the same limited Mutation both sequentially and concurrently. A request with a hard limit, such as a single item in stock or a one-time voucher, is first sent sequentially to confirm the server enforces the limit, then fired concurrently. If the count stays within the limit in sequence but exceeds it in parallel, that is a race condition. This separates limits that were never enforced from limits that break only under concurrency.

Finally, each case is labeled vulnerable, no signal, or inconclusive. Vulnerable means the signals meet the criteria, no signal means nothing relevant appeared, and inconclusive means a suspicious signal was not confirmable from a single response. Every result is saved with its request, response, criteria, signal, and schema elements.

4 Evaluation

This section evaluates the schema recovery and vulnerability detection performance of **MIKI-Fuzz** according to the following research questions. **RQ1.** Can **MIKI-Fuzz** recover the schema of different GraphQL servers under introspection-disabled conditions? **RQ2.** Does the recovered schema lead to vulnerability detection?

4.1 Experimental Setup

Evaluated Application. This study locally deployed three GraphQL servers, DVGA, poc-graphql, and Saleor. DVGA and poc-graphql were used for schema recovery and for scoring against the ground-truth vulnerability list. Saleor was used to evaluate recovery on a production-grade server and to assess zero-day discovery.

- **DVGA.** DVGA (Damn Vulnerable GraphQL Application) [14] is an educational GraphQL API intentionally made vulnerable for GraphQL vulnerability testing.
- **poc-graphql.** poc-graphql [15] is an open-source GraphQL server implemented in graphql-java and includes publicly documented vulnerability scenarios.
- **Saleor.** Saleor [16] is an open-source e-commerce GraphQL server built on Python and Graphene. This study used version 3.23.12 with all published CVE patches.

Implementation and Configuration. **MIKI-Fuzz** was implemented in Python using HTTP POST requests, with introspection disabled. All targets ran locally on an Intel Core i7-14700K workstation with 64 GB RAM running Windows 11, so the Section 4.5 timings exclude network latency. Schema recovery ran five times per target–tool combination and was deterministic across runs (sample standard deviation 0), so Table 1 reports single values. The full pipeline, detection included, also ran five times per target for the Section 4.5 measurements. Table 2 reports one binary outcome per scenario, scored against the ground-truth vulnerability list.
Baseline Selection. Baselines were selected according to the evaluation criterion. For schema recovery, GraphQLer and Clairvoyance were selected as baselines because both can infer schemas from error responses under introspection-disabled conditions. For vulnerability detection, GraphQLer and EvoMaster were selected as baselines because they employ different schema acquisition strategies.
Evaluation Metrics. Schema recovery and vulnerability detection are reported separately. Schema recovery is measured across four dimensions, namely Field, Type, Edge, and Overall. Field, Type, and Edge are the proportions of field names, type names, and type–field relationships recovered, and Overall is their arithmetic mean. Vulnerability detection is a binary outcome per scenario, scored against the ground-truth list.

4.2 Schema Recovery (RQ1)

Schema recovery begins not from the ground-truth schema of a specific target but from the fixed initial candidate list of Section 3.1, used identically across all targets and built from common GraphQL vocabulary such as id, name, and user.

MIKI-Fuzz recovered the schema at 100% in all five runs on DVGA. On poc-graphql, **MIKI-Fuzz** achieved an Overall recovery rate of 61.7%. Under the same

conditions, GraphQLer and Clairvoyance obtained no recovery results (0.0%). On Saleor, Overall recovery was 13.3%. The rate is low because its production schema is far larger than those of the other two targets, so a fixed request budget covers a smaller fraction of it, yet the recovered portion still reached the operations in Section 4.3.

Table 1. Schema recovery rate (%) under introspection-disabled conditions.

Target	Tool	Field	Type	Edge	Overall
DVGA	**MIKI-Fuzz**	**100.0**	**100.0**	**100.0**	**100.0**
	GraphQLer	94.9	92.9	94.0	93.9
	Clairvoyance	61.5	57.1	58.0	58.9
poc-graphql	**MIKI-Fuzz**	**58.3**	**60.0**	**66.7**	**61.7**
	GraphQLer	0.0	0.0	0.0	0.0
	Clairvoyance	0.0	0.0	0.0	0.0
Saleor	**MIKI-Fuzz**	**22.5**	**8.9**	**8.5**	**13.3**
	GraphQLer	0.0	0.0	0.0	0.0
	Clairvoyance	0.0	0.0	0.0	0.0

The poc-graphql result shows why the recovery method matters. This graphql-java server never suggests a correct field name and returns only terse, structured errors, so GraphQLer and Clairvoyance had nothing to follow and recovered nothing. **MIKI-Fuzz**, in contrast, reads the shape of those errors to rebuild the schema and reached 61.7% where the others reached zero. Across all three targets, **MIKI-Fuzz** recovered more schema than either baseline, and on poc-graphql and Saleor it recovered a usable schema where both baselines recovered nothing, addressing **RQ1**.

4.3 Vulnerability Detection (RQ2)

This subsection evaluates whether the recovered schema leads to vulnerability detection. The baselines are GraphQLer (introspection-first, with wordlist-based recovery when introspection is disabled) and EvoMaster (requires introspection).
Table 2 summarizes the per-tool detection results for DVGA and poc-graphql.

Table 2. Per-tool detection results. MF: **MIKI-Fuzz**, GQLer: GraphQLer, EM: EvoMaster. ✓: detection signal confirmed, ✗: none. EvoMaster acquired no schema under disabled introspection. GraphQLer recovered schema on DVGA (Table 1) but confirmed no signal in any scenario.

Target	Category	Attack	MF	GQLer	EM
DVGA	Injection	SQL Injection	✓	✗	✗
		OS Command Injection	✓	✗	✗
		Stored XSS	✓	✗	✗
		HTML Injection	✓	✗	✗

Target	Category	Attack	MF	GQLer	EM
		SSRF	✓	✗	✗
		Path Traversal	✓	✗	✗
	Information Disclosure	Field Suggestions	✓	✗	✗
		Stack Trace Errors	✓	✗	✗
		Log Injection	✓	✗	✗
	Access Control	Query Deny List Bypass	✓	✗	✗
		Weak Password Protection	✓	✗	✗
		JWT Forgery	✓	✗	✗
		GraphiQL Protection Bypass	✓	✗	✗
	Denial of Service	Batch Query	✓	✗	✗
		Deep Recursion Query	✓	✗	✗
		Resource Intensive Query	✓	✗	✗
		Field Duplication	✓	✗	✗
		Field Aliasing	✓	✗	✗
poc-graphql	Injection	SQL Injection	✓	✗	✗
	Information Disclosure	Sensitive Field Exposure	✓	✗	✗
		Internal Error Disclosure	✓	✗	✗
	Access Control	Broken Access Control	✓	✗	✗
		IDOR	✓	✗	✗
	Denial of Service	Deep Recursion	✓	✗	✗
	Transport-layer Misconfiguration	Subscription over HTTP	✓	✗	✗
		CORS Misconfiguration	✓	✗	✗

Vulnerability Detection on DVGA. MIKI-Fuzz constructed queries that the server could process based on the recovered schema and confirmed detection signals across injection, information disclosure, access-control bypass, and denial-of-service scenarios. As shown in Table 2, **MIKI-Fuzz** confirmed detection in all 18 scenarios (6/6 injection, 3/3 information disclosure, 4/4 access control, 5/5 denial of service), while GraphQLer and EvoMaster confirmed no detection signals in any scenario.

Vulnerability Detection on poc-graphql. poc-graphql is a graphql-java server whose vulnerability scenarios are documented in its source. As shown in Table 2, **MIKI-Fuzz** confirmed detection signals in all 8 scenarios, 6 through valid queries generated from the recovered schema (1/1 injection, 2/2 information disclosure, 2/2 access control, 1/1 denial of service) and 2 through transport-layer inspection. SQL injection was confirmed from the response to a UNION payload. Lacking the schema and performing no transport-layer inspection, GraphQLer and EvoMaster missed all scenarios.

Vulnerability Detection on Saleor. MIKI-Fuzz confirmed three zero-day vulnerabilities in Saleor 3.23.12, independent of the public CVEs already patched in this build. Two are race conditions reached through the same recovered operations,

`voucherCreate` then `checkoutComplete`, but they abuse different limits. One over-redeems a single-use voucher past its usage limit for unlimited discounts, while the other oversells a stock-limited variant past its available inventory. The third is a privileged, sequential over-refund and is not a race.

This result demonstrates the connection between recovery and detection. Without response-based candidate expansion, these operations are not recovered and the requests cannot be generated (Section 4.4). With expansion they are recovered and detected, which addresses **RQ2**.

The three vulnerabilities were reported privately to the vendor, and reproduction details are withheld until public disclosure.

4.4 Ablation Study

To assess how much response-based candidate expansion contributes to schema recovery and vulnerability detection, we compare each target with and without expansion under identical run conditions. Table 3 reports this single-factor ablation. Isolating the remaining components is left to future work.

Table 3. Effect of response-based candidate expansion. w/o exp. uses only the fixed initial candidate list, and w/ exp. uses response-based expansion. Recovery is the Overall schema recovery rate (%). Detection is the number of scenarios with a confirmed detection signal.

Target	Recovery w/o exp.	Recovery w/ exp.	Detection w/o exp.	Detection w/ exp.
DVGA	87.6	100.0	18	18
poc-graphql	6.0	61.7	2	8
Saleor	1.1	13.3	0	3

As shown in Table 3, expansion improves both recovery and detection on every target where the initial list is insufficient. For DVGA, the core operations are already recovered from the initial candidates, so detection is identical with or without expansion. For poc-graphql, expansion raises recovery from 6.0% to 61.7% and detection from 2 to 8 scenarios. The effect is most decisive on Saleor, where expansion raises recovery from 1.1% to 13.3% and turns zero detected cases into three confirmed zero-day vulnerabilities. Expansion is therefore what reaches operations beyond the initial vocabulary, and what connects schema recovery to vulnerability detection.

4.5 Efficiency

To address efficiency, Table 4 reports the request count, runtime, and time to detection for each target. In every case the server remained responsive and recovered after fuzzing, indicating no lasting impact on normal operation.

Total requests counts all HTTP requests across the full pipeline, covering schema recovery and vulnerability detection. Runtime is the pipeline's wall-clock time. Time to first detection (TTFD) is the time from the start of detection to the first confirmed

vulnerability, excluding recovery, so it reflects how quickly the recovered schema yields a finding. Server recovered indicates the target returned to normal responses after fuzzing. Values are the mean and sample standard deviation over five runs. The request count was identical across the five runs. The runtime gap between DVGA and poc-graphql follows from the targets rather than the request budget, because at a comparable request count DVGA's denial-of-service scenarios hold each request open far longer.

Table 4. Efficiency of **MIKI-Fuzz** (mean ± sample standard deviation over five runs): total requests, wall-clock runtime, time to first detection (TTFD), and server availability after fuzzing.

Target	Total req	Runtime	TTFD	Server rec.
DVGA	1,106	189.5 ± 11.4 s	0.32 ± 0.02 s	Yes
poc-graphql	1,001	11.8 ± 0.04 s	3.08 ± 0.05 s	Yes
Saleor	1,046	93.5 ± 6.4 s	1.16 ± 0.07 s	Yes

5 Conclusion and Future Research

This paper proposes **MIKI-Fuzz**, a response-based black-box fuzzer for introspection-disabled GraphQL APIs. **MIKI-Fuzz** recovers a partial schema from validation errors and successful responses and connects the recovered information to query generation, structural mutation, and vulnerability detection. **MIKI-Fuzz** confirmed detection signals for every evaluated public vulnerability scenario on DVGA and poc-graphql under disabled introspection, and three previously undisclosed race conditions in a local Saleor 3.23.12 deployment, where limit enforcement breaks under concurrency.

The study has limitations. Recoverable schema is bounded by what the server exposes in error and successful responses. The HTTP-based implementation does not cover Subscription runtime execution, which needs WebSocket transport. Detection is reported per scenario rather than as a false-positive rate or an overhead on benign traffic. Future work will strengthen argument and type inference, extend Subscription and access-control coverage by combining WebSocket support with information-flow analysis, and add production-scale targets.

Acknowledgements. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00241376, Development of core security technology to respond to international smart ship regulations, 50%) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2026-25585348, Graduate School of Security Convergence(AI)(Gachon University), 50%)

References

1. Postman: 2025 State of the API Report. https://www.postman.com/state-of-api, , last accessed 2026/06/30
2. Gartner: When to Use GraphQL to Accelerate API Delivery. https://www.gartner.com/en/documents/7488253, last accessed 2026/06
3. Arcuri, A., Galeotti, J.P., Marculescu, B., Zhang, M.: EvoMaster: A Search-Based System Test Generation Tool. Journal of Open Source Software 6(57), 3055 (2021)
4. Tsai, O., Li, J., Cheung, T.T., Huang, L., Zhu, H., Xiao, J., Sharafaldin, I., Tayebi, M.A.: GraphQLer: Enhancing GraphQL Security with Context-Aware API Testing. arXiv:2504.13358 (2025)
5. Liu, S., Marefat, S., Tsai, O., Chen, Y., Deng, Z., Wang, J., Tayebi, M.A.: PrediQL: Automated Testing of GraphQL APIs with LLMs. arXiv:2510.10407 (2025)
6. Stupin, N.: Clairvoyance: Obtain GraphQL API Schema Even If Introspection Is Disabled. https://github.com/nikitastupin/clairvoyance, last accessed 2026/06/30
7. GraphQL Foundation: GraphQL: A Query Language for Your API. https://graphql.org, , last accessed 2026/06/30
8. Atlidakis, V., Godefroid, P., Polishchuk, M.: RESTler: Stateful REST API Fuzzing. In: Proc. 41st IEEE/ACM ICSE, pp. 748–758 (2019)
9. Cha, A., Wittern, E., Baudart, G., Davis, J.C., Mandel, L., Laredo, J.A.: A Principled Approach to GraphQL Query Cost Analysis. In: Proc. 28th ACM ESEC/FSE, pp. 257–268 (2020)
10. McFadden, S., Maugeri, M., Hicks, C., Mavroudis, V., Pierazzi, F.: WENDIGO: Deep Reinforcement Learning for Denial-of-Service Query Discovery in GraphQL. In: 2024 IEEE Security and Privacy Workshops (SPW), pp. 68–75 (2024)
11. Perera, I., Abeyrathne, H., Malalgoda, S., Ifthikar, A.: Enhancing GraphQL Security by Detecting Malicious Queries Using Large Language Models, Sentence Transformers, and Convolutional Neural Networks. arXiv:2508.11711 (2025)
12. Barabanov, A., Dergunov, D., Makrushin, D., Teplov, A.: Automatic Detection of Access Control Vulnerabilities via API Specification Processing. arXiv:2201.10833 (2022)
13. Lambers, L., Sakizloglou, L., Al-Wardi, O., Khakharova, T.: Taint Analysis for Graph APIs Focusing on Broken Access Control. In: Harmer, R., Kosiol, J. (eds.) ICGT 2024. LNCS, vol. 14774. Springer, Cham (2024)
14. Farhi, D., McKinnon, C.: Damn Vulnerable GraphQL Application (DVGA). https://github.com/dolevf/Damn-Vulnerable-GraphQL-Application, last accessed 2026/06/30
15. righettod: poc-graphql: Research on GraphQL from an AppSec Point of View. https://github.com/righettod/poc-graphql, last accessed 2026/06/30
16. Saleor Commerce: Saleor: A GraphQL-first, Headless E-commerce Platform. https://github.com/saleor/saleor, last accessed 2026/06/30

Quantum Circuit Optimization with LLMs under a Structured Guideline

Kyungbae Jang[1], Hyunji Kim[2], Hwajeong Seo[2], and Anupam Chattopadhyay[1]

[1] College of Computing and Data Science, Nanyang Technological University, Singapore
[2] Division of IT Convergence Engineering, Hansung University, Seoul, South Korea
starj1023@gmail.com, khj1594012@gmail.com, hwajeong84@gmail.com, anupam@ntu.edu.sg

Abstract. The cost of quantum cryptanalysis is dominated by the quantum circuit of the target cipher. Estimating the quantum attack cost of a cipher thus requires building that circuit and measuring its qubit count, Toffoli count, and Toffoli depth. This is manual work that needs expert knowledge and must be redone for each cipher and each cost target. Large language models handle ordinary programming well, but their use in constructing quantum circuits for ciphers is still limited. In this work, we collect quantum circuit optimization techniques that apply across many ciphers. We write these techniques into a guideline for a general-purpose LLM. Given this guideline and a single target cipher, the model produces two circuits. One minimizes the qubit count, and the other minimizes the Toffoli depth. Each circuit is verified against the test vectors of the cipher before its resources are estimated. Using this approach, we implement quantum circuits of CRAFT, MANTIS, QARMA, mCrypton, EPCBC, and Pyjamask for which quantum circuit implementations have not previously been reported. We further apply the same approach to ciphers with existing implementations. Without access to prior results, the generated circuits reach resource counts comparable to manually optimized ones.

Keywords: Quantum Computing · Quantum Circuit · Quantum Cryptanalysis · Large Language Model

1 Introduction

Quantum computers change how we judge the security of symmetric ciphers. Grover's algorithm searches a key space of size N in about $\sqrt{N}$ steps, so the real question is how costly that search actually is. NIST set the security levels in its post-quantum standardization from the concrete cost of running Grover's algorithm against AES [7,14]. The largest part of that cost is the quantum circuit of the cipher itself, which sits at the core of the Grover oracle. Implementing these circuits and measuring their cost in qubits, Toffoli gates, and Toffoli depth has thus become an active research topic. Building such a circuit is manual work that needs expert knowledge and must be redone for each cipher and each cost

target. Much of this knowledge is never written down and is reused by hand for every new cipher.

Large language models work well on ordinary programming, but their use for quantum circuit construction is still limited. Most of the existing work [16,6,23] targets variational circuits and OpenQASM generation, and reversible circuits for ciphers have drawn little attention. On its own, a model often returns circuits that are incorrect or circuits that work but use more qubits or gates than needed [23]. Other fields face the same problem and solve it by giving the model clear written instructions instead of retraining it. MetaGPT puts standard operating procedures into the prompts of its agents [8]. An agent skill groups instructions, scripts, and references into a module that is loaded only when needed [24]. A controlled study also shows that guidelines written by domain experts help a lot, while guidelines the model writes for itself do not [18].

We take the same approach for quantum circuit construction. We gather optimization techniques that work for many ciphers rather than just one, and write them up as a guideline for a general-purpose language model. With a reversible circuit simulator and resource counters, the model uses this guideline to build the circuits for target ciphers. For each cipher it produces two versions, one that lowers the qubit count and one that lowers the Toffoli depth. Using this guideline, we implement quantum circuits of symmetric ciphers, CRAFT, MANTIS, QARMA, mCrypton, EPCBC, and Pyjamask. Note that we adopted these ciphers because no quantum circuit for them has been public so far. We also apply the guideline to ciphers that already have hand-optimized implementations, LED, PRESENT, GIFT, and PRINCE [22,12,2], which lets us check the performance of the generated circuits. Note that the model was given no access to the implementations in [22,12,2].

1.1 Contribution

Our contributions are as follows:

- We write a guideline that collects quantum circuit optimization techniques reusable across ciphers. It is packaged as an agent skill, so a general-purpose LLM can use it without fine-tuning. We treat two targets separately, qubit count and Toffoli depth, since the two rarely admit the same circuit.
- We use the guideline to build quantum circuits for CRAFT, MANTIS, QARMA, mCrypton, EPCBC, and Pyjamask. None of these ciphers has a reported quantum circuit implementation. We verify each circuit against the test vectors and report its qubit count, Toffoli count, and Toffoli depth.
- We repeat the process on ciphers that already have hand-optimized implementations: LED, PRESENT, GIFT, and PRINCE [22,12,2]. The model is given no access to these results, yet it achieves performance close to the reported costs.

2 Foundations

2.1 Grover's Exhaustive Key Search

Grover's algorithm provides a quantum speedup for searching the key space of a cipher, cutting the classical search effort by a square-root factor. A k-bit key that costs $O(2^k)$ to search exhaustively on a classical machine can be recovered in about $\sqrt{2^k}$ steps with Grover search on a quantum computer. Given a known plaintext-ciphertext pair, the search for a k-bit key proceeds as *prepare* $\rightarrow$ (*Grover oracle* and *diffusion operator*)$^{\sqrt{2^k}}$ $\rightarrow$ *measure*.

The procedure starts by placing the key in superposition: a Hadamard (H) gate on each of the k qubits spreads the state evenly over all 2^k key values, as in Equation 1.

$$H^{\otimes k} |0\rangle^{\otimes k} (|\psi\rangle) = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} |x\rangle \tag{1}$$

X gates then set the plaintext register according to the known plaintext P_{known}. At the heart of the oracle sits the quantum circuit of the target cipher: it encrypts P_{known} under the k-qubit key $\psi(k)$, yielding a superposition of the ciphertexts produced by every candidate key. The oracle compares each of these with the known ciphertext C_{known}, and whenever they agree (Equation 2) it marks that key by flipping the sign of the corresponding state (Equation 3).

$$f(x) = \begin{cases} 1 \text{ if } Enc_{\psi(k)}(P_{known}) = C_{known} \\ 0 \text{ if } Enc_{\psi(k)}(P_{known}) \neq C_{known} \end{cases} \tag{2}$$

$$U_f(|\psi\rangle |-\rangle) = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle |-\rangle \tag{3}$$

The diffusion operator then raises the amplitude of the key marked by the oracle, making the correct key more likely to be observed. Applying the oracle and the diffusion operator in turn, repeated about $\sqrt{2^k}$ times, builds the amplitude of the solution high enough for the key to be measured with high probability. Because the encryption circuit of the cipher is the dominant component inside the oracle, reducing the resources it consumes is the key lever for lowering the overall cost of the Grover key search.

2.2 NIST Post-quantum Security

To measure how well a cipher resists quantum attacks, NIST set resource bounds for several levels [19,20]:

- Level 1: An attack costs about as much as breaking AES-128 ($2^{170} \rightarrow \mathbf{2^{157}}$).
- Level 2: An attack costs about as much as breaking SHA-256/SHA3-256.
- Level 3: An attack costs about as much as breaking AES-192 ($2^{233} \rightarrow \mathbf{2^{221}}$).

– Level 4: An attack costs about as much as breaking SHA-384/SHA3-384.
– Level 5: An attack costs about as much as breaking AES-256 ($2^{298} \rightarrow \mathbf{2^{285}}$).

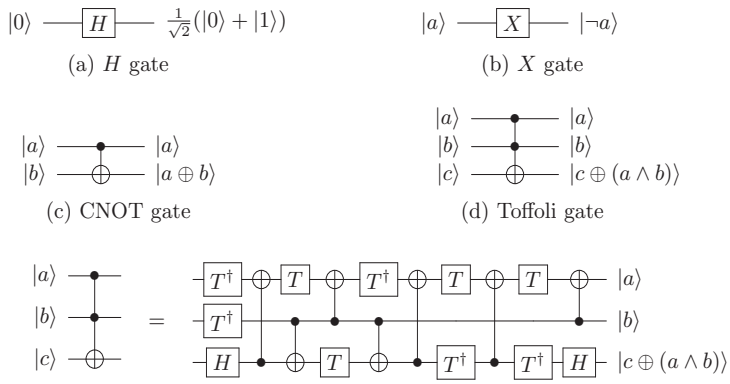
For Levels 1, 3, and 5, which correspond to the three AES variants, NIST took the attack complexities to be 2^{170}, 2^{233}, and 2^{298}, obtained as the total gate count multiplied by the depth of the Grover search, following the cost estimates of Grover key search on AES reported by Grassl et al. [7]. These figures in [19] therefore come from the PQCrypto'16 results [7]. Quantum circuits for AES have been refined steadily since then, and the cost of the attack has dropped considerably in recent years [15,25,10,9]. NIST itself notes that these per-level complexities are relative, precisely because quantum circuits for AES keep improving (see page 17 in [19]); a more efficient attack would call for the benchmarks to be revised. NIST has since updated the AES bounds [20] in light of the Eurocrypt 2020 results [15], where the quantum attack costs for AES-128, -192, and -256 fell to 2^{157}, 2^{221}, and 2^{285}, matching the revised values in [20].

NIST MAXDEPTH Even with Grover search, an exhaustive key search is still far out of reach for current quantum computers. The square-root reduction in security is only a theoretical bound. A working attack would have to run a quantum circuit of enormous depth. To capture this, NIST fixes a maximum allowable depth for a quantum attack (see page 46 of [20]). This limit is called MAXDEPTH and is set between 2^{40} and 2^{96}. Once an attacker exceeds MAXDEPTH, the Grover key search must be carried out in parallel by either an outer or an inner method [17]. However, parallelization of Grover's search is generally inefficient. This is because the reduction in depth is small compared with the loss in success probability. The product of depth and qubit count is a common measure of how good a quantum circuit is. But lowering the depth of the Grover search by a factor of S requires running S^2 instances in parallel [17,15]. The relevant cost for a parallel Grover search is therefore the depth^2 times qubit count rather than the depth times qubit count. For this reason, keeping the depth low becomes the more useful goal under parallelization.

2.3 Quantum Gates

Cryptographic algorithms are built in quantum circuits from a small set of gates, mainly the X (NOT), CNOT, and Toffoli (CCNOT) gates. The X gate is the quantum form of the classical NOT and inverts a qubit ($\mathrm{X}(x) = \sim x$). The CNOT gate acts on two qubits, a control and a target, and flips the target when the control is 1 and leaves it unchanged when the control is 0 ($\mathrm{CNOT}(x, y) = (x, x \oplus y)$). This is exactly an XOR of the control into the target, so CNOT plays the role of the classical XOR. The Toffoli gate acts on three qubits, two controls and one target, and flips the target only when both controls are 1 ($\mathrm{Toffoli}(x, y, z) = (x, y, z \oplus xy)$).

This XORs the AND of the two controls into the target, so the Toffoli gate plays the role of the classical AND. With these gates we can implement encryption algorithms on a quantum computer, replacing the classical NOT, XOR, and AND operations.



(a) H gate (b) X gate

(c) CNOT gate (d) Toffoli gate

(e) Toffoli gate decomposition from [1]

Fig. 1: Common quantum gates for quantum circuit implementations.

Optimizing a quantum circuit centers on reducing the number of Toffoli gates, since they are the most expensive to implement. A Toffoli gate is realized from T gates, which determine the T-depth, together with Clifford gates. Several ways to decompose the Toffoli gate exist, and the term *full depth* denotes the circuit depth after this decomposition. In this work we estimate resources using a decomposition with 7 T gates and 8 Clifford gates, which gives a T-depth of 4 and a full depth of 8 for one Toffoli gate [1].

3 Quantum Circuit Optimization with LLMs

In this section, we present our method for deriving optimized quantum circuits with LLMs. Figure 2 illustrates the proposed workflow for building optimized quantum circuits with LLMs. First, we create guideline for LLMs how to implement correct quantum circuits and how to optimize.

3.1 Guideline for LLMs

In the prompt, we include the guideline `Guideline.md`. Using it, the LLM generates a reversible quantum circuit for the encryption function of the cipher. The guideline covers various aspects of quantum implementation and its optimization, including verification. Note that it is designed for a wide range of ciphers rather than a specific one.

Verification of Quantum Circuits First, the guideline specifies what makes a quantum circuit valid for Grover based key search. Without this, we observed that the LLM sometimes reduces the resources of its implementation by fixing the key to a constant value, which makes the circuit invalid for quantum key search.

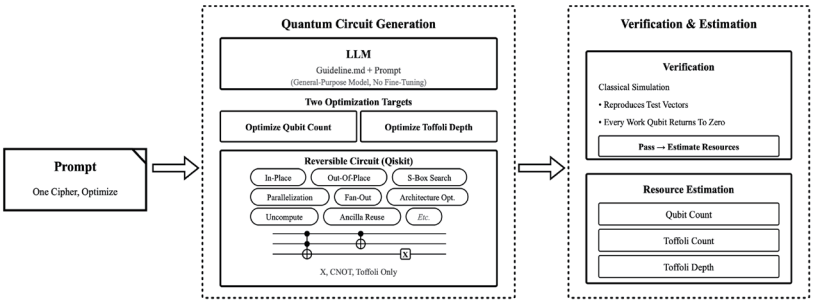


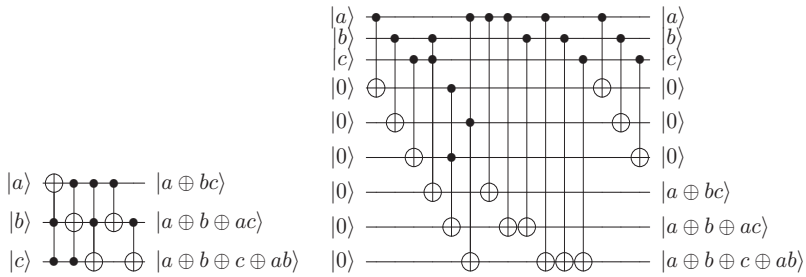
Fig. 2: Overview of the proposed LLM-guided quantum circuit generation

Following the guideline, the input plaintext and the input key are allocated as quantum registers. The generated circuit must then be verified against test vectors. Once it passes, the quantum resources it uses are estimated in terms of qubit count, Toffoli count, and Toffoli depth.

Optimization Strategy In our guideline, two optimization goals are specified and a separate circuit for each is generated by the LLM. One minimizes the qubit count, and the other minimizes the Toffoli depth. Depending on the optimization strategy, the LLM can select from the pool of techniques in the guideline those best suited to that goal. The guideline contains several techniques that can be applied when optimizing quantum circuits for cryptographic algorithms. Importantly, these techniques are generic, since they apply to a wide range of ciphers rather than to a single one.

In-Place and Out-of-Place Implementations In-place and out-of-place approaches are two common options for implementing quantum circuits. The in-place method computes the output on the input qubits, so it requires no separate output qubits. On the other hand, the out-of-place method allocates new output qubits and computes the result on them. Figure 3 shows an example of in-place and out-of-place implementations for same function. As expected, the in-place method suits the qubit count optimization, whereas the out-of-place method targets the Toffoli depth optimization by enabling greater parallelization. Beyond Figure 3, in-place and out-of-place implementations come in many forms. For example, among quantum adders, the Takahashi's adder [21] is in-place and suited to the qubit count optimization, whereas the Draper's out-of-place adder [5] is suited to the Toffoli depth optimization.

Recycling Work Qubits To reduce the qubit count, ancilla qubits are often reused through uncomputation. In the qubit count optimization, we uncompute the circuit to reset the ancilla qubits to $|0\rangle$ and then reuse them for later operations. Figure 4 illustrates this concept: once the result has been copied out, applying the $f^\dagger$ operation uncomputes the f operation and restores the work qubit



(a) In-place. (b) Out-of-place.

Fig. 3: In-place and shallow out-of-place implementations of the same function.

so that it can be reused. The technique is central to qubit optimization and is applied broadly across quantum implementations. Note that uncomputation roughly doubles the circuit depth.

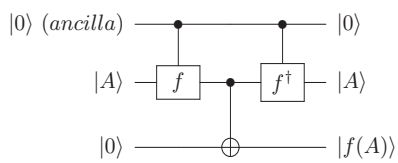
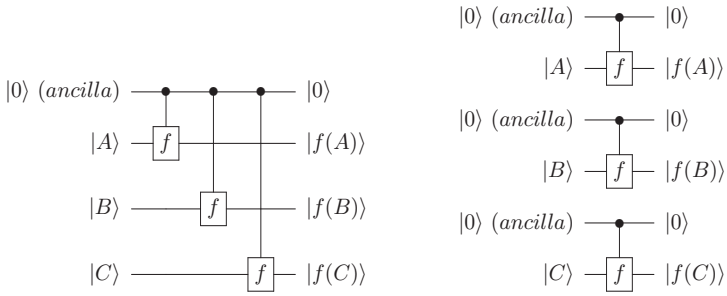


Fig. 4: Cleaning a qubit through uncomputation ($f^\dagger$).

Balancing Qubit Count and Depth Optimizing too heavily for one side often fails to achieve balanced performance. Suppose that one ancilla qubit is required for each f operation and that a single round needs 3 f operations. To minimize the qubit count, we can reuse a single ancilla qubit across all 3 operations. Figure 5(a) illustrates this approach. However, this forces the operations to run serially and increases the depth substantially. In such a case, it is better to parallelize the f operations by allocating 3 ancilla qubits and reusing them across rounds. Figure 5(b) shows this strategy. The right choice therefore depends on the target, and a good implementation balances qubit count against depth rather than optimizing either in isolation[3]. Thus, the guideline advises that when a relatively small number of additional qubits can improve parallelization effectively, those qubits should be allocated.

S-box Synthesis The S-box dominates the Toffoli cost and largely determines the overall optimization of the quantum circuit. We search over $\{X, \mathrm{CNOT}, \mathrm{CCX}\}$

[3]As a side note, this trade-off has already been discussed for the SubBytes implementation of AES quantum circuits in [11, Section 3.3].



(a) Sequential operation (b) Parallelization

Fig. 5: Reuse of a single ancilla versus multiple ancillas.

for the ancilla-free in-place implementation with the fewest Toffoli gates, which gives a provably minimal result[4]. An exact search matters here because approximate methods do not guarantee a minimal Toffoli count and may introduce multi-controlled gates (C^nX, $n \geq 3$) that fall outside this gate set. For our Toffoli depth optimization, a minimal Toffoli count does not imply a minimal depth. An in-place implementation shares wires and runs its Toffoli gates serially, so reducing the depth requires an out-of-place implementation.

Optimization at the Architecture Level The optimizations described so far act on individual operations. A higher level of optimization looks beyond a single operation and considers how the rounds of the cipher are composed as a whole. Inspired by the *shallow* architecture of [11], we generalize this architecture-level optimization. We argue that this optimization applies generically to any cipher with a round-based structure. Figure 6 shows the quantum circuit of the shallow architecture. The uncomputation of one round (i.e., $f^\dagger$) is run in parallel with the normal computation (i.e., f) of the next round. To enable this overlap two sets of ancilla qubits (i.e., $ancilla_0$ and $ancilla_1$) are maintained, one for each of the two rounds that are active at the same time. Put simply, one ancilla set is used for odd rounds and another ancilla set is used for even rounds.

One Cipher per Prompt. Asking the LLM to implement multiple ciphers in a single prompt highly degrades its output. This effect is consistent with prior reports that batched or multi-task prompting degrades LLM performance [4,3]. For quantum circuit optimization, the degradation appears far more severe.

4 Results

In this section we report the quantum circuits produced under the guideline. The model receives the guideline together with a prompt naming a single target

[4]This approach does not scale; for larger S-boxes the space is too large to enumerate, and heuristic synthesis becomes necessary.

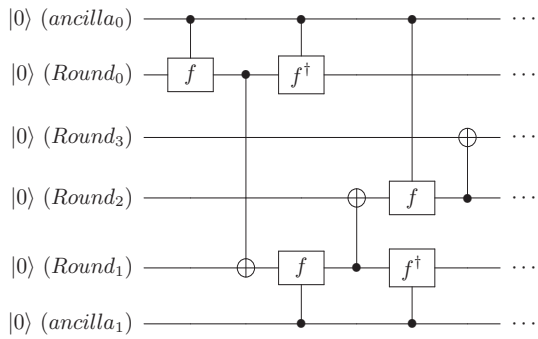


Fig. 6: Generic form of the shallow architecture in [11].

cipher and returns two circuits: one minimizing the qubit count (Q *opt*) and one minimizing the Toffoli depth (D *opt*). Each circuit is simulated classically and is accepted only when it reproduces the test vectors of the cipher. For an accepted circuit we then estimate the qubit count, the Toffoli count, and the Toffoli depth.

We adopt `GPT-5.5` as the general-purpose language model, with no fine-tuning. To keep the runs independent and reproducible, we generate every circuit in a temporary chat session with memory disabled, so that no context from earlier conversations carries over into the current generation.

This work focuses on ciphers for which no quantum circuit implementation has previously been publicly available: CRAFT, MANTIS, QARMA, mCrypton, EPCBC, and Pyjamask. Table 1 reports these results. For the Q opt version, the generated circuits use the optimal number of qubits, equal to the sum of the block, tweak, and key sizes. For the D opt version, the circuits allocate additional qubits to reduce the Toffoli depth.

Table 2 compares the circuits generated by our workflow with previously reported implementations of the same ciphers, LED, PRESENT, GIFT, and PRINCE [22,12,2]. In the qubit-optimized setting the results are nearly identical. Lightweight block ciphers have simple round structures, and the qubit count is determined mainly by the in-place S-box construction. The minimum is therefore close to fixed and leaves little room for divergence between independent implementations. We do not claim an improvement here. The agreement does indicate that the generated circuits reach the same resource figures as manually optimized ones. To rule out the possibility that this comes from recall of published circuits, we prepared the generation environment so that no prior implementation, paper, or reported resource count was accessible to the model. Each circuit was derived only from the cipher specification and our implementation guideline

4.1 Discussion

Our guideline based approach performs well on lightweight ciphers with relatively simple round structures. For these ciphers the generic techniques in the guideline

Table 1: Quantum resource estimation for lightweight block ciphers without prior full-cipher quantum implementations.

Cipher	Version	Qubits	Toffoli count	Toffoli depth
CRAFT-64/128	Q opt	256	1984	124
	D opt	2720	7680	62
MANTIS5-64/128	Q opt	256	768	48
	D opt	1600	3168	24
MANTIS7-64/128	Q opt	256	1024	64
	D opt	1856	4320	32
QARMA64/128	Q opt	256	1280	80
	D opt	1888	4800	32
QARMA128/256	Q opt	512	3840	120
	D opt	4800	14720	48
mCrypton-64/64	Q opt	148	2072	182
	D opt	1456	3760	38
EPCBC-48/96	Q opt	144	4608	256
	D opt	4080	11904	64
Pyjamask-96/128	Q opt	224	1344	42
	D opt	1952	2496	14
Pyjamask-128/128	Q opt	256	1792	56
	D opt	2688	3328	28

are sufficient, and the LLM produces correct circuits whose qubit count, Toffoli count, and Toffoli depth are competitive. However, it does not yet match highly optimized implementations for ciphers with more complex structures. AES is a representative example. Starting from the first quantum resource estimates of Grassl et al. [7], a long line of work has steadily refined the AES quantum circuit through increasingly dedicated optimizations [14,9,10], reaching a very high level today. Our guideline focuses on techniques that generalize across many ciphers. Extending it toward the high-level optimizations used in well-studied implementations is left as future work.

5 Conclusion

We study whether a general purpose LLM can build optimized quantum circuits for symmetric ciphers when it is handed a written guideline instead of being fine tuned. The guideline collects optimization techniques that the model can follow on its own. With this single artifact and no model training, the LLM produces circuits for CRAFT, MANTIS, QARMA, mCrypton, EPCBC, and Pyjamask, none of which have a public quantum implementation. Note that every circuit passes verification before its resources are estimated.

Table 2: Comparison of the generated circuits with previously reported implementations.

Cipher	Source	Qubits	Toffoli count	Toffoli depth
LED-64	Wang et al. [22]	128	2048	128
	This paper (Q opt)	128	2048	128
	This paper (D opt)	2912	10912	64
PRESENT-80	Jang et al. [13]	144	2108	124
	This paper (Q opt)	144	2108	124
	This paper (D opt)	5628	11220	62
GIFT-64	Jang et al. [13]	192	1792	112
	This paper (Q opt)	192	1792	112
	This paper (D opt)	2688	6912	56
PRINCE	Bonnetain and Jaques [2]	192	1152	72
	This paper (Q opt)	192	960	60
	This paper (D opt)	2176	4768	24

The guideline treats the two optimization objectives separately. For each cipher the model returns one circuit that minimizes the qubit count and one that minimizes the Toffoli depth. Overall, this approach produces correct and competitive circuits for target ciphers. However, it does not yet reach the highly optimized implementations developed through extensive prior research, as the AES case shows. In this context, closing this gap by adding high level and cipher specific techniques to the guideline remains future work.

6 Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2019-II190033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 50%) and this work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2025-23523436, 50%).

References

1. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (Jun 2013). https://doi.org/10.1109/tcad.2013.2244643, http://dx.doi.org/10.1109/TCAD.2013.2244643 5
2. Bonnetain, X., Jaques, S.: Quantum period finding against symmetric primitives in practice. arXiv preprint arXiv:2011.07022 (2020) 2, 9, 11

3. Chen, J., Pilehvar, M.T., Camacho-Collados, J.: Understanding llm performance degradation in multi-instance processing: The roles of instance count and context length. arXiv preprint arXiv:2603.22608 (2026) 8
4. Di Maio, F., Gozzi, M.: Degradation of multi-task prompting across six nlp tasks and llm families. Electronics **14**(21), 4349 (2025) 8
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
6. Fu, Z., Chen, F., Jiang, L.: Qagent: An llm-based multi-agent system for autonomous openqasm programming. arXiv preprint arXiv:2508.20134 (2025) 2
7. 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) 1, 4, 10
8. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Yau, S., Lin, Z., Zhou, L., et al.: Metagpt: Meta programming for a multi-agent collaborative framework. In: International Conference on Learning Representations. vol. 2024, pp. 23247–23275 (2024) 2
9. Huang, Z., Sun, S.: Synthesizing quantum circuits of AES with lower T-depth and less qubits. Cryptology ePrint Archive, Report 2022/620 (2022), https://eprint.iacr.org/2022/620 4, 10
10. 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 4, 10
11. Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., Chattopadhyay, A.: Quantum analysis of aes. IACR Communications in Cryptology **2**(1) (2025) 7, 8, 9
12. Jang, K., Song, G., Kim, H., Kwon, H., Kim, H., Seo, H.: Efficient implementation of present and gift on quantum computers. Applied Sciences **11**(11), 4776 (2021) 2, 9
13. Jang, K., Song, G., Kim, H., Kwon, H., Kim, H., Seo, H.: Efficient implementation of PRESENT and GIFT on quantum computers. Applied Sciences **11**(11) (2021), https://www.mdpi.com/2076-3417/11/11/4776 11
14. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing grover oracles for quantum key search on AES and LowMC. Cryptology ePrint Archive, Report 2019/1146 (2019), https://eprint.iacr.org/2019/1146 1, 10
15. 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 4
16. Jern, L., Uotila, V., Yu, C., Zhao, B.: Agent-q: fine-tuning large language models for quantum circuit generation and optimization. In: 2025 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 1, pp. 1621–1632. IEEE (2025) 2
17. Kim, P., Han, D., Jeong, K.C.: Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to aes and sha-2. Quantum Information Processing **17**, 1–39 (2018) 4
18. Li, X., Chen, W., Liu, Y., Zheng, S., Chen, X., He, Y., Li, Y., You, B., Shen, H., Sun, J., et al.: Skillsbench: Benchmarking how well agent skills work across diverse tasks. arXiv preprint arXiv:2602.12670 (2026) 2

19. 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 3, 4
20. NIST.: Stateless hash-based digital signature standard (2023), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.ipd.pdf 3, 4
21. Takahashi, Y., Tani, S., Kunihiro, N.: Quantum addition circuits and unbounded fan-out (2009), https://arxiv.org/abs/0910.2530 6
22. Wang, S., Jang, K., Baksi, A., Chakraborty, S., Lee, B., Chattopadhyay, A., Seo, H.: New results in quantum analysis of led: Featuring one and two oracle attacks. In: International Conference on Cryptology in India. pp. 339–364. Springer (2025) 2, 9, 11
23. Yu, C., Uotila, V., Deng, S., Wu, Q., Shi, T., Jiang, S., You, L., Zhao, B.: Quasar: Quantum assembly code generation using tool-augmented llms via agentic rl. arXiv preprint arXiv:2510.00967 (2025) 2
24. Zhang, B., Lazuka, K., Murag, M.: Equipping agents for the real world with agent skills. Anthropic Engineering Blog (2025) 2
25. Zou, J., Wei, Z., Sun, S., Liu, X., Wu, W.: Quantum circuit implementations of AES with fewer qubits. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 697–726. Springer International Publishing, Cham (2020) 4

Improved Quantum Circuits for Information Set Decoding with Application to Code-Based Cryptography

Hyunji Kim[1], Kyungbae Jang[2], and Hwajeong Seo[1†]

[1] Division of IT Convergence Engineering, Hansung University, Seoul, South Korea
[2] College of Computing and Data Science, Nanyang Technological University, Singapore
khj1594012@gmail.com, starj1023@gmail.com, hwajeong84@gmail.com

Abstract. Information set decoding (ISD) is the standard generic decoding attack considered for code-based cryptography. A concrete quantum-resource estimate for Grover-accelerated ISD requires an oracle whose dominant component is Gauss–Jordan elimination.
We improve the elimination circuit of Perriello et al. [25] and Jang et al. [15] by not updating the entries that no later pivot or the final weight predicate reads. The required result vector is recovered by a parallel back-substitution on the syndrome register. For the target schemes, our elimination circuit improves the qubit count by about 22% compared to [15]. The Toffoli count improves by about 20% compared to both [25] and [15], while the Toffoli depth remains comparable to [15] and is about 50% lower than the value derived from the implementation of [25].
We report logical resource estimates for the quantum ISD attack on HQC and Classic McEliece. The product of the total gate count and the full depth exceeds the NIST post-quantum security thresholds, and the full depth exceeds the MAXDEPTH upper bound.
We also provide fault-tolerant estimates of the physical qubit count and the runtime under a surface-code model with magic-state distillation. As one example, HQC-128 requires about 2^{43} physical qubits and about 2^{53} years at a 1 μs code cycle.

Keywords: Grover's Algorithm · Code-based Cryptography · Quantum Cryptanalysis · Information Set Decoding

1 Introduction

Code-based cryptography is a leading family of post-quantum cryptography: NIST has recently selected HQC [1] for standardization as a key-encapsulation mechanism [23], while Classic McEliece [6] remains a long-studied, conservative alternative. Both schemes rely on the hardness of syndrome decoding, believed to resist both classical and quantum attack, though their code families differ: Classic McEliece uses binary Goppa codes and HQC uses quasi-cyclic codes.

†: Corresponding author.

Knowing how much quantum effort such an attack would actually take is therefore important for setting and trusting their parameters.

The best-known generic decoding attack on these schemes is *information set decoding* (ISD), a family of decoders that still runs in exponential time [26]. On a quantum computer, Grover's search lowers the cost of ISD by a square root [5], so its concrete cost is a central measure of post-quantum security against generic decoding. Turning this idea into a resource estimate requires an explicit quantum circuit for one ISD iteration, whose dominant component is a Gauss–Jordan elimination over $\mathbb{F}_2$.

Several circuit-level studies give concrete estimates: Perriello et al. presented the first complete quantum circuit for ISD [24] and later improved it [25], reporting resource estimates for HQC, BIKE, and Classic McEliece. Jang et al. focused on the dominant linear-algebra subroutine and proposed a depth-optimized parallel circuit for binary Gauss–Jordan elimination [15]. More recently, Chevignard et al. studied a low-qubit alternative based on Wiedemann sparse inversion, which optimizes a different point of the time–space trade-off [8].

In this work we improve the elimination oracle, lowering its qubit and gate counts with only a lower-order depth change, and evaluate the ISD attack on HQC and Classic McEliece at both the logical and physical level.

Contribution

Our contributions are the following.

- We propose a modified elimination procedure for the ISD oracle based on [15]. It keeps their parallel pivot fan-out, but avoids the redundant row updates and instead uses a parallel back-substitution that recovers $H_{\mathcal{S}}^{-1}s$ (Section 3.2). This lowers the qubit count by about 22% and the T-count by about 20%, while the T-depth increases only by the lower-order term $4r-4$.
- We report logical estimates for the quantum ISD attack on HQC and Classic McEliece and evaluate them against the NIST criteria (Sections 4). For every target the gate count times the full depth and the full depth exceed the NIST security-level threshold and the MAXDEPTH upper bound, respectively.
- For a more realistic cost assessment, we provide fault-tolerant estimates under the surface code with magic-state distillation (Section 4.2). As one of our results, HQC-128 requires about 2^{43} physical qubits and 2^{53} years.

2 Background

2.1 Information Set Decoding

Code-based cryptosystems rely on the hardness of the *Syndrome Decoding Problem* (SDP), whose bounded-weight decision version is NP-complete [4]. For a binary $[N,k]$ code with parity-check matrix $H \in \mathbb{F}_2^{(N-k)\times N}$ and redundancy $r = N-k$, the SDP seeks, given a syndrome $s = He^\top \in \mathbb{F}_2^r$ and a weight t, an error e with $\mathrm{wt}(e) = t$.

The two schemes we analyze instantiate the SDP over different code families: Classic McEliece [6] over binary Goppa codes, and HQC [1], recently selected by NIST, over quasi-cyclic codes. For these schemes, *information set decoding* (ISD) is regarded as the best known attack.

We adopt the parameter sets from the current specifications, Classic McEliece [9, Section 7] and HQC [14, Section 4.1]. For Classic McEliece, each set lists m, N, and t, from which $r = mt$ and $k = N - r$. For HQC, the key-recovery attack decodes the underlying $[2n, n]$ quasi-cyclic code with secret weight $t = 2w$, so that $N = 2n$ and $r = n$, where n and w are the ambient length and weight given in [14, Table 5]. The circuit cost depends only on N and r, while the iteration count also depends on t.

Given an SDP instance (H, s, t), Prange's information set decoding [26] selects r column positions $\mathcal{S}$ and forms the submatrix $H_{\mathcal{S}}$. If $H_{\mathcal{S}}$ is invertible, the vector e equal to $H_{\mathcal{S}}^{-1}s$ on $\mathcal{S}$ and zero elsewhere satisfies $He^{\top} = s$, and it solves the SDP when all t error positions lie in $\mathcal{S}$.

A random $\mathcal{S}$ captures all t errors with probability $p_{\text{sup}} = \binom{r}{t}/\binom{N}{t}$ and is invertible with probability about 0.288, giving the Prange success probability p. Grover's search turns the classical $1/p$ repetitions into about $1/\sqrt{p}$ iterations, and we take the concrete counts from the Prange analysis of [25], which already includes the invertibility factor. More advanced ISD and quantum-walk variants lower the asymptotic work factor [17,19], but no circuit translation considered in [25,10] gives a clear concrete advantage under our cost model, so we adopt Prange as the circuit-level baseline [5].

2.2 Grover's Algorithm in Quantum ISD

Grover's algorithm gives a quadratic speedup for unstructured search [13]. In this work, the search problem is to find a column set S for which one Prange iteration succeeds. The search register is therefore prepared in a uniform superposition over all r-column choices, not only over nonsingular choices. The oracle marks a choice S when $H_{\mathcal{S}}$ is nonsingular and the corresponding vector $H_{\mathcal{S}}^{-1}s$ has Hamming weight t.

For each S in superposition, the oracle gathers the selected columns, forms the submatrix $H_{\mathcal{S}}$, runs the elimination circuit, and checks the Hamming weight of the resulting syndrome vector. Singular branches are not aborted; they simply fail the nonsingularity condition and are not marked. If the nonsingularity and weight tests both succeed, a decision qubit, equivalently a phase oracle, flips the phase of that basis state. The work registers are then uncomputed, so that only the phase mark on the successful choices remains.

After the oracle, the diffusion operator amplifies the marked choices. Repeating the oracle and diffusion operator increases the amplitude of the successful column sets, and a measurement returns a marked set with high probability. If one classical Prange iteration succeeds with probability p, Grover search uses about $\pi/(4\sqrt{p})$ oracle calls instead of the classical $1/p$ trials. In this paper, p is the Prange success probability defined in Section 2.1, and the oracle whose cost we optimize is the Prange-oracle circuit constructed in Section 3.

2.3 Quantum Gates

We adopt a Toffoli decomposition using 7 T-gates at a T-depth of 4 [3]. Where the target qubit is guaranteed clean, we instead use the AND gate of Jaques et al. [16], which costs 4 T-gates at a T-depth of 1 with one ancilla, and whose uncomputation AND† needs only Clifford gates and a measurement [12]. The input stage also uses the single-qubit rotation R_y, which lies outside Clifford+T and is synthesized into about 149 T-gates at precision 10^{-15} [20].

3 Quantum Circuit for Information Set Decoding

This section presents our quantum circuit for the Grover oracle. The oracle marks the column sets $\mathcal{S}$ in superposition for which $H_{\mathcal{S}}^{-1}s$ has weight t. The QISD oracle is composed of the following three arithmetic modules. The *input setting* first prepares the superposition and gathers the selected columns into the submatrix $H_{\mathcal{S}}$ (see Section 3.1). From this submatrix, the *Gauss–Jordan elimination* computes $H_{\mathcal{S}}^{-1}s$ (see Section 3.2), after which a *Hamming weight check* determines the decision qubit whenever the result has weight t (see Section 3.3). The elimination is by far the most expensive stage, so we focus on optimizing the Gauss–Jordan elimination circuit.

3.1 Input Setting

The input setting places the $\binom{N}{r}$ column choices in superposition and gathers the selected submatrix $H_{\mathcal{S}}$ into a contiguous block for the later stages. The choice is a *Dicke state* $\left|D_r^N\right\rangle$ on an N-qubit index register, prepared by the construction of [21], and the marked columns of H are then moved to the left by the reversible sorting network of [25, Section 3.1], to which we refer for both constructions. Perriello et al. [25, Section 3.1 and the footnote of Table 3] fix every gate to depth one. Table 1 therefore reports the required quantum resources, re-derived under the Toffoli decomposition we adopt (see Section 2.3).

Table 1: Quantum resources for the input setting, re-expressed in closed form.

Stage	#Qubit	#T	T-depth
Dicke state [21]	N	$149\,(4Nr - 4r^2 - 2N + 1)$	–
Sorting network [25]	$Nr + N + \dfrac{N\ell(1+\ell)}{4}$	$\dfrac{N\ell(1+\ell)}{4}\,(7r+7)$	$2\ell(1+\ell) + 4r$

N: code length, $r = N - k$, and $\ell = \lceil \log_2 N \rceil$.

3.2 Gauss–Jordan Elimination

The elimination stage transforms $[\,H_{\mathcal{S}} \mid s\,]$ until the syndrome register holds $H_{\mathcal{S}}^{-1}s$, processing the r pivots one after another. Unlike a full identity-reduction of the selected matrix, our circuit only preserves the entries needed to recover

$[\,H_{\mathcal{S}}^{-1} \mid s\,]$ for the final weight test. Each pivot is handled in two steps. The first sets the diagonal entry H_{jj} to 1: if it is already 1, nothing is done, and otherwise a lower row whose entry in column j is 1 is added (i.e., XOR) to the pivot row. The second conceptually uses the pivot row to eliminate the entries below the pivot in column j by adding the pivot row to each lower row whose entry in column j is 1.

Gauss–Jordan elimination without redundant updates. We adopt forward elimination followed by back-substitution to update only the matrix elements that later steps use. This lowers both the gate count and the qubit count. At each pivot, [15, Section 3.3] adds the pivot row to every lower row with a 1 in the pivot column and updates the whole row: the left elements of the pivot (L), its right elements (R), and the syndrome elements S, requiring ancilla copies of the pivot data for each update.

Here, our key observation is that later steps do not refer to L. A later pivot updates only the right columns R, while the back-substitution and the Hamming weight test refer only to the upper-triangular part and the syndrome.

We therefore update only R and S without L (see Figure 1). The back-substitution recovers $H_{\mathcal{S}}^{-1}s$ from the retained part, so the result is identical. Removing the L-updates lowers the leading Toffoli and qubit coefficients, while the added back-substitution contributes only lower-order terms.

	col0	col1	col2	col3	col4	col5	col6	col7	s
row0	.	.	.	.	.	.	.	.	.
row1	.	.	.	.	.	.	.	.	.
row2	.	.	P	.	.	.	.	.	.
row3	L	L	.	R	R	R	R	R	S
row4	L	L	.	R	R	R	R	R	S
row5	L	L	.	R	R	R	R	R	S
row6	L	L	.	R	R	R	R	R	S
row7	L	L	.	R	R	R	R	R	S

	col0	col1	col2	col3	col4	col5	col6	col7	s
row0	.	.	.	.	.	.	.	.	.
row1	.	.	.	.	.	.	.	.	.
row2	.	.	P	.	.	.	.	.	.
row3	.	.	.	R	R	R	R	R	S
row4	.	.	.	R	R	R	R	R	S
row5	.	.	.	R	R	R	R	R	S
row6	.	.	.	R	R	R	R	R	S
row7	.	.	.	R	R	R	R	R	S

P: pivot L: lower-left updates R: lower-right updates S: s updates .: not updated at this pivot

Figure 1: Update targets at one pivot P (left: baseline [15]; right: ours).

Algorithm 1 states the full procedure, writing H for $H_{\mathcal{S}}$. Its forward loop updates only R and the syndrome. In general, some column choices give a singular $H_{\mathcal{S}}$ where not every pivot can be set, so PivotFix records the success of each pivot. The phase flip is applied only when all pivots succeed and the weight is t. On a nonsingular branch, the forward elimination corresponds to a product E of elementary row operations, leaving the entries on and above the diagonal of the unit upper-triangular matrix $U = EH_{\mathcal{S}}$ in the matrix register and the eliminated syndrome $\tilde{s} = Es$ in the syndrome register. The back-substitution solves $Ue = \tilde{s}$, returning $e = H_{\mathcal{S}}^{-1}s$ because $U^{-1}\tilde{s} = (EH_{\mathcal{S}})^{-1}Es = H_{\mathcal{S}}^{-1}s$. With $U_{i,j} = H_{ij}$ for

the retained coefficients ($i < j$), it computes $e_i = \tilde{s}_i + \sum_{j=i+1}^{r-1} U_{i,j}\, e_j$ for $i = r-1$ down to 0, in place on the syndrome register.

We parallelize the back-substitution so that its effect on the depth stays low. In Algorithm 1, the outer loop starts with s_{r-1} and proceeds to s_1. In the iteration for s_j, this component is used to update $s_0, \ldots, s_{j-1}$. Each of these j updates has a different target qubit. Fan-out of s_j provides a separate control for each update, allowing all of them to be executed in one Toffoli layer. This parallelism keeps the additional T-depth of the back-substitution to a lower-order term. The iterations are sequential because the iteration for s_j updates s_{j-1}, which is used in the next iteration. Here, the fan-out ancillas are uncomputed after each iteration and reused. Moreover, the numbers of Toffoli gates are $r-1, r-2, \ldots, 1$ across the iterations, giving a total of $r(r-1)/2$ Toffoli gates for the back-substitution. Figure 2 shows the circuit for $r = 3$. First, s_2 is used to update s_1 and s_0 in parallel. The updated s_1 is then used to update s_0.

Algorithm 1: Gauss–Jordan elimination with syndrome back-substitution.

Require: $[\,H \mid s\,]$ with $H \in \mathbb{F}_2^{r\times r}$ (H_{ik} is row i, column k)
Ensure: $s_{\text{out}} = H^{-1} s_{\text{in}}$ on the nonsingular branches

```
for j = 0 to r − 1 do                                  ▷ forward elimination
    ok_j ← PivotFix(H, j)                              ▷ set the pivot H_jj to 1; record success
    for i = j + 1 to r − 1 do
        for k = j + 1 to r − 1 do                      ▷ right block R only
            H_ik ← H_ik ⊕ H_ij H_jk
        end for
        s_i ← s_i ⊕ H_ij s_j                           ▷ syndrome
    end for
end for
for j = r − 1 downto 1 do                              ▷ back-substitution
    for i = 0 to j − 1 do
        s_i ← s_i ⊕ H_ij s_j
    end for
end for
```

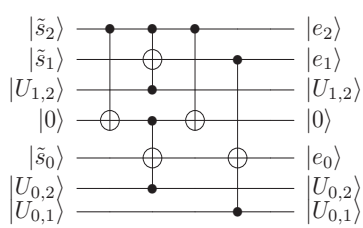


Figure 2: Parallel back-substitution at the gate level for $r = 3$.

Table 2 compares the elimination costs with those of Perriello et al. [25] and Jang et al. [15]. Jang et al. [15] report concrete counts only for matrix sizes up to 48, whereas Perriello et al. [25] use a different Toffoli decomposition. We express all three in closed form[3]. We compare at the Toffoli level to assess the effect of the algorithmic optimization on its own, independent of the choice of Toffoli decomposition. For [15], the closed-form expressions in Table 2 reproduce the reported qubit, T-count, Toffoli-depth, and T-depth values for all evaluated matrix sizes up to 48. For [25], we do not use their reported circuit depth directly, because it includes non-Toffoli gates, each assigned unit depth. Instead, the closed-form Toffoli depth in Table 2 is derived from their implementation[4] by counting only the CCNOT layers while preserving the gate dependencies. For the parameters of target schemes, our circuit improves the qubit count by about 22% and the Toffoli count by about 20% over [15], while the Toffoli-depth increases only by the lower-order term $r-1$, which is negligible relative to the leading $\frac{1}{2}r^2$. Compared with [25], our Toffoli depth is about 50% lower: its leading term is $\frac{1}{2}r^2$, compared with r^2 for their circuit.

Table 2: Quantum resources for the Gauss-Jordan elimination compared with prior works [25,15].

Source	#Qubit*	#T*	T-depth*	#Toffoli✻	Toffoli-depth✻
P^+ [25]	–	–	–	$\frac{r(r-1)(5r+8)}{6}$	r^2+2r-1
J^+ [15]	$\frac{9r^2-11r+6}{2}$	$\frac{7r(5r^2+3r-8)}{6}$	$2r^2+2r-4$	$\frac{r(5r^2+3r-8)}{6}$	$\frac{r^2+r-2}{2}$
This work✿	$\frac{7r^2-5r+4}{2}$ (22%)	$\frac{7r(2r^2+3r-5)}{3}$ (20%)	$2r^2+6r-8$ (−0.3%)	$\frac{r(2r^2+3r-5)}{3}$ (20%/20%)	$\frac{r^2+3r-4}{2}$ (50%/−0.3%)

✻ and ✻: Improvement percentages relative to J^+ [15] and P^+ [25]/J^+ [15], respectively (a negative value denotes a degradation).
✿: Our work includes the added syndrome back-substitution.

3.3 Hamming Weight Check

The last stage of the oracle turns $H_{\mathcal{S}}^{-1}s$ in the syndrome register into a marking decision: it computes the Hamming weight of this r-bit vector and flips the sign of the state when the weight equals t. Prior work computes this weight with the adder tree of [25, Section 2.6]. We instead accumulate all r bits in parallel with the tree-based quantum carry-save adder [18], which lowers the depth. This requires more ancillas than the adder tree, but we reuse the clean ancilla qubits from the elimination phase, achieving a lower T-depth without additional qubits.

[3] The full depth is not given in closed form, as it depends on how the Clifford and Toffoli gates are scheduled, and is reported only for explicitly generated circuits.
[4] https://github.com/paper-codes/2023-TQuantum

4 Quantum Resource Estimates for ISD on HQC and Classic McEliece

In this section, we present quantum resource estimates for the ISD attack on the two schemes, obtained by combining the oracle of Section 3 with the Grover iteration count of [25]. We report logical costs, then derive physical qubits and runtime from them under a surface-code model, and finally assess both against the NIST security levels and the MAXDEPTH.

Table 3[5] reports the resulting logical quantum resources, which include the three oracle stages (i.e., sorting, elimination, and weight check). For HQC, the iteration count includes the decoding-one-out-of-many speedup that the quasi-cyclic structure permits [25, Section 3.1], lowering it by about $\sqrt{r}$. Classic McEliece, whose Goppa codes lack this structure, uses the plain single-syndrome count. The two schemes exhibit different cost trade-offs: HQC has larger N and r, which decide the matrix size, so its oracle uses more qubits, whereas its iteration count is smaller.

Another line of work [8] replaces Gauss–Jordan elimination with Wiedemann's sparse matrix inversion, minimizing the number of qubits instead of circuit depth and gates. As one illustrative point in this trade-off, we roughly compare with the space-optimized variant of Chevignard et al. [8, Table 1] (excluding the Grover iteration count). For instance, on Classic McEliece-348864 our circuit requires about 2^8 fewer Toffoli gates ($2^{30.2}$[6] against $2^{37.8}$ of their space-optimized version), while theirs requires about 2^8 fewer qubits ($2^{14.1}$ against our 2^{22}). Calculating the other parameter sets in the same way, we can observe a trade-off: the Gauss-Jordan line we improve lowers gates and depth, while the matrix inversion approach lowers qubits.

Table 3: Quantum resources for the ISD attack in base-2 logarithm.

Scheme	Iterations	#Qubit	#Total gates (G)	T-depth (T_d)	Full-depth (FD)	$G \times FD$*
HQC-128	59	30	105	89	101	206
HQC-192	93	32	142	125	138	280
HQC-256	123	34	175	156	170	345
Classic McEliece-348864	72	22	106	93	100	206
Classic McEliece-460896	93	23	129	116	123	252
Classic McEliece-6688128	131	24	168	155	163	331
Classic McEliece-8192128	150	24	187	174	182	369

*: The gate-depth cost metric used to evaluate the NIST post-quantum security levels.

[5] Here, G denotes the total gate count after decomposition, including both Clifford and T.

[6] This is a per-iteration Toffoli count, not the total gate count G in Table 3. From the total T-count 2^{105} for QISD, we divide out the 2^{72} iterations and the seven T gates per Toffoli in our decomposition: $2^{105} \div 2^{72} \div 7 \approx 2^{30.2}$.

4.1 NIST Post-Quantum Security

NIST specifies security levels, each defined by the resources required to break a reference primitive (e.g., AES) [22]. This cost is measured as the product of the gate count and the full circuit depth (i.e., $G \times FD$), which NIST requires to be at least 2^{157}, 2^{221}, and 2^{285} at levels 1, 3, and 5 [22]. Table 3 shows that $G \times FD$ exceeds the threshold of the corresponding NIST security level for the listed instances, so each scheme satisfies its claimed security. The full depth exceeds the MAXDEPTH upper bound.

4.2 Physical Resources

We estimate the physical-qubit count and the runtime under the surface code [11], following the methodology of Amy et al. [2]. On the surface code each T-gate is supplied by a magic state, which we produce by Reed–Muller 15-to-1 distillation [7]. For the physical parameters, we assume a physical error rate $p_{phys} = 10^{-3}$, a per-gate error rate $p_g = 10^{-4}$, and code-cycle times of $1\,\mu$s, $10\,\mu$s, $100\,\mu$s, and 1 ms.

As an illustrative example, we derive the physical cost for HQC-128. Following [2, Algorithm 4] with p_{phys} and p_g, we obtain three distillation layers at code distances $\{35, 14, 7\}$ such that the output error rate satisfies $1/T_c$, where the total T-count (i.e., T_c) is 2^{104}. A logical qubit at code distance d is encoded into $2.5 \times 1.25\, d^2$ physical qubits. The bottom layer is at distance 7 and consists of $16 \times 15^2 = 3600$ input states. Each state is a logical qubit, so the bottom layer occupies $3600 \times (2.5 \times 1.25 \times 7^2) = 551250$ physical qubits and runs in $10 \times 7 = 70$ surface-code cycles. Computed in the same way, the second layer occupies 147000 physical qubits and runs in 140 cycles, while the top layer is completed in 350 cycles. A single distillation takes $70 + 140 + 350 = 560$ surface-code cycles, and one factory supplies $\lfloor 551250/147000 \rfloor = 3$ magic states. We need to supply $32768\,(= \lceil T_c/T_d \rceil)$ magic states within 560 cycles, so we parallelize across multiple factories. The required number of factories is $\lceil 32768/3 \rceil = 10923$, occupying $10923 \times 551250 \approx 2^{32.5}$ physical qubits.

We also account for the physical qubits of the Clifford gates. For the 2^{30} logical qubits, we take the smallest code distance d_c with $(p_{phys}/0.0125)^{(d_c+1)/2} < 1/C_g$ for $C_g = 2^{105}$ Clifford gates, giving $d_c = 57$. The Clifford part then requires $2^{30} \times 2.5 \times 1.25 \times 57^2 \approx 2^{43.3}$ physical qubits. Here, the Clifford gates are executed within the magic-state distillation time, so they do not affect the runtime. For HQC-128, with 10923 factories supplying the magic states, the runtime is $2^{89} \times 560 \times 1\,\mu\text{s} \approx 2^{53}$ years with about 2^{43} physical qubits[7]. We compute the other parameter sets in the same manner and report all results in Table 4.

[7] The physical qubits for the Clifford part dominate the total.

Table 4: Physical resource estimates for QISD using surface codes (in base-2 logarithm).

Scheme	#Physical qubits	Runtime for code cycle time			
		1 μs	10 μs	100 μs	1 ms
HQC-128	43	53	57	60	63
HQC-192	46	90	93	96	100
HQC-256	49	121	124	128	131
Classic McEliece-348864	35	57	61	64	67
Classic McEliece-460896	37	81	84	87	91
Classic McEliece-6688128	39	120	123	127	130
Classic McEliece-8192128	39	139	142	146	149

5 Conclusion

Post-quantum cryptography requires concrete resource estimates to assess its security against quantum attacks. This paper presents an improved quantum circuit for the Gauss–Jordan elimination that dominates an information-set decoding oracle, and estimates the cost of Grover-based ISD attacks on HQC and Classic McEliece.

At the elimination stage, our circuit updates only the matrix entries used by the remaining pivots and the syndrome entries needed for the final weight test. It then recovers the result vector by a parallel back-substitution on the syndrome register. Compared with [15], this reduces the qubit count by about 22%. The Toffoli count is reduced by up to about 20% compared with both [15] and [25]. The Toffoli depth remains close to that of [15], while it is about 50% lower than the closed-form Toffoli depth derived from the implementation of [25].

Combining our oracle with the Grover iteration counts from [25], we report logical costs for the evaluated HQC and Classic McEliece instances in Table 3. Under the $G \times FD$ metric, these instances exceed the thresholds for the corresponding NIST post-quantum security levels. Separately, the reported full depths exceed the MAXDEPTH bounds.

For fault-tolerant resources, we estimate the physical qubit count and runtime under a surface-code model with magic-state distillation (Table 4). As an example, HQC-128 requires about 2^{43} physical qubits and about 2^{53} years at a 1 μs code cycle under a physical error rate $p_{phys} = 10^{-3}$.

Considering more advanced ISD variants and reassessing the physical cost under lower-overhead fault-tolerant architectures remain possible directions for future work.

6 Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2019-II190033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 50%) and this work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2025-23523436, 50%).

References

1. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G., Bourges, I.: Hamming quasi-cyclic (HQC). NIST Post-Quantum Cryptography Standardization submission (2018), https://pqc-hqc.org 1, 3
2. Amy, M., Di Matteo, O., Gheorghiu, V., Mosca, M., Parent, A., Schanck, J.: Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In: Avanzi, R., Heys, H. (eds.) Selected Areas in Cryptography – SAC 2016. pp. 317–337. Springer International Publishing, Cham (2017) 9
3. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (Jun 2013). https://doi.org/10.1109/tcad.2013.2244643, http://dx.doi.org/10.1109/TCAD.2013.2244643 4
4. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory **24**(3), 384–386 (1978) 2
5. Bernstein, D.J.: Grover vs. mceliece. In: Post-Quantum Cryptography: Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings 3. pp. 73–80. Springer (2010) 2, 3
6. Bernstein, D.J., Chou, T., Lange, T., von Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Wang, W.: Classic McEliece: conservative code-based cryptography. NIST Post-Quantum Cryptography Standardization submission (2017), https://classic.mceliece.org 1, 3
7. Bravyi, S., Kitaev, A.: Universal quantum computation with ideal clifford gates and noisy ancillas. Physical Review A—Atomic, Molecular, and Optical Physics **71**(2), 022316 (2005) 9
8. Chevignard, C., Fouque, P.A., Schrottenloher, A.: Reducing the number of qubits in quantum information set decoding. In: Advances in Cryptology – ASIACRYPT 2024. Lecture Notes in Computer Science, vol. 15491, pp. 299–329. Springer (2025) 2, 8
9. Classic McEliece team: Classic McEliece: round-4 specification. https://classic.mceliece.org/mceliece-spec-20221023.pdf (2022) 3
10. Esser, A., Ramos-Calderer, S., Bellini, E., Latorre, J.I., Manzano, M.: An optimized quantum implementation of ISD on scalable quantum resources. Cryptology ePrint Archive, Paper 2021/1608 (2021), https://eprint.iacr.org/2021/1608 3

11. Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: Surface codes: Towards practical large-scale quantum computation. Physical Review A—Atomic, Molecular, and Optical Physics **86**(3), 032324 (2012) 9
12. Gidney, C.: Halving the cost of quantum addition. Quantum **2**, 74 (2018) 4
13. 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) 3
14. HQC team: HQC: Hamming quasi-cyclic — updated specification. https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (2025) 3
15. Jang, K., Oh, Y., Seo, H.: Depth-optimized quantum circuit of Gauss–Jordan elimination. Applied Sciences **14**(19), 8579 (2024) 1, 2, 5, 7, 10
16. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing grover oracles for quantum key search on aes and lowmc. In: 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 30. pp. 280–310. Springer (2020) 4
17. Kachigar, G., Tillich, J.P.: Quantum information set decoding algorithms. In: Post-Quantum Cryptography (PQCrypto 2017). LNCS, vol. 10346, pp. 69–89. Springer (2017) 3
18. Kim, H., Lim, S., Jang, K., Wang, S., Baksi, A., Chattopadhyay, A., Seo, H.: Tree-based quantum carry-save adder. Cryptology ePrint Archive, Paper 2025/626 (2025), https://eprint.iacr.org/2025/626 7
19. Kirshanova, E.: Improved quantum information set decoding. In: Post-Quantum Cryptography (PQCrypto 2018). LNCS, vol. 10786, pp. 507–527. Springer (2018) 3
20. Kliuchnikov, V., Maslov, D., Mosca, M.: Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits. IEEE Transactions on Computers **65**(1), 161–172 (2016) 4
21. Mukherjee, C.S., Maitra, S., Gaurav, V., Roy, D.: Preparing Dicke states on a quantum computer. IEEE Transactions on Quantum Engineering **1**, 1–17 (2020) 4
22. National Institute of Standards and Technology: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf (2016) 9
23. National Institute of Standards and Technology: HQC announced as a fourth round selection for standardization. https://csrc.nist.gov/news/2025/hqc-announced-as-a-4th-round-selection (2025) 1
24. Perriello, S., Barenghi, A., Pelosi, G.: A complete quantum circuit to solve the information set decoding problem. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). pp. 366–377. IEEE (2021) 2
25. Perriello, S., Barenghi, A., Pelosi, G.: Improving the efficiency of quantum circuits for information set decoding. ACM Transactions on Quantum Computing **4**(4), 1–40 (2023) 1, 2, 3, 4, 7, 8, 10
26. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962) 2, 3

Cayley Factorization Signatures via MPC-in-the-Head

Hyungrok Jo[1][0000-0001-9678-5188] and Junji Shikata[1,2][0000-0003-2861-359X]

[1] Institute of Advanced Sciences, Yokohama National University, 79-5 Tokiwadai Hodogaya-ku, Yokohama 240-8501, Japan
[2] Graduate School of Environment and Information Sciences, Yokohama National University, 79-7 Tokiwadai Hodogaya-ku, Yokohama 240-8501, Japan
jo-hyungrok-xz, shikata-junji-rb@ynu.ac.jp

Abstract. We propose a candidate post-quantum signature based on the hardness of *Cayley factorization*: given public generators $A, B \in \mathrm{SL}_n(\mathbb{F}_p)$ and $y = x(A, B)$, recover a short reduced word x. We treat this as an NP relation: the secret key is a length-L reduced word x, the public key is $y = x(A, B)$, and a signature is a Fiat–Shamir-transformed MPC-in-the-Head proof of knowledge of x, a Picnic-style scheme with the symmetric one-way function replaced by Cayley factorization. We adopt this route because the direct homomorphic-response analogue is insecure, and we make that observation precise as a standalone no-go for word responses. We give an explicit arithmetic circuit over $\mathbb{F}_p$ for the relation, a framework EUF-CMA theorem conditional on a simulation-extractable MPC-in-the-Head proof system, and a parameter analysis based on meet-in-the-middle and quantum claw-finding bounds. While the underlying assumption is not yet standard, it contributes a hardness source of a genuinely different algebraic nature to the post-quantum toolkit, and we present the scheme as a first candidate of this kind.

Keywords: Post-quantum signatures · Cayley factorization · Cayley hash functions · MPC-in-the-Head · Zero-knowledge proofs · Special linear groups

1 Introduction

A prominent line of post-quantum signatures builds a signature from a zero-knowledge proof of knowledge for a hard relation, made non-interactive by the Fiat–Shamir transform. The MPC-in-the-Head (MPCitH) framework of Ishai, Kushilevitz, Ostrovsky, and Sahai [17] turns the evaluation of essentially any one-way function into such a proof, and this is the engine behind Picnic [10, 18], syndrome-decoding signatures (SDitH) [15], and VOLE-in-the-Head [4]. The security of the resulting signature rests on two pillars: the hardness of inverting the chosen one-way function, and the knowledge soundness and zero knowledge of the proof. The choice of one-way function is therefore a design degree of freedom, and assumption diversity motivates exploring sources beyond symmetric primitives, lattices, and codes.

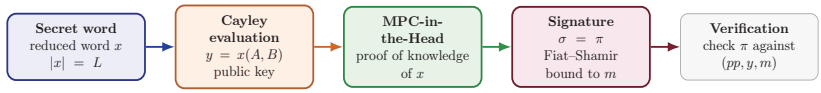


Fig. 1. Overview of the proposed signature. The public key is a Cayley evaluation $y = x(A, B)$, and a signature is a Fiat–Shamir MPC-in-the-Head proof of knowledge of the factorization word x.

Cayley factorization as a hard relation. Cayley-graph-based hash functions are an old and well-studied source of algebraic one-way candidates [25, 23, 9]. Given public generators A, B of a matrix group such as $\mathrm{SL}_n(\mathbb{F}_p)$, a word w over the alphabet $\{a, a^{-1}, b, b^{-1}\}$ is evaluated as $w(A, B)$, and recovering a short w from $y = w(A, B)$ is the *Cayley factorization* problem (equivalently, preimage resistance of the hash). Recent work of Le Coz, Battarbee, Flores, Koberda, and Kahrobaei instantiates this paradigm in $\mathrm{SL}_n(\mathbb{F}_p)$ for $n \geq 3$, where fast mixing and high girth are argued to give post-quantum preimage and collision resistance [19]. We take Cayley factorization not as a hash but as the hard relation $y = x(A, B)$, and build a signature as an MPCitH non-interactive zero-knowledge proof of knowledge of an admissible word x, with the message bound through the Fiat–Shamir hash. Structurally this is the Picnic recipe with the symmetric one-way function replaced by Cayley factorization.

A failed shortcut. The homomorphism of $\mathsf{CEval}_{A,B}$ suggests a Schnorr-like construction with a masked word response $\hat{x} = x^* X_r^{-1}$. We show that this direct approach is insecure: free reduction cancels only at seams, so long witness suffixes survive in the response and enable a polynomial-time peeling attack. This no-go result, stated in Proposition 1 and Section 5, motivates replacing word responses by a zero-knowledge proof of knowledge.

Contributions.

- **Cayley factorization as a signature relation.** We use exact-length Cayley factorization as the underlying NP relation and assume its average-case hardness for suitable parameters.
- **A no-go result for direct word responses.** We prove that the direct Schnorr-like word-response approach leaks witness suffixes under free reduction, yielding a polynomial-time peeling attack.
- **MPC-in-the-Head construction and security.** We build a Picnic-style MPC-in-the-Head signature for Cayley factorization, give the corresponding arithmetic circuit, and prove a framework EUF-CMA reduction under simulation-extractability of the proof system.

2 Related Work

Cayley hashing and its cryptanalysis. Group-theoretic hashing originates with Zémor [25] and the Tillich–Zémor (TZ) hash over $\mathrm{SL}_2(\mathbb{F}_{2^n})$ [23], was placed in

the expander-graph framework in [9], with $\mathrm{SL}_2(\mathbb{F}_p)$ navigation studied in [8]; Le Coz et al. recently moved the paradigm to $\mathrm{SL}_n(\mathbb{F}_p)$, $n \geq 3$, arguing post-quantum preimage and collision resistance from fast mixing and high girth [19], and this is the platform we adopt. The cryptanalytic history is cautionary: the TZ hash was broken [16], preimage attacks and the underlying factorization / representation / balance problems were given in [21, 22], the LPS and Morgenstern hashes were broken [20], and subexponential Gröbner-basis factoring in $\mathrm{SL}_2(\mathbb{F}_{2^n})$ was shown in [13]; Section 4.2 states why these routes do not transfer to our platform. All prior work targets a *hash*; we use the same algebraic relation as the hard problem for a *signature*.

MPCitH signatures and closest relatives. The MPCitH paradigm [17] yields Fiat–Shamir signatures for the preimage of essentially any one-way function, as in Picnic [10, 18], SDitH [15], and VOLE-in-the-Head [4], with (Q)ROM analyses in [24, 12]; ours is the same recipe with a length-L matrix-product relation in place of a fixed symmetric circuit. Cryptographic group actions likewise lack the maskable structure for a Schnorr-style response and resort to proof-of-knowledge constructions [2]; our obstruction differs in origin (the evaluation map *is* a homomorphism, but F_2 admits no blinding distribution) yet agrees in conclusion. Finally, Spinel [11] builds on the same $\mathrm{SL}_n(\mathbb{F}_p)$ hash but via the SPHINCS$^+$ framework, a complementary route from the same algebraic source.

3 Preliminaries

3.1 Signature schemes and EUF-CMA

A signature scheme is a tuple $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with $\mathsf{Setup}(1^\lambda) \to pp$, $\mathsf{KeyGen}(pp) \to (pk, sk)$, $\mathsf{Sign}(pp, sk, m) \to \sigma$, and $\mathsf{Verify}(pp, pk, m, \sigma) \to \{0, 1\}$, satisfying correctness $\Pr[\mathsf{Verify}(pp, pk, m, \mathsf{Sign}(pp, sk, m)) = 1] \geq 1 - \mathsf{negl}(\lambda)$. It is existentially unforgeable under chosen-message attacks (EUF-CMA) if no efficient adversary with pk and a signing oracle outputs (m^*, σ^*) with $\mathsf{Verify}(pp, pk, m^*, \sigma^*) = 1$ and m^* never queried, except with negligible probability. The formal experiment is standard and deferred to the full version.

3.2 NIZK proofs of knowledge and MPC-in-the-Head

A NIZKPoK for an NP relation $\mathcal{R}$ is a tuple $\Pi = (\mathsf{Prove}, \mathsf{Vrfy})$ with simulator Sim and extractor Ext, satisfying completeness, zero knowledge (per-proof advantage $\varepsilon_{\mathrm{zk}}$), and knowledge soundness. We require the *simulation-online-extractable* form, which is exactly what a reduction needs when it both simulates signatures and extracts from a forgery: a prover P^* has access to H and to a simulation oracle returning $\mathsf{Sim}(x, m)$ (programming H) on statement–message pairs of its choice; if P^* outputs an accepting (x^*, m^*, π^*) not produced by that oracle, then Ext, given P^*'s own H-queries, outputs w with $(x^*; w) \in \mathcal{R}$ without rewinding, except with probability $\varepsilon_{\text{s-ext}}$ [14, 12]. MPCitH [17] realizes such proofs for any arithmetic circuit (secret-share the witness among virtual parties, commit to their views, open a challenged subset; Fiat–Shamir makes this non-interactive

and binds the message); we treat Π abstractly through the properties above and defer per-instantiation details to the full version.

3.3 Notation and parameters

Throughout, λ is the security parameter, $n \geq 3$, and p is a large prime. Public parameters are $pp = (n, p, A, B, L, H)$, where $A, B \in \mathrm{SL}_n(\mathbb{F}_p)$ are public generators, L is a word-length parameter, and H is a hash function modeled as a random oracle and used for the Fiat–Shamir transform. We distinguish pp, which is shared by all users, from a user public key, which is a single matrix y.

3.4 Free groups and reduced words

Let $F_2 = \langle a, b \rangle$ be the free group on two generators. Elements of F_2 are reduced words over $\Sigma = \{a, a^{-1}, b, b^{-1}\}$; a word is reduced if it contains no subword of the form aa^{-1}, $a^{-1}a$, bb^{-1}, or $b^{-1}b$. We write $|w|$ for the length of a reduced word w. Multiplication in F_2 is concatenation followed by free reduction, and cancellation occurs only at the boundary between the concatenated words. The number of reduced words of length m is $4 \cdot 3^{m-1}$.

3.5 Cayley evaluation

The pair (A, B) induces a group homomorphism $\mathsf{CEval}_{A,B} \colon F_2 \to \mathrm{SL}_n(\mathbb{F}_p)$ by $a \mapsto A$, $b \mapsto B$; for $w \in F_2$ we write $w(A, B) := \mathsf{CEval}_{A,B}(w)$.
Remark 1. $\mathsf{CEval}_{A,B}$ is a group homomorphism: $\mathsf{CEval}_{A,B}(w_1 w_2) = \mathsf{CEval}_{A,B}(w_1)$ $\mathsf{CEval}_{A,B}(w_2)$. This motivates the (insecure) Schnorr-like attempt of Section 5, but plays no role in the secure construction of Section 6, where CEval only defines the public relation.

4 The Cayley Factorization Problem

4.1 Problem and relation

Definition 1 (Cayley factorization). *Let $A, B \in \mathrm{SL}_n(\mathbb{F}_p)$ be public. Given $y \in \mathrm{SL}_n(\mathbb{F}_p)$, find a reduced word $x \in F_2$ with $x(A, B) = y$.*

For cryptographic use we fix the length, which both simplifies the proof circuit and pins down the assumption.

Definition 2 (Exact-length relation). *For a length parameter L,*

$$\mathcal{R}_{\mathrm{CF}}^{=L} = \left\{ (y; x) \ : \ x \in F_2,\ x \text{ reduced},\ |x| = L,\ x(A, B) = y \right\}.$$

Assumption 1 (Exact-length Cayley factorization). *Sample $x \leftarrow \mathcal{D}_L$ uniformly from reduced words of length L and set $y = x(A, B)$. For every quantum polynomial-time algorithm $\mathcal{A}$,*

$$\Pr_{x \leftarrow \mathcal{D}_L} \left[\mathcal{A}(pp, y) = x' \ : \ |x'| = L,\ x'(A, B) = y \right] \leq \mathsf{negl}(\lambda).$$

Target	Best attack	Cost	Required L
128-bit classical	MITM	$2^{0.79L}$	≥ 162
128-bit quantum	claw-finding	$2^{0.53L}$	≥ 243
256-bit quantum	claw-finding	$2^{0.53L}$	≥ 485

Table 1. Lower bounds on the word length L, ignoring polynomial factors and assuming meet-in-the-middle / claw-finding is optimal. For a post-quantum signature the quantum rows are binding.

We emphasize that any length-L preimage breaks the assumption; the adversary need not return the sampled x. Consequently the security of the signature does not require CEval to be injective on length-L words.

Although not standard, the assumption is not an isolated novelty: preimage resistance of the hash of Le Coz et al. [19] on length-L inputs *implies* Assumption 1 (an exact-length factorization is precisely such a preimage), and exact-length factorization is in turn the length-constrained instance of the *representation problem* of Petit and Quisquater [22]. The cryptanalysis called for below thus concerns a length-restricted member of an existing problem family.

4.2 Cryptanalysis and Choice of L

Because the scheme reduces to Assumption 1, the parameters L, n, p and the generators must be justified against the best known attacks.

Meet-in-the-middle. Splitting $x = x_1x_2$ with $|x_i| = L/2$ and tabulating $\{x_1(A, B)\}$ against $\{y\, x_2(A, B)^{-1}\}$ finds a factorization in time and memory $\approx 3^{L/2} \approx 2^{0.79L}$, with comparable time at low memory via van Oorschot–Wiener collision search; quantum claw-finding (Tani / Buhrman et al.) solves the two-list problem in $\approx 3^{L/3} \approx 2^{0.53L}$. These bounds, not naive brute force, determine L as in Table 1; we are not aware of an attack beating them for well-chosen generators, but establishing this is an open task (Section 7).

Girth and saturation. Three conditions on the Cayley graph must be separated. *(i) Padding.* If a reduced u with $u(A, B) = I$ and $|u| \leq L$ exists, any short preimage of y pads to length L; since *any* length-L preimage breaks Assumption 1, $\mathrm{girth}(\{A^{\pm 1}, B^{\pm 1}\}) > L$ is a primary security parameter, not a heuristic preference. *(ii) Uniqueness.* Excluding length-L collisions would need girth $> 2L$, but this is *not* required: the reduction of Section 6.4 extracts *some* length-L preimage. *(iii) Saturation.* We require $3^L \ll |\langle A, B\rangle|$ so that the exact-length problem does not degenerate through preimage multiplicity (expected $\approx 4 \cdot 3^{L-1}/|\langle A, B\rangle| = o(1)$ under equidistribution). One caveat: the $4 \cdot 3^{g-1}$ reduced words of length up to the girth g evaluate injectively, so $g \leq (n^2-1)\log_3 p + O(1)$, and with $n = 3$, $L \geq 243$, girth $> L$ needs $\log_2 p \gtrsim 48$ (the saturation bound forces p of the same order); random pairs achieve girth $\Theta(\log p)$, but *certifying* girth $> L$ for a concrete pair means verifying the absence of relations of length $\leq L$, itself nontrivial, and is part of the open instantiation problem.

Expansion and known attacks. For one-way hardness the distribution of $w(A, B)$, $w \leftarrow \mathcal{D}_L$, should be close to uniform on $\langle A, B\rangle$, as quantified for $\mathrm{SL}_2(\mathbb{F}_p)$ by the Bourgain–Gamburd machine [7], with analogous spectral gap results to be invoked for the chosen generators; this matrix-level property is independent of the word-level seam bound of Lemma 1. As for the attacks listed in Section 2 (following the analysis of [19]): the characteristic-2 identities and subgroup structure behind [16, 21] are unavailable for odd p, the genus-0 arithmetic structure behind [20] and the triangular generators behind [13] are excluded by the criteria below, and no hidden-subgroup formulation is known for the non-abelian free F_2, leaving generic claw-finding (Table 1) as the best known quantum attack. This records why existing attacks do not *transfer*, not that no attack exists; dedicated cryptanalysis with a commitment to concrete generators remains the central open problem.

Generator criteria. A, B should generate a Zariski-dense (ideally the whole) subgroup, yield girth $> L$ ($> 2L$ if length-L uniqueness is additionally desired), satisfy $3^L \ll |\langle A, B\rangle|$, have a quantified spectral gap, and avoid common eigenvectors, simultaneous triangularizability, small invariant subspaces, and the special structures exploited in [16, 21, 20].

5 The Direct Word-Response Attempt and Its Failure

The homomorphism of CEval invites a Schnorr-like scheme. For $j \in [k]$ and $b \in \{0, 1\}$, publish $y_{j,b} = x_{j,b}(A, B)$ for secret $x_{j,b} \leftarrow \mathcal{D}_L$; to sign m, sample $x^* \leftarrow \mathcal{D}_L$, set $y^* = x^*(A, B)$ and $r = H(y^*, m) \in \{0, 1\}^k$, and output $\hat{x} = x^* X_r^{-1}$ with $X_r = x_{1,r_1} \cdots x_{k,r_k}$; the verifier checks $y^* = \hat{x}(A, B) \prod_{j=1}^{k} y_{j,r_j}$. Correctness is immediate from the homomorphism, but the scheme is insecure because $\hat{x}$ is transmitted as a *reduced word*: free reduction cancels only at seams, and almost the entire witness survives.

Lemma 1 (Seam cancellation). *Let $u, v \leftarrow \mathcal{D}_L$ be independent. The number c of letters cancelled when forming the reduced product uv satisfies $\Pr[c \geq t] \leq \frac{1}{4} 3^{-(t-1)}$ for all $t \geq 1$. In particular, over q signatures every seam cancellation is at most $C = O(\log(q/\varepsilon))$ except with probability ε.*

Proof (Sketch). Cancelling t letters forces the last t letters of u to match, in reverse order, the inverses of the first t letters of v, which occurs with probability at most $\frac{1}{4} \cdot (\frac{1}{3})^{t-1}$ by the no-backtracking constraint; a union bound over the $O(q)$ seams gives the second claim.

Proposition 1 (Suffix survival in a word response). *Let $v \in F_2$ be a reduced word of length ℓ and let $\beta \in F_2$ be an independent reduced word sampled so that each successive letter of β is near-uniform among the three non-backtracking choices; in particular the matching bound of Lemma 1 applies to the seam of $\beta \cdot v$. Let $z = \mathrm{reduce}(\beta \cdot v)$ and let c denote the number of letters cancelled at the seam. Then:*

(i) whenever $c \le \ell - t$, the length-t suffix of v appears verbatim as the length-t suffix of z;

(ii) consequently, for every $C \ge 1$, with probability at least $1 - \frac{1}{4}\,3^{-C}$ the length-$(\ell - C)$ suffix of z equals the length-$(\ell - C)$ suffix of v.

In particular, taking $C = \Theta(\log \lambda)$, the response z reveals $\ell - O(\log \lambda)$ trailing letters of the witness v except with negligible probability, so no signing procedure that transmits such a z as a reduced word can be witness-hiding.

Proof. Cancellation at the single seam of $\beta \cdot v$ removes the last c letters of β and the first c letters of v; once cancellation stops, the resulting juncture is non-cancelling and both remnants are reduced, so z is reduced and its last $\ell - c$ letters are exactly the last $\ell - c$ letters of v. This gives (i), since $c \le \ell - t$ implies $t \le \ell - c$. For (ii), Lemma 1 gives $\Pr[c \ge C+1] \le \frac{1}{4}\,3^{-C}$, and on the complementary event $c \le C$, item (i) with $t = \ell - C$ applies.

The peeling attack. Applied to $\hat{x} = x^* \, x_{k,r_k}^{-1} \cdots x_{1,r_1}^{-1}$, Proposition 1(ii) shows that, except with probability $\frac{1}{4}\,3^{-C}$, the rightmost $L - C$ letters of $\hat{x}$ reveal x_{1,r_1}^{-1} up to at most C missing prefix letters; enumerating the at most $\sum_{c=0}^{C} 4 \cdot 3^{c-1} \le 2 \cdot 3^C = \mathrm{poly}(\lambda)$ candidates against the public y_{1,r_1} recovers an effective preimage, which is peeled off via $\hat{x} \leftarrow \mathrm{reduce}(\hat{x}\,\tilde{x}_{1,r_1})$ and the argument repeated along the entire selected path. Collecting $O(\log k)$ signatures covers both values of every challenge bit (a coupon-collector bound), yielding effective preimages of all $2k$ public-key components and hence universal forgery; full details are deferred to the extended version. The attack exploits no weakness of Cayley factorization, only the transparency of the free-group normal form, and is scoped to the word-response form rather than all Schnorr-style designs: no parameter choice rescues it, which motivates replacing the word response by a zero-knowledge proof.

6 Cayley Factorization Signatures

We now give the construction. We collect the notation fixed earlier: $\mathcal{D}_L$ is the uniform distribution over reduced words of length L, $x(A,B) = \mathsf{CEval}_{A,B}(x)$ is the Cayley evaluation of x, $\mathcal{R}_{\mathrm{CF}}^{=L}$ is the exact-length Cayley factorization relation in Section 4, and H is the random oracle used by the Fiat–Shamir transform. A signature is a NIZKPoK of a Cayley factorization of the public key, with the message bound through H; the proof-system properties are those of Section 3.2.

6.1 Underlying proof system

Let $\Pi = (\mathsf{Prove}, \mathsf{Vrfy})$ be a NIZKPoK for $\mathcal{R}_{\mathrm{CF}}^{=L}$ obtained by applying the Fiat–Shamir transform to an MPC-in-the-Head proof for the circuit of Section 6.3. We require Π to be zero-knowledge, with simulator Sim that programs H, and *simulation-online-extractable*, with extractor Ext that recovers a witness from any accepting proof by inspecting the prover's random-oracle queries, even when the prover has previously observed simulated proofs on statement–message pairs

of its choice [18, 15, 4, 14]. Straight-line (online) extraction avoids rewinding and the associated multi-query forking loss, and is the cleaner route in the (quantum) random oracle model [24, 12]. We write $\mathsf{Prove}_H(pp, y, m; x)$ and $\mathsf{Vrfy}_H(pp, y, m, \pi)$ to make explicit that the message m is an input to the Fiat–Shamir hash, so that the proof, and hence the signature, is bound to m.

6.2 The scheme

Our scheme is as follows.

$\mathsf{Setup}(1^\lambda)$**:** choose n, p, L and generators A, B meeting the criteria of Section 4.2; output $pp = (n, p, A, B, L, H)$.
$\mathsf{KeyGen}(pp)$**:** sample $x \leftarrow \mathcal{D}_L$, set $y = x(A, B)$; output $sk = x$ and $pk = y$.
$\mathsf{Sign}(pp, sk, m)$**:** output $\sigma = \pi$ where $\pi \leftarrow \mathsf{Prove}_H(pp, y, m; x)$.
$\mathsf{Verify}(pp, pk, m, \sigma)$**:** accept iff $\mathsf{Vrfy}_H(pp, y, m, \pi) = 1$.

Correctness. An honest signer knows x with $(y; x) \in \mathcal{R}_{\mathrm{CF}}^{=L}$, so π verifies by correctness of Π.

6.3 MPC-in-the-Head Circuit for Cayley Factorization

We give an arithmetic circuit over $\mathbb{F}_p$ for $\mathcal{R}_{\mathrm{CF}}^{=L}$, i.e. for $x(A, B) = y$ with $|x| = L$ and x reduced.

Selector encoding Let $G_1 = A$, $G_2 = A^{-1}$, $G_3 = B$, $G_4 = B^{-1}$. For each position $i \in [L]$ introduce selectors $e_{i,1}, \ldots, e_{i,4} \in \mathbb{F}_p$ subject to

$$\text{Booleanity:} \quad e_{i,t}(e_{i,t} - 1) = 0 \quad (t \in \{1, 2, 3, 4\}), \tag{1}$$

$$\text{One-hot constraint:} \quad \textstyle\sum_{t=1}^{4} e_{i,t} = 1. \tag{2}$$

Over a field, (1) forces $e_{i,t} \in \{0, 1\}$ and then (2) forces exactly one selector to be 1. Without (1) the selectors could be an arbitrary affine combination summing to 1, and the matrix below would be an affine combination of generators, breaking soundness. The selected matrix at position i is $W_i = \sum_{t=1}^{4} e_{i,t} G_t$.

Reduced-word constraints To forbid the only cancelling adjacencies aa^{-1}, $a^{-1}a, bb^{-1}, b^{-1}b$, impose for all $i \in [L - 1]$

$$e_{i,1}e_{i+1,2} = 0, \quad e_{i,2}e_{i+1,1} = 0, \quad e_{i,3}e_{i+1,4} = 0, \quad e_{i,4}e_{i+1,3} = 0.$$

These four products are exactly the reduced-word conditions for F_2: reduced words correspond to non-backtracking walks on the 4-regular Cayley graph, so no state-dependent hash encoding is needed—the signature proves a word-evaluation relation, not a hash property.

Matrix-product constraints Set $M_0 = I_n$ and, for $i = 1, \ldots, L$, impose $M_i = M_{i-1}W_i$, with the final constraint $M_L = y$. The intermediate matrices $M_1, \ldots, M_{L-1}$ are part of the witness. Each step is a product of an $n \times n$ secret matrix and a matrix linear in the selectors, expanded entrywise as $M_i[a, b] = \sum_c M_{i-1}[a, c]\, W_i[c, b]$.

Circuit size The dominant cost is the matrix-product chain, with #multiplications $\approx L\,n^3$ field multiplications using schoolbook multiplication. The selectors G_t are public constants, so each W_i is a known-coefficient linear form and the products $M_{i-1}W_i$ have one public operand structure that can be exploited; sparse or otherwise structured generators reduce the n^3 factor. As a Picnic-style signature, the proof size grows with this circuit size and the soundness repetitions of the MPC-in-the-Head proof.

6.4 Security

Theorem 1 (EUF-CMA, conditional on the proof system). *Suppose Π is a zero-knowledge, simulation-online-extractable NIZKPoK for $\mathcal{R}_{\mathrm{CF}}^{=L}$ with simulator* Sim *and extractor* Ext*, with zero-knowledge advantage $\varepsilon_{\mathrm{zk}}$ per simulated proof and simulation-extraction failure $\varepsilon_{\text{s-ext}}$. Then for every QPT adversary $\mathcal{A}$ making q_s signing queries there is a QPT algorithm $\mathcal{B}$ against Assumption 1 with*

$$\mathsf{Adv}_{\mathcal{A}}^{\text{euf-cma}} \le \mathsf{Adv}_{\mathcal{B}}^{\text{cf}} + q_s\,\varepsilon_{\mathrm{zk}} + \varepsilon_{\text{s-ext}}.$$

The statement is a framework reduction: it isolates the three properties of Π on which security depends and is agnostic to the concrete MPC-in-the-Head instantiation, which fixes the values of $\varepsilon_{\mathrm{zk}}$ and $\varepsilon_{\text{s-ext}}$.

Proof (Sketch). **Game** G_0 is the real EUF-CMA game. **Game** G_1 replaces each honest signing proof by a simulated proof $\mathsf{Sim}(y, m)$, programming H; since Π is zero-knowledge, $|\Pr[G_1] - \Pr[G_0]| \le q_s\varepsilon_{\mathrm{zk}}$, and G_1 no longer uses sk.

Reduction. $\mathcal{B}$ receives a Cayley factorization challenge (pp, y) with $y = x(A, B)$, $x \leftarrow \mathcal{D}_L$, sets $pk := y$, and runs $\mathcal{A}$ in G_1, answering signing queries with Sim (no witness needed) and simulating H. The view is identical to G_1 because y is distributed exactly as an honest public key. When $\mathcal{A}$ outputs an accepting forgery (m^*, π^*) on a fresh message, $\mathcal{B}$ runs Ext on π^*. Since m^* was never queried to the signing oracle, the tuple (y, m^*, π^*) differs from every tuple produced via the simulation oracle, so the forgery is an admissible input to the simulation-extractability experiment: Ext returns x^* with $|x^*| = L$ and $x^*(A, B) = y$, except with probability $\varepsilon_{\text{s-ext}}$, and $\mathcal{B}$ outputs x^* as a Cayley factorization of y. Hence $\Pr[G_1] \le \mathsf{Adv}_{\mathcal{B}}^{\text{cf}} + \varepsilon_{\text{s-ext}}$, and the claim follows.

Because there is a single instance and the challenge is embedded directly as the public key, the reduction has no guessing loss: it does not need to predict which of several public-key components a forgery will use. This is a direct consequence of removing the challenge-selected key structure of the naive scheme.

Remark 2 (Instantiating the proof-system parameters). For an MPCitH proof with per-execution soundness error κ (e.g. $\kappa = 1/N$ for N parties) repeated τ times, $\varepsilon_{\text{s-ext}} \approx \kappa^\tau$ up to a q_H-dependent Fiat–Shamir factor, with τ chosen so that $\kappa^\tau \le 2^{-\lambda}$. In the classical ROM, simulation-extractability of Fiat–Shamir proofs follows from the non-malleability analysis of [14]; for commit-and-open MPCitH protocols (KKW, TCitH, VOLEitH) the straight-line extractor is unaffected by simulated proofs on distinct statement–message pairs, giving the

additive bound above without forking losses. In the QROM the corresponding guarantees are given by [12] (see also [24]). Concrete $(\kappa, \tau, \varepsilon_{\text{zk}}, \varepsilon_{\text{s-ext}})$ are deferred to a fixed instantiation in the full version.

6.5 Efficiency and Comparison

It is fair to ask why one would prove a Cayley factorization rather than a preimage of a well-studied one-way function. We do not claim efficiency; on current understanding there is none. The quantum bound of Section 4.2 forces $L \gtrsim 243$ for 128-bit post-quantum security, so the relation circuit is a sequential chain of $\approx L\, n^3 \approx 6.6 \times 10^3$ multiplications for $n = 3$, an order of magnitude larger and deeper than a symmetric circuit, and the assumption is far less mature.

To make the cost concrete rather than implicit: with a TCitH/VOLEitH-style backend ($N = 256$ parties, $\tau = 16$ repetitions) the witness comprises $\approx (L-1)\, n^2 \approx 2.2 \times 10^3$ elements of $\mathbb{F}_p$ with $\log_2 p \approx 48$ forced by the girth analysis of Section 4.2, placing the signature in the *low hundreds of kilobytes*, roughly two orders of magnitude above ML-DSA (FIPS 204, 2.4 kB) and one to two above MPCitH candidates such as Mirath or SDitH.

The motivation is assumption diversity. Existing MPC-in-the-Head signatures draw their relation from a small set of established sources: symmetric primitives in Picnic [10, 18] and FAEST [3, 4], syndrome decoding in SDitH [15] and RYDE [6], multivariate systems in MQOM [5], and MinRank in Mirath [1]. Our scheme instead proves a word factorization, navigation in the Cayley graph of a non-abelian matrix group, whose hardness is governed by expansion, a source disjoint from the code-based, multivariate, lattice, and symmetric ones. To our knowledge this is the first MPC-in-the-Head signature built on a non-abelian group-navigation problem, and that independence is its value: it hedges against progress that would weaken many candidates at once. Making the cost competitive, via structured generators that collapse the n^3 factor, is the main problem left open in Section 7. Extensions such as a bounded-length relation $|x| \leq L$, supported by a fifth identity selector with a monotone-padding constraint and selector-gated reduced-word constraints, are deferred to the full version.

7 Conclusion

We have proposed a Picnic-style Fiat–Shamir signature based on Cayley factorization. The scheme treats exact-length Cayley factorization as an NP relation and proves knowledge of a factorization word using an MPC-in-the-Head proof system. Under simulation-extractability of the proof system, our framework theorem reduces EUF-CMA security to exact-length Cayley factorization, positioning the scheme as a first candidate based on this assumption. There are some open tasks such as explicit generator selection with certified girth, dedicated cryptanalysis and parameter setting, circuit optimization, and a full QROM analysis.

Acknowledgments. This work was in part supported by JSPS KAKENHI Grant Numbers JP26K02910, JP26K14821. This work was in part supported by JST K Program Grant Number JPMJKP24U2, Japan.

References

1. Adj, G., Aragon, N., Barbero, S., Bardet, M., Bellini, E., Bidoux, L., Chi-Domínguez, J., Dyseryn, V., Esser, A., Feneuil, T., Gaborit, P., Neveu, R., Rivain, M., Rivera-Zamarripa, L., Sanna, C., Tillich, J., Verbel, J., Zweydinger, F.: Mirath (merger of MIRA/MiRitH). Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
2. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 411–439. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_14
3. Baum, C., Braun, L., Beullens, W., Delpech de Saint Guilhem, C., Klooß, M., Majenz, C., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., Roy, L., Scholl, P.: FAEST. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
4. Baum, C., Braun, L., Delpech de Saint Guilhem, C., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. Cryptology ePrint Archive, Report 2023/996 (2023), https://eprint.iacr.org/2023/996
5. Benadjila, R., Bouillaguet, C., Feneuil, T., Rivain, M.: MQOM — MQ on my Mind. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
6. Bidoux, L., Chi-Domínguez, J.J., Feneuil, T., Gaborit, P., Joux, A., Rivain, M., Vinçotte, A.: RYDE: a digital signature scheme based on rank syndrome decoding problem with MPC-in-the-head paradigm. DCC **93**(5), 1451–1486 (2025). https://doi.org/10.1007/s10623-024-01544-1
7. Bourgain, J., Gamburd, A.: Uniform expansion bounds for Cayley graphs of $\mathrm{SL}_2(\mathbb{F}_p)$. Annals of Mathematics **167**(2), 625–642 (2008)
8. Bromberg, L., Shpilrain, V., Vdovina, A.: Navigating in the cayley graph of $SL_2(F_p)$ and applications to hashing. Cryptology ePrint Archive, Report 2014/814 (2014), https://eprint.iacr.org/2014/814
9. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of Cryptology **22**(1), 93–113 (Jan 2009). https://doi.org/10.1007/s00145-007-9002-x
10. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 1825–1842. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3133997
11. Cherkaoui, A., Heravi, F., Kahrobaei, D., Shahandashti, S.F.: Spinel: A post-quantum signature scheme based on $\mathrm{SL}_n(\mathbb{F}_p)$ hashing. Cryptology ePrint Archive, Paper 2026/221 (2026), https://eprint.iacr.org/2026/221

12. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Efficient NIZKs and signatures from commit-and-open protocols in the QROM. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 729–757. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_25
13. Faugère, J.C., Perret, L., Petit, C., Renault, G.: New subexponential algorithms for factoring in $SL(2, \mathbb{F}_{2^n})$. Cryptology ePrint Archive, Report 2011/598 (2011), https://eprint.iacr.org/2011/598
14. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: Galbraith, S.D., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 60–79. Springer, Berlin, Heidelberg (Dec 2012). https://doi.org/10.1007/978-3-642-34931-7_5
15. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 541–572. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_19
16. Grassl, M., Ilic, I., Magliveras, S.S., Steinwandt, R.: Cryptanalysis of the Tillich-Zémor hash function. Journal of Cryptology **24**(1), 148–156 (Jan 2011). https://doi.org/10.1007/s00145-010-9063-0
17. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) 39th ACM STOC. pp. 21–30. ACM Press (Jun 2007). https://doi.org/10.1145/1250790.1250794
18. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 525–537. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243805
19. Le Coz, C., Battarbee, C., Flores, R., Koberda, T., Kahrobaei, D.: Post-quantum hash functions using $\mathrm{SL}_n(\mathbb{F}_p)$. Advances in Mathematics of Communications **19**(3), 996–1009 (2025), cryptology ePrint Archive, Paper 2022/896
20. Petit, C., Lauter, K., Quisquater, J.J.: Full cryptanalysis of LPS and morgenstern hash function. Cryptology ePrint Archive, Report 2008/173 (2008), https://eprint.iacr.org/2008/173
21. Petit, C., Quisquater, J.J.: Preimages for the Tillich-Zémor hash function. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 282–301. Springer, Berlin, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-19574-7_20
22. Petit, C., Quisquater, J.J.: Rubik's for cryptographers. Cryptology ePrint Archive, Report 2011/638 (2011), https://eprint.iacr.org/2011/638
23. Tillich, J.P., Zémor, G.: Hashing with SL_2. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 40–49. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_5
24. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 755–784. Springer, Berlin, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_25
25. Zémor, G.: Hash functions and graphs with large girths (rump session). In: Davies, D.W. (ed.) EUROCRYPT'91. LNCS, vol. 547, pp. 508–511. Springer, Berlin, Heidelberg (Apr 1991). https://doi.org/10.1007/3-540-46416-6_44

Channel-Aware Secure Key Transmission for PQC- and QKD-Enabled LEO Satellite Networks

Na-Hyun Kim[1[0009-0004-5514-237X]] and Il-Gu Lee[2[0000-0002-5777-4029]]

[1,2] Sungshin Women's University, Seongbuk-gu 02844, Republic of Korea
cselab.nhk@gmail.com, iglee@sungshin.ac.kr

Abstract. Low Earth Orbit (LEO) satellite networks have emerged as a key technology for enabling global connectivity in 5G (Generation mobile communications) and 6G systems, providing communication services even in geographically challenging regions. However, frequent handovers, rapidly changing channel conditions, and limited satellite visibility windows make reliable and efficient secure key delivery challenging. Moreover, although post-quantum cryptography (PQC) and quantum key distribution (QKD) provide strong security against quantum attacks, the communication overhead associated with key establishment and distribution can further degrade transmission efficiency in dynamic LEO satellite environments. In this paper, we propose a channel-aware secure key transmission scheme for PQC- and QKD-enabled LEO satellite networks. The proposed method dynamically determines the transmission path for PQC key Protocol Data Units (PDU) based on channel state information, while employing cooperative multi-ground-station architecture for QKD key distribution to improve transmission reliability. Simulation results demonstrate that the proposed methods significantly improve both key delivery reliability and transmission efficiency compared with conventional methods. At five handovers per minute, proposed 1 achieves 6× and 8× improvements, while proposed 2 maintains improvements of approximately 2x and 3x, respectively. Furthermore, both proposed methods maintained a 100% key transmission success rate and a 100% average key delivery ratio even at a packet loss rate of 50%, whereas the conventional methods experience substantial performance degradation. These results demonstrate that the proposed scheme provides an efficient and reliable secure key transmission mechanism for PQC- and QKD-enabled LEO satellite communication systems.

Keywords: Low Earth Orbit Satellite Network, Quantum Key Distribution, Handover Management, Multi-Channel Communication, Multi-ground station

1 Introduction

Recently, Low Earth Orbit (LEO) satellite communications have emerged as next-generation communication infrastructure supporting public, industrial, and military services, and are gaining attention as a key technology for realizing global connectivity in the 5G (Generation mobile communications) and 6G eras [1, 2, 3]. With large-scale

satellite constellation projects—such as SpaceX's Starlink, Amazon's Project Kuiper, and OneWeb—now in full swing, it has become possible to provide wide-area communication services even in maritime, mountainous, disaster-stricken, and border regions where existing terrestrial networks struggle to operate [4, 5, 6]. In particular, satellite communications are increasingly utilized as an essential means of communication in mission-critical environments such as national defense, disaster response, and the operation of national backbone networks, thereby continuously growing in importance as a national security and strategic asset.

However, as the strategic value of satellite communication systems increases, they are emerging as prime targets for cyberattacks. A prime example is the cyberattack on the KA-SAT satellite communications network that occurred early in the Russia-Ukraine war; this incident is considered to have disrupted thousands of satellite terminals and had a widespread impact on both military and civilian communications services [7]. This incident demonstrated that satellite communications systems are more than mere communications infrastructure—they are critical assets directly linked to national security—and served as a catalyst to further highlight the need for secure authentication and key management technologies in the satellite communications environment.

Currently, most satellite communication systems utilize AES (Advanced Encryption Standard)-based symmetric keys and public-key cryptosystems based on ECC (Elliptic Curve Cryptography) and RSA (Rivest-Shamir-Adleman) for authentication and key exchange [8]. However, with advancements in quantum computing technology raising the possibility of decrypting public-key cryptosystems using Shor's algorithm, the long-term security of existing cryptosystems is under threat [9]. Consequently, international standardization bodies, including the United States National Institute of Standards and Technology (NIST), are adopting quantum-resistant cryptography as the next-generation security technology, and research on Post-Quantum Cryptography (PQC)-based authentication and key management technologies is actively underway in the satellite communications environment as well [10, 11]. However, existing studies have primarily focused on verifying the security of PQC algorithms and designing protocols and have not sufficiently considered the communication overhead issues caused by the significantly increased sizes of public keys, signatures, and authentication messages compared to existing ECC- and RSA-based cryptosystems. In particular, in LEO satellite environments with limited link capacity and short visibility times, this overhead can lead to authentication delays, increased packet retransmissions, and degraded handover performance. Nevertheless, research on efficiently transmitting PQC-based authentication information remains insufficient.

Meanwhile, since LEO satellites orbit the Earth at a speed of approximately 7.6 km/s, their visibility time with ground users is limited to around a few minutes [12]. This results in frequent handovers, leading to persistent issues such as high channel BER (Bit Error Rate), Doppler shift, and the possibility of link disconnections [13]. In particular, the control signal exchange and authentication procedures that occur during handovers cause additional communication overhead, leading to service delays and degraded connection quality. Therefore, efficient technologies are needed to ensure stable connection continuity and service quality within a limited visibility time.

Previous studies have proposed path selection and link adaptation techniques to mitigate the problem of frequent handovers in LEO satellite environments [14, 15]. However, many of these studies rely on static path determination—which is often based on a single communication channel or fails to reflect changes in channel conditions in real time—and thus cannot sufficiently guarantee the stability of key transmission and handovers in the rapidly changing LEO satellite environment. To address these limitations, this paper proposes a channel-aware secure key transmission scheme for PQC- and QKD-enabled LEO Satellite networks. The proposed scheme continuously monitors the state of available communication channels and dynamically determines the transmission path for PQC key Protocol Data Units (PDU) according to channel state information. In addition, cooperative multi-ground-station architecture is employed for QKD key distribution, allowing quantum keys generated at multiple ground stations to be securely reconstructed at the satellite. By exploiting channel awareness during key transmission, the proposed approach minimizes packet loss and retransmissions while improving successful key delivery within limited satellite visibility windows. Consequently, both transmission reliability and handover efficiency are significantly enhanced.

2 Related Work

Table 1 provides an overview of existing research on channel characteristics and the application of handover technology in LEO satellite communications, as well as research on PQC key transmission methods and management.

Table 1. Analysis of Previous Studies

Previous studies	Ref.	Main technique	Limitation
Channel Characteristics and Handover in LEO Satellite Communication	[14]	In the SDNS (Software-Defined Network Satellites) architecture, this paper proposes a seamless handover technique in which a central controller dynamically updates the flow tables of satellite switches from the ground and manages the MDL (Main Data Link) and WDL (Weak Data Link) links in parallel to ensure a handover without interruption to the data path	Focuses solely on communication performance and service continuity, without considering certification overhead or channel stability
	[15]	Propose an intelligent CHO (Conditional Handover) technique that enables handover at the optimal time by utilizing a deep Q-network (DQN) to learn and determine the timing of	Because decision-making relies on a single channel, there is a lack of alternative strategies to respond to handover failures or sudden link degradation

		conditional handover between LEO satellites based on state information such as signal strength (CNR), elevation angle, and distance	
Key Transmission and Key Management	[16]	Proposed an SDN-based hybrid control plane that secures the ground-to-satellite link using QKD and the inter-satellite links using PQC in a QKD-enabled LEO constellation environment, and analyzed the practical key distribution workflow and latency characteristics in a mixed deployment environment	There are no metrics for communication overhead—a key evaluation metric resulting from increases in public key and signature sizes—and LEO handover is not considered
	[17]	Immutably records the satellite's encrypted local updates and RSA-signed metadata on a HAP-based, permissioned Proof-of-Authority blockchain	It does not consider vulnerabilities to quantum computing threats and lacks long-term security

3 Channel-Aware Secure Key Transmission Technique

3.1 Channel-Aware Key Transmission Framework

This study proposes a channel-aware secure key transmission framework for PQC- and QKD-enabled LEO satellites networks. Unlike conventional approaches that rely on predetermined communication paths, the proposed framework continuously monitors the state of available quantum and non-quantum communication channel and dynamically selects the transmission path for PQC key PDU according to channel state information. By adapting key transmission to rapidly changing channel conditions during frequent satellite handovers, the proposed framework improves transmission reliability while reducing packet loss and retransmission overhead.

Figure 1 illustrates the architecture of the proposed multi-channel key exchange system. The proposed method consists of a channel state assessment phase and a key transmission phase. First, during the channel state assessment phase, the satellite and the ground station periodically collect information on the quantum channel's QBER (Quantum Bit Error Rate), key generation rate, and link duration, as well as the non-quantum channel's packet loss rate, delay, and signal quality. Based on the collected information, the system assesses the usability of each channel and determines the optimal transmission method. In the subsequent key transmission phase, one of the following three transmission modes is selected based on the evaluated channel state. In Quantum-Priority Mode, the session key generated via QKD is prioritized when the quantum channel

state is favorable. Conversely, in Non-Quantum-Priority Mode, if the quantum channel quality deteriorates, PQC-based key exchange is performed to transmit the key via the non-quantum channel. Furthermore, in Hybrid Parallel Mode, when both the quantum and non-quantum channels are stable, both channels are utilized simultaneously to transmit key-related information in parallel, thereby improving the probability of successful key delivery and the overall system throughput. Conversely, when both the quantum and non-quantum channels are unstable, the key is not transmitted immediately; instead, the generated key is temporarily stored in a key fragment buffer within the ground station. Once the channel conditions stabilize, the stored key information is retransmitted, thereby minimizing key loss and the overhead of repeated retransmissions that can occur in unstable link environments.

Furthermore, this study additionally considers a key transmission structure based on multi-ground-station cooperation in conjunction with a multi-channel architecture. In a LEO satellite environment, where the time of visibility with a single ground station is limited, key transmission methods that rely on a specific ground station increase the likelihood of transmission failure. To address this, a structure is implemented in which multiple ground stations cooperate to distribute QKD- or PQC-based key PDU. Each ground station transmits a segmented key PDU to the satellite in a distributed manner, and the satellite recombines the received PDU to reconstruct the final key. This multi-ground-station cooperative structure minimizes the impact of transmission failures due to link failures involving a specific ground station or the expiration of the visibility window. Furthermore, by having multiple ground stations participate in key transmission simultaneously, the probability of successful key delivery is improved even within the limited visibility window. In conclusion, this approach eliminates dependence on a single link and enhances the reliability, availability, and resilience of the satellite-based key management system.

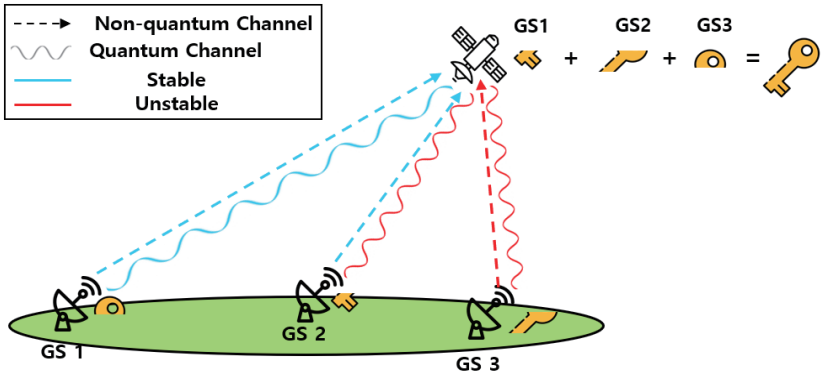


Fig. 1. System architecture of the proposed multi-channel key distribution framework

3.2 PQC Key Protocol Data Unit

Figure 2 illustrates the structure of the PQC key PDU. This study is based on the PDU structure of the OTAR (Over-The-Air Rekeying) Key Management Service defined in the CCSDS (Consultative Committee for Space Data Systems) standard and extends the existing symmetric-key-based key management system to a quantum-resistant cryptography-based structure. CCSDS OTAR is a standard key management mechanism designed with space communication environments in mind, which require limited bandwidth and high reliability; it performs key distribution and renewal through a single PDU structure containing a key identifier, an encrypted key value, and an MAC (Message Authentication Code). However, while this structure operates effectively in the current satellite communication environment, it has the limitation that it is difficult to guarantee the security of existing public-key and symmetric-key management systems in the long term if they are exposed to quantum computing-based attacks. For satellite systems performing long-term missions lasting several years or more, replacing the cryptographic system during operation is not easy; therefore, a proactive security structure prepared for quantum attacks is required. Therefore, this study proposes a new PDU structure that redesigns the internal key delivery mechanism to be PQC-based while maintaining the structure of the CCSDS OTAR Key Management PDU. The proposed PDU consists of a PDU Header and a PDU Data Field; the header includes Tag and length fields, identical to the existing CCSDS, to identify services and manage data length. The Data Field has been modified from the existing AES-based structure and now consists of Master QKD ID, Encrypted QKD ID, and Encrypted PQC Key. In the proposed structure, ML-KEM-768 is employed for PQC key establishment, and the resulting PDU size is maintained within the 552 to 4,616 bits range considered for the CCSDS OTAR PDU. Therefore, the structure proposed in this study enables secure key distribution and renewal even in a quantum computing environment by transitioning the internal cryptographic key management method to a PQC-based approach while maintaining the existing CCSDS OTAR protocol framework.

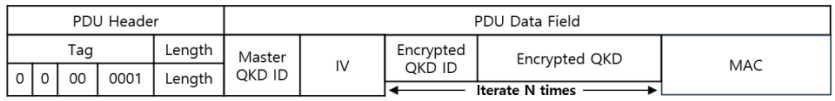


Fig. 2. System architecture of the proposed QKD-Based key Management PDU

4 Performance Evaluation and Analysis

4.1 Experiment setup

In this study, simulation-based experiments were conducted to evaluate secure key delivery between a single LEO satellite and a ground station under time-varying non-quantum and quantum channel conditions. The satellite visibility window was set to 600 seconds, and the revisit interval was set to 5,400 seconds, resulting in a total visibility cycle of 6,000 seconds. The minimum elevation angle was set to 10°, and a geometric link model was applied in which the satellite elevation angle varied over the

visibility period and the corresponding slant range was determined accordingly. The simulation was performed using 1-second time slots. The transmission payload was set to the PQC Protocol Data Unit proposed in this paper, with ML-KEM-768 employed as the key-encapsulation mechanism. The total key data size was calculated according to the CCSDS OTAR Key Management PDU specification, resulting in a total transmission volume of 947,200 bits. The channel environment was modeled by distinguishing between non-quantum and quantum channels. Transmission availability over the non-quantum channel was determined based on packet loss rate (PLR), propagation delay, and signal-to-noise ratio (SNR). Specifically, non-quantum channel transmission was considered available only when all three conditions—PLR < 5×10^{-3}, delay < 30 ms, and SNR > 6 dB—were simultaneously satisfied. For the quantum link environment, atmospheric attenuation, cloud obstruction, additional clear-sky attenuation, pointing jitter, scintillation, and background-light noise, diffraction loss and handover-induced loss were incorporated into the simulation model. Cloud obstruction was modeled as a Bernoulli random event with a probability of 0.25. When the quantum link was not blocked by clouds, the total optical loss was determined based on the instantaneous satellite geometry and the generated optical impairments. The resulting channel loss, together with background-light noise, was used to determine the received quantum-signal characteristics and the corresponding QBER. The SKR was then calculated based on the resulting quantum-link conditions, including the received signal level and QBER. Thus, QBER and SKR were not independently sampled but were jointly derived from the same instantaneous quantum-channel state. The quantum channel was considered available only when both QBER < 0.05 and SKR > 5,000 bps were satisfied. Two conventional methods were representative single-channel baselines to reflect the distinct characteristics of non-quantum and quantum communication. Non-quantum communication generally provides higher transmission availability and transmission efficiency, whereas quantum communication provides QKD-based security but is more sensitive to atmospheric and optical channel conditions and is constrained by the achievable secret key generation rate. The comparison group consisted of four methods. Conventional method 1 transmits PQC key PDU exclusively over the non-quantum channel, whereas conventional method 2 relies solely on the quantum channel for QKD-based key transmission. Proposed method 1 employs a channel-aware secure key transmission framework that dynamically selects the transmission path for PQC key PDU based on channel state information while utilizing a cooperative three-ground-station architecture for QKD key distribution. Proposed method 2 employs and the exact same operational procedure as proposed method 1 but utilizes a single ground station. All proposed methods dynamically operate in hybrid-parallel, non-quantum-priority, quantum-priority, or buffering mode according to the instantaneous availability of the two channels. All experiments were conducted 10,000 times to ensure statistical reliability.

4.2 Evaluation Results and Analysis

Figure 3 shows the key delivery completion times for 100 ML-KEM-768 instances as the number of handovers per minute increases. To reflect the high mobility of LEO

satellite links, handover events were modeled stochastically using 1-second time slots, with the probability of a handover in each slot set to N/60. At a handover rate of 1 per minute, conventional method 1 took 104 seconds and conventional method 2 took 138 seconds, respectively, to complete the delivery of all ML-KEM-768 keys. In contrast, the proposed 1 and proposed 2 completed the key delivery in 20 seconds and 58 seconds, respectively. Even when the handover rate increased to 5 per minute, the completion times of proposed 1 and proposed 2 increased only slightly to 21 and 62 seconds, demonstrating stable performance under frequent handovers. This is because the proposed method does not rely on a single channel but utilizes both non-quantum and quantum channels to deliver the key, thereby maintaining stable delivery performance even in a satellite environment characterized by high mobility and frequent handovers.

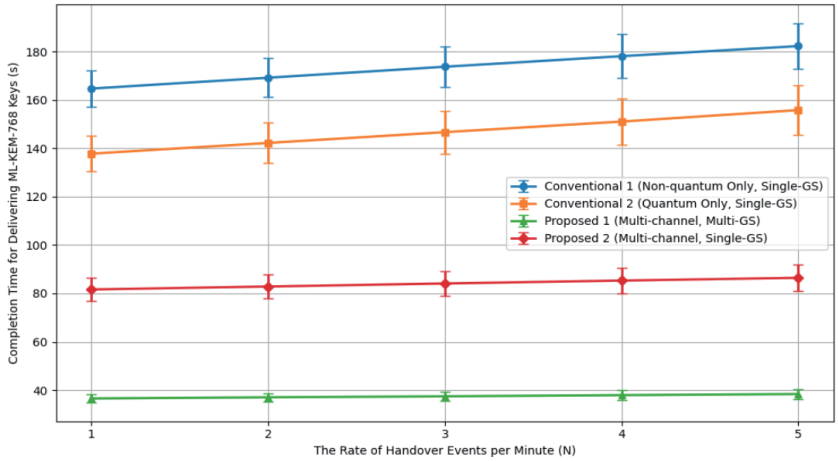


Fig. 3. Completion time for delivering ML-KEM-768 keys based on the rate of handover events per minute

Figure 4 shows the key transmission success rate as a function of packet loss rate. In this experiment, the success rate is defined as the proportion of trials in which all 100 ML-KEM-768 encapsulated keys were successfully delivered within a 180-second time constraint. Conventional method 1 maintained a success rate of 100% at packet loss rates of 0% and 10% and remained above 98% up to a 30% packet loss rate. However, its success rate decreased sharply to 43% to 40% packet loss and to 0.5% at 50. Conventional method 2 showed greater sensitivity to packet loss, with its success rate decreasing from nearly 100% at 0% packet loss to 94% at 10%, 43% at 20% and becoming nearly zero at packet loss rates of 40% or higher. In contrast, both proposed method 1 and proposed 2 consistently achieved a 100% key transmission success rate across all packet loss conditions from 0% to 50%. This improvement is attributed to the proposed methods to maintain reliable key delivery even under severe packet loss conditions by utilizing multiple transmission paths.

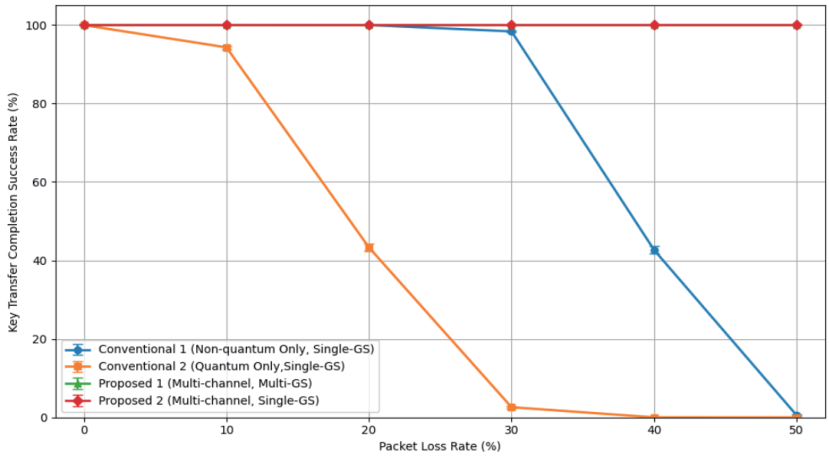


Fig. 4. Key transfer completion success rate according to the Packet Loss Rate

Figure 5 illustrates the average key delivery ratio under varying packet loss rates. The average key delivery ratio is defined as the proportion of successfully delivered key bits to the total key size, including partial deliveries under incomplete transmission. Conventional method 1 maintained a key delivery ratio of nearly 100%up to a 30% packet loss rate, but decreased to 96% at 40%, 82% at 50%. Conventional method 2 showed greater sensitivity to packet loss, with its delivery ratio decreasing from 100% to 0% packet loss to 96% at 20%, 86% at 30%, and 61% at 50%. In contrast, both proposed method 1 and proposed 2 consistently maintained a 100% average key delivery ratio across all packet loss conditions from 0% to 50%. These results demonstrate that the proposed methods provide robust key delivery performance even under severe packet loss conditions by utilizing multiple transmission paths. Specifically, even if some packets are lost during transmission, remaining key fragments can be buffered and reconstructed once channel conditions improve or alternative transmission paths become available, thereby preserving overall key delivery performance.

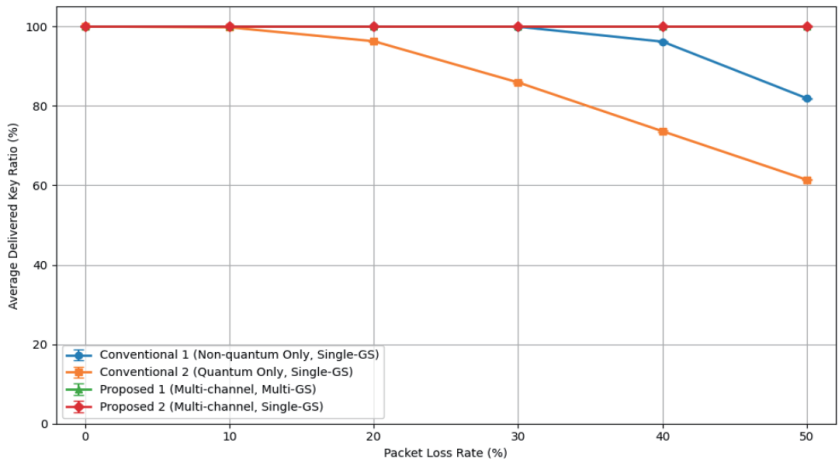


Fig. 5. Average Delivered Key Rate by Packet Loss Rate

5 Conclusion

In this study, we propose a channel-aware secure key transmission scheme for PQC- and QKD-enabled low earth orbit (LEO) satellite network to improve secure key delivery under frequent handovers and rapidly changing channel conditions. The proposed scheme dynamically determines the transmission path for PQC key Protocol Data Unit (PDU) based on channel state information while employing a cooperative multi-ground-station architecture for QKD key distribution. In addition, a new PQC key Protocol Data Unit structure suitable for PQC environments was designed based on the CCSDS OTAR Key Management Service specification. Simulation results demonstrate that the proposed methods significantly improve both transmission reliability and efficiency compared with conventional methods. At a handover rate of 1 per minute, proposed 1 reduced the key delivery completion time by approximately 5x and 7x compared with conventional 1 and conventional 2, respectively, while proposed 2 achieved improvements of approximately 2× and 3x. Even at 5 handovers per minute, proposed 1 achieved approximately 6x and 8×, while proposed 2 achieved improvements of approximately 2× and 3×, respectively. Furthermore, both proposed methods maintained a 100% key transmission success rate and a 100% average key delivery ratio even at a packet loss rate of 50%, whereas the conventional methods experienced substantial performance degradation. These results demonstrate that the proposed channel-aware transmission mechanism effectively enhances secure key delivery reliability while reducing retransmission overhead in dynamic LEO satellite communication environments. As future work, we plan to conduct a formal security analysis of the proposed multi-ground-station key distribution scheme, including threat models for compromised ground stations and the applicability of secret-sharing techniques to protect segmented key PDU. Furthermore, AI-based channel prediction and optimization

techniques will be investigated to further improve secure key transmission in dynamic LEO satellite networks.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry [grant number RS-2024-00415520] supervised by the Korea Institute for Advancement of Technology (KIAT), the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program [grant number IITP-2022-RS-2022-00156310] and National Research Foundation of Korea (NRF) grant [RS-2025-00518150], and the Information Security Core Technology Development program [grant number RS-2024-00437252] supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP), and by the Korea Information Security Industry Association (KISIA).

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

References

1. Park, J., Kim, S., Shin, W., Vincent Poor, H.: Embracing Beam-Squint Effects for Wideband LEO Satellite Communications: A 3D Rainbow Beamforming Approach. IEEE Transactions on Wireless Communications **25**, 16136–16152 (2026)
2. Hosseinian, M., Choi, J.-P., Chang, S.-H., Lee, J.: Review of 5G NTN Standards Development and Technical Challenges for Satellite Integration With the 5G Network. IEEE Aerospace and Electronics Systems Magazine **36**(8), 22-31 (2021)
3. Kim, S.-K., Kim, S.-Y., Lee, I.-G: A Secure Satellite Communication Using Shamir's Secret Sharing-Based Frequency Hopping Technique. The 9th International Conference on Mobile Internet Security (Mobisec) 2025. pp. 1-13, Sapporo, Japan (2025)
4. Hassan, N.-U., Huang, C., Yuen, C., Ahmad, A., Zhang, Y.: Dense Small Satellite Networks for Modern Terrestrial Communication Systems: Benefits, Infrastructure, and Technologies. arXiv, 1-8 (2020)
5. Su, Y., Liu, Y., Zhou, Y., Yuan, J., Cao, H., Shi, J.: Broadband LEO Satellite Communications: Architectures and Key Technologies. IEEE Wireless Communications **26**(2), 55-61 (2019)
6. Abo-zeed, M., Din, J.-B., Shayea, I., Ergen, M.: Survey on Land Mobile Satellite System: Challenges and Future Research Trends. IEEE Access **7**, 137291-137304 (2019)
7. Carlo, A., Obergfaell, K.: Cyber attacks on critical infrastructures and satellite communications. ELSEVIER **46**, 1-8 (2024)
8. Ahmad, I., Suomalainen, J., Porambage, P., Gurtov, A., Huusok, J, HÖYHTYÄ, M.: Security of Satellite-Terrestrial Communications: Challenges and Potential Solutions. IEEE Access **10**, 96038-96052 (2022)
9. Shahwar, D.-E., Imran, M., Altamimi, A.-B., Khan, W., Hussain, S., Alsaffar, M.: Quantum cryptography for future networks security: A systematic review. IEEE Access **12**, 180048-180078 (2024)
10. Yu, D.-E., Jee, S.-H., Lee, I.-G.:A Link-State-Based Intelligent Hybrid Transmission Scheme for Low Earth Orbit Satellite Communications. KCI **35**(6), 1641-1650 (2025)
11. Mattsson, J.-P., Smeets, B., Thormarker, E.: Quantum-Resistant Cryptography. arXiv, 1-45 (2021)

12. Darwish, T., Kurt, G.-K., Yanikomeroglu, H., Lamontagne, G., Bellemare, M.: Location Management in Internet Protocol-Based Future LEO Satellite Networks: A Review. IEEE Open Journal of the Communications Society **3**, 1035-1062 (2022)
13. Yeh, B.-H., Wu, J.-M., Chang, R.-Y.: Efficient Doppler Compensation for LEO Satellite Downlink OFDMA System. IEEE Transactions on Vehicular Technology **73**(12), 18863-18877 (2024)
14. Yang, B., Wu, Y., Chu, X., Song, G.: Seamless Handover in Software-Defined Satellite Networking. IEEE Communications Letters **20**(9), 1768-1771 (2016)
15. Zhang, H., Li, B.: DQN-Based Conditional Handover Algorithm for Leo Satellites Network. International Conference on Electronics Technology (ICET) 2025, pp. 203-208, Chengdu, China (2025)
16. Diaz-Bricio, A., Vidal, I., Valera, F.: Hybrid Quantum-Secure LEO Satellite Network. 2026 International Conference on Quantum Communications, Networking, and Computing (QCNC), pp. 343-347, Kobe, Japan (2026)
17. Decentralized Trust for Space AI: Blockchain-Based Federated Learning Across Multi-Vendor LEO Satellite Networks. The 13th IEEE International Conference on Big Data (IEEE BigData 2025), pp. 1-10, Macau, China (2025)

Failure-Probability Boosting on NTRU+

Gwang-Sik Kim[1], Woosang Im[1], Suseong Lee[2], and Young-Sik Kim[2]

[1] InnoCORE Research Group for Bio-Embodied Physical AI
[2] Department of Electrical Engineering and Computer Science
Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, Republic of Korea
ysk@dgist.ac.kr

Abstract. NTRU+ is a lattice key-encapsulation mechanism (KEM) whose reencryption-free Fujisaki–Okamoto decapsulation (the $\overline{\mathrm{FO}}^{\perp}$ variant, abbreviated $\mathrm{FO}^{\perp}$ below) reduces post-decryption verification to a few checks, most notably an equality on recovered randomness. We ask whether an adaptive chosen-ciphertext adversary can amplify base-decryption errors near boundary cases and convert the resulting reaction into a usable advantage. We do not introduce a novel key-recovery attack against NTRU+ or a new FO proof; instead, we give a scheme-specific case study using measurement accounting that separates internal error amplification, usable API-level acceptance, and physically observable reactions. Concretely: (i) FAB is a surrogate-guided measurement loop with rare-event interval estimation with a no-wrap margin diagnostic; (ii) $\mathrm{FO}^{\perp}$ binding makes rigid SOTP and recovered-randomness equality give exact off-language rejection in the canonical model; (iii) separating F_{dec} from a side-channel reaction L_{ext} makes the modelled observable boost decay as leakage is suppressed; and (iv) threat-axis separation gives hardening guidance with cost trends. Matched full-size NTRU+768 measurements distinguish fixed valid/tampered ciphertext pairs both inside equality and across the complete flow. Eight frozen full-flow points precede the terminal equality reference by at least 32.6 ms, while identical-input controls remain at chance. This is path localisation, not a latent-base-error oracle or fresh-ciphertext attack.

Keywords: NTRU+ · lattice KEM · decapsulation failure · reaction attack · Fujisaki–Okamoto · side-channel leakage · constant-time implicit rejection.

1 Introduction

Lattice-based KEMs are the default mechanism for post-quantum key establishment. NTRU+ pursues two engineering goals: pushing decryption correctness toward the average-case frontier through an ACWC2/SOTP-style encoder, and shrinking the post-decryption verification surface by replacing decapsulation re-encryption with a single equality on recovered randomness in an $\mathrm{FO}^{\perp}$ setting [9,12,13,15]; its final specification fixes three degrees, 768, 864, and 1152,

at $q = 3457$ [16]. Imperfect correctness interacts with chosen-ciphertext oracles: successful decapsulations guide an adversary toward rare failure regions [4], a single failure can be bootstrapped [8], and multi-target settings amplify weak advantages [7]. For NTRU, decryption-failure and reaction attacks have a long history on weakly padded variants [10,14], and a chosen-ciphertext attack against NTRU+ itself has been reported [17].

This Paper. This paper does not propose a new attack against NTRU+ or a new security proof for the Fujisaki–Okamoto transformation. Instead, we present a scheme-specific case study evaluating whether a reaction-guided failure-boosting adversary can obtain a usable advantage against NTRU+'s $\mathrm{FO}^{\perp}$ decapsulation, and detailing how that advantage should be quantified. Our answer separates internal amplification, API-level usability, and physical observability. Two difficulties drive the treatment: at full parameters the measurable quantity is the decryption *margin*, not a failure-rate ratio; and constant-time implicit rejection removes accept/reject from the returned value unless an observation channel is explicit.

Contributions.

1. **A Measurement Case Study for Failure-Probability Boosting** (Section 3). For NTRU+ we formalise surrogate-guided base-error amplification and its measurable no-wrap margin diagnostic with rare-event and risk-ratio intervals, plus exact tests [1,2,6]; the primary contribution lies in the measurement methodology, rather than a universal cross-KEM classification.
2. **Exact Off-Language Rejection** (Section 4). Using a valid-ciphertext language defined relative to the random-oracle realisation, we instantiate the $\mathrm{FO}^{\perp}$ binding (Theorem 2): ciphertexts outside the honest image accept only through an exceptional random-oracle match, so off-language wrong-key acceptance is negligible. This does not rule out honest-image correctness failures or other API attacks.
3. **Layered Threat Model and Observability Boundary** (Secs. 2, 4). We separate F_{dec} from a side-channel reaction L_{ext} and show that the modelled observable boost decays to one as leakage is suppressed [11,20]. Block-held-out measurements then test whole-flow class separation without identifying it with the latent-error channel; post-hoc localisation places the dominant fixed-input separation in randomness regeneration before equality.
4. **Threat-Axis–Separated Hardening** (Section 6). We separate reaction from fault countermeasures, summarise the spot-check false-acceptance bound and duplication condition, and report cost trends.

Empirical claims are confined to the reference model and calibrated measurements below; they support this NTRU+ reaction-boost case study, rather than a practical key-recovery attack or a universal classification of all FO KEMs.

2 Preliminaries and Threat Model

2.1 Ring, Notation, and NTRU+ Decapsulation

We work over $R_q = \mathbb{Z}_q[x]/(x^n - x^{n/2} + 1)$ of degree n and prime modulus q, with $p = 3$; final degrees are 768, 864, and 1152, with $q = 3457$ [16]. Small polynomials are sampled coefficientwise from a centred binomial distribution. NTRU+ uses $f = pf' + 1$ and a public key $h = p\,g\,f^{-1}$ with small f', g; honest encryption forms $c = hr + m$ for small r and a message-carrying m from the ACWC2/SOTP encoder [15]. We distinguish the *encoded representative* $\widetilde{m}'$ recovered inside the ciphertext from the *decoded payload* μ'. Decapsulation is deterministic:

$$\widetilde{m}' = \mathrm{Dec}(sk, c), \qquad r' = \mathrm{RRec}(pk, \widetilde{m}', c) = (c - \widetilde{m}')\,h^{-1}, \tag{1}$$

$$\mu' = \mathrm{Decode}\big(\widetilde{m}', G(r')\big), \quad (R', K') = H(\mu'),\ r'' = \mathrm{Sample}_\psi(R'), \tag{2}$$

with a decode guard requiring the recovered bits to lie in $\{0, 1\}$, and the session key is accepted iff $\mu' \neq \perp$ and $r' = r''$, an FO$^\perp$ check that replaces full re-encryption with an equality on recovered randomness [9,12,13]. Randomness recovery (1) uses $\widetilde{m}'$, not μ'. We write G, H for the hash/expansion functions, modelled as random oracles where stated.

2.2 Valid-Ciphertext Language, Predicate, and Reaction

Fix a key pair and a random-oracle realisation (G, H). Define the *valid-ciphertext language*

$$\mathcal{V}_{pk}^{G,H} = \big\{\, c = h\,r + \widetilde{m} :\ \exists\,\mu,\ (R, K) = H(\mu),\ r = \mathrm{Sample}_\psi(R),\ \widetilde{m} = \mathrm{Encode}(\mu, G(r)) \,\big\}, \tag{3}$$

the ciphertexts consistent with an honest encryption *under that realisation*. Honest encryptions lie in $\mathcal{V}_{pk}^{G,H}$ and, over key generation and encapsulation, are accepted except with the correctness error probability; our perturbations lie outside $\mathcal{V}_{pk}^{G,H}$.

For an evaluator query, fix its intended representative $\widetilde{m}$. The actual base-error event $F_{\mathrm{dec}}(c; \widetilde{m})$ is one iff $\mathrm{Dec}(sk, c) \neq \widetilde{m}$. Let $R_{\mathrm{dec}} = F_{\mathrm{guard}} \vee F_{\mathrm{eq}}$ be composite rejection; base error affects it only when a check detects its consequence. Section 4 defines $M_{\mathrm{nw}} = \mathbf{1}\{\mu(c) < 0\}$ and proves $F_{\mathrm{dec}} \leq M_{\mathrm{nw}}$.

Under constant-time implicit rejection the decapsulator returns a real key on accept and a pseudorandom key on reject within one instruction skeleton and time envelope [12,13], so R_{dec} is not an explicit output. An instrumented evaluator exposes F_{dec} or M_{nw}; an attacker observes only the KEM output or a calibrated reaction,

$$L_{\mathrm{ext}}(c) = F_{\mathrm{dec}}(c) \oplus e, \qquad \Pr[e = 1] = \varepsilon \in [0, \tfrac{1}{2}]. \tag{4}$$

This is an F_{dec}-labelled observation, not raw R_{dec} leakage; the latter requires controlling other rejection causes and calibrating its error. The equality check and

Table 1. Three boost layers. Base-error and margin ratios are distinct; physical boost requires calibrated F_{dec} sensitivity d'.

Quantity	Accessible to	Meaning
B_{dec}; diagnostic B_{margin}	evaluator	unmeasured base-error ratio versus the measured no-wrap ratio
usable boost	KEM API	off-language wrong-key acceptance; zero in the canonical rigid-SOTP model
B_{obs}	side channel	ratio of noisy external reactions based on L_{ext}; shrinks toward 1 as $d' \to 0$ or $\varepsilon \to 1/2$

re-encryption/expansion are documented FO-KEM leakage sources: the comparison leaks through timing if not constant-time [11], and re-encryption leaks through power/EM as a plaintext-checking oracle [20].

2.3 Leakage Model and Adversary

We use a standard univariate Gaussian leakage model for the targeted step [19]: one measurement is $\ell = d'\, F_{\mathrm{dec}} + \mathcal{N}(0,1)$ with sensitivity d'; the optimal single-trace error is $\Phi(-d'/2)$, and averaging N traces gives

$$\varepsilon(N) = \Phi\Big(-\tfrac{d'}{2}\sqrt{N}\Big)\,. \tag{5}$$

The sensitivity values $d' = 1.0$ and $d' = 0.02$, discussed in later sections, serve as illustrative examples for unprotected and suppressed leakage, respectively. Section 5 instead uses a matched ChipWhisperer contrast to localise fixed-input separation in an in-situ NTRU+ equality window and complete flow. It neither estimates ε nor instantiates the latent F_{dec}-labelled channel in (4). The adversary is adaptive with budget $\mathcal{B} = (q_D, m, N)$ of decapsulation queries, per-candidate evaluations, and observations per query. Timing and power/EM are examples of remote and local channel types abstracted by L_{ext}, with N denoting repeated timings or traces. We exclude chosen-key attacks and invasive faults.

3 The FAB Method

Before defining the search loop we fix terminology: Table 1 separates the internal quantity FAB can optimise, the API-level quantity usable by a remote adversary, and the physically observable quantity governed by Theorem 3.

FAB produces boundary candidates as diagnostics, not attack benchmarks. For full parameter sets, the evaluation indicates that the decryption margin is too large to sample; a reduced regime validates margin search, while observable boost needs separate calibration.

Given a baseline ciphertext $c^\star$ with a known intended representative, FAB seeks a perturbation that increases the actual base-error probability, $\Delta^\star \in \arg\max_{\Delta\in\mathcal{D}} p_{\mathrm{dec}}(c^\star + \Delta)$ with $p_{\mathrm{dec}}(c) = \Pr[F_{\mathrm{dec}}(c) = 1]$, over a trust region $\mathcal{D} = \{\|\Delta\|_\infty \le \rho\}$ of small-norm structured moves $\Delta \in \{\alpha x_i, \alpha(x_i - x_j), h \cdot s\}$, $\alpha \in \{-2, -1, 1, 2\}$, $s \leftarrow \psi_1^n$, with feature vector $\phi(\Delta) = (\{\langle e_i, \Delta\rangle\}_{i\in S_1}, \{\langle e_i - e_j, \Delta\rangle\}_{(i,j)\in S_2}, \langle h, \Delta\rangle)$. The local target-event probability is modelled by a logistic surrogate $\tilde{p} = \sigma(\theta_0 + \langle\theta, \phi(\Delta)\rangle)$ fitted by ridge-penalised maximum likelihood, and the trust region is explored by simultaneous-perturbation stochastic approximation [18]: with Rademacher u_t, schedules a_t, c_t, and $Z_t^\pm = Z(c^\star + \Delta_t \pm c_t u_t) \in \{0, 1\}$,

$$\Delta_{t+1} = \Pi_{\mathcal{D}}\left(\Delta_t + a_t \frac{Z_t^+ - Z_t^-}{2c_t} u_t\right). \tag{6}$$

The raw two-sided difference lies in $\{-1, 0, +1\}$; the surrogate and mini-batch averaging supply the smooth signal a single query lacks. Here $Z = F_{\mathrm{dec}}$ in evaluator mode and $Z = L_{\mathrm{ext}}$ in attacker mode; when actual errors are infeasible to sample, the evaluator may instead use the explicitly labelled diagnostic $Z = M_{\mathrm{nw}}$. We report the boost $B = \hat{p}_T/\hat{p}_0$ with Clopper–Pearson/Jeffreys intervals per rate and a Katz log-risk-ratio interval for B [1,6]; significance uses Fisher's exact test with Benjamini–Hochberg control [2], and a zero count is reported as a rule-of-three bound. Within the internal layer, B_{dec} and diagnostic B_{margin} remain distinct; usable boost and B_{obs} are the API and physical layers. Although $\Pr[F_{\mathrm{dec}}] \le \Pr[M_{\mathrm{nw}}]$ pointwise, their ratio need not bound B_{dec}; only B_{obs} depends on reaction-channel hardening.

4 Formal Results

The integer decryption noise of a ciphertext is $V = p\,g\,r + m\,f \in R$ (lifted to $\mathbb{Z}$ and centred).

Theorem 1 (Sufficient No-Wrap Margin). *With the centred reduction into $\{-\lfloor q/2\rfloor, \ldots, \lfloor q/2\rfloor\}$ (q odd), base decryption returns the correct message whenever $\|V\|_\infty \le \lfloor q/2\rfloor$. Define the no-wrap margin $\mu(c) = \lfloor q/2\rfloor - \|V\|_\infty$; $\mu(c) < 0$ indicates that this sufficiency certificate is lost, which does not inherently imply a decryption failure resulting from multiple modular wrap-arounds. A no-wrap violation is $M_{\mathrm{nw}}(c) = \mathbf{1}\{\mu(c) < 0\}$; by contraposition of the sufficient condition, $F_{\mathrm{dec}}(c) \le M_{\mathrm{nw}}(c)$ for evaluator queries with a known intended representative. A perturbation Δ shifts the noise by Δf: $V_\Delta = V + \Delta f$.*

Proof. Decryption computes $a = cf \bmod q$; with $c = hr + m$ and $hf = pg$, $a \equiv pgr + mf = V$. If every $|V_i| \le \lfloor q/2\rfloor$, the centred representative equals V; reducing mod p then gives m since $f \equiv 1 \pmod p$. Outside this interval the representative differs by a multiple of q, which can occasionally remain congruent to m modulo p, hence no converse is claimed. Finally $(c + \Delta)f = cf + \Delta f$.

Lemma 1 (Infeasibility of Direct Measurement). *If the n coefficients of V are independent, zero-mean, and σ-sub-Gaussian, then*

$$\Pr[\textit{no-wrap violation}] \;\le\; 2n\exp\big(-(\lfloor q/2\rfloor)^2/(2\sigma^2)\big), \tag{7}$$

which also upper-bounds base-decryption failure. Under this model the honest NTRU+ baseline is negligible; any feasible budget q_D has expected violation count far below one. Direct baseline-error measurement is therefore unavailable, so FAB uses the margin diagnostic.

Proof. A union bound over n sub-Gaussian coordinates at $t = \lfloor q/2 \rfloor$ gives the tail; the expected no-wrap-violation count in q_D honest queries is the product of q_D and this bound, and Markov bounds the chance of even one.

Theorem 2 (Exact off-language rejection under rigid SOTP). *Fix a key pair (pk, sk) and an oracle realization (G, H_{pk}), where $H_{pk}(\mu) := H(\mu, F(pk))$. Assume that non-canonical ciphertext encodings are rejected and that the SOTP is rigid. Then, for every canonical ciphertext c,*

$$\mathsf{Accept}(sk, c) = 1 \quad \Longrightarrow \quad c \in \mathcal{V}_{pk}^{G,H_{pk}}.$$

Consequently,

$$c \notin \mathcal{V}_{pk}^{G,H_{pk}} \quad \Longrightarrow \quad \mathsf{Accept}(sk, c) = 0.$$

Proof. Let

$$\widetilde{m}' = \mathsf{Dec}(sk, c), \qquad r' = \mathsf{RRec}(pk, \widetilde{m}', c),$$

and let

$$\mu' = \mathsf{Decode}(\widetilde{m}', G(r')), \qquad (R', K') = H_{pk}(\mu'), \qquad r'' = \mathsf{Sample}_{\psi}(R').$$

If c is accepted, then $\mu' \neq \perp$ and $r' = r''$. By SOTP rigidity,

$$\widetilde{m}' = \mathsf{Encode}(\mu', G(r')).$$

Moreover, the definition of RRec gives

$$c = hr' + \widetilde{m}'.$$

Since $r' = r'' = \mathsf{Sample}_{\psi}(R')$ and $(R', K') = H_{pk}(\mu')$, the pair (μ', r') witnesses $c \in \mathcal{V}_{pk}^{G,H_{pk}}$.

For the NTRU+ sampler, $\psi = \psi_1$ is coefficientwise centred binomial: each coefficient is $b - b'$ for independent uniform bits $b, b' \in \{0, 1\}$ [16]. Hence $\Pr[X = 0] = 1/2$ and the per-coefficient min-entropy is $H_\infty^{(1)} = 1$ bit. Since H supplies seeds long enough for Sample_ψ to realise ψ_1^n, whose coefficients are independent, the output-distribution min-entropy used in Theorem 2 is $H_\infty^{(n)} = n$. For the parameter sets considered here this gives $H_\infty^{(n)} = 768$, 864, or 1152 bits, so the factor $(q_H + q_G + 1)2^{-H_\infty^{(n)}}$ remains negligible for any feasible random-oracle query budget.

For the off-language perturbations tested here, Theorem 2 explains why composite rejection does not become API-level wrong-key acceptance except through

the bounded exceptional match. The residual question is whether a *physical* measurement separates the resulting execution classes.

The theorem does not block classical failure attacks using honest-image ciphertexts. At deployed NTRU+ parameters their feasibility is governed by the margin and correctness bound above; Theorem 2 only rules out converting off-language FAB perturbations into API-level wrong-key acceptances.

Theorem 3 (Observable-Boost Collapse). *Let the F_{dec}-labelled side channel satisfy* (4) *with $\varepsilon = \Phi(-\frac{d'}{2}\sqrt{N})$. The observable objective and boost are*

$$J(\Delta) = \varepsilon + (1-2\varepsilon)p_{\mathrm{dec}}(\Delta),$$
$$B_{\mathrm{obs}} = \frac{\varepsilon + (1-2\varepsilon)p_T}{\varepsilon + (1-2\varepsilon)p_0}. \tag{8}$$

Here p_0 and p_T are the baseline and terminal probabilities of F_{dec}. Hence (i) $B_{\mathrm{obs}} \to 1$ as $\varepsilon \to \frac{1}{2}$; (ii) the de-biased estimator $\hat{p} = (\hat{J} - \varepsilon)/(1-2\varepsilon)$ is unbiased for $\varepsilon < \frac{1}{2}$; and (iii) any single-query distinguisher between the baseline and boosted regimes has advantage $\leq \frac{1}{2}|1-2\varepsilon|$, which $\to 0$ as $d' \to 0$.

Proof. Flipping each Bernoulli(p_{dec}) bit with probability ε gives $\Pr[L_{\mathrm{ext}} = 1] = \varepsilon + (1-2\varepsilon)p_{\mathrm{dec}}$, hence J and B_{obs}; (i)–(ii) are immediate, and (iii) is half the total-variation distance between two Bernoulli laws, $\frac{1}{2}|(1-2\varepsilon)(p_T - p_0)|$.

5 Implementation and Evaluation

Scope. Experimental results are obtained from an NTRU+ FO$^{\perp}$ reference model, the Gaussian model of Section 2.3, and matched ChipWhisperer measurements of an instrumented full-size NTRU+768 equality window and complete flow. The physical result provides temporal hold-out on one hardware setup and parameter set, not hardware or parameter replication. All final parameter sets receive known-key margin measurement; margin-proxy boosting and FO$^{\perp}$ rejection are demonstrated only in a weakened, measurable regime.

Full-Parameter Margins. Table 2 reports known-key margins for the three final parameter sets (12,000 honest ciphertexts each). No failure or no-wrap violation occurs. The last column is a *fitted sub-Gaussian diagnostic tail* from per-coordinate σ (Lemma 1); these diagnostics are separate from parameter correctness estimates. Official $\log_2 \delta'$ values are -379, -340, and -260 for the three degrees [16]; our diagnostics are looser but likewise far below any feasible budget, so boost is not directly measurable at full parameters.

FAB Margin-Proxy Boosting in a Measurable Regime. Table 3 reports no-wrap diagnostic boosting where the adversary controls the plaintext, at equal search budget, over twelve seeds. Surrogate-guided FAB exceeds blind random search, the margin-oracle upper bound is far above both, and the signal-free control is not significant; the surrogate is weakly but measurably predictive (mean ROC AUC 0.545 vs 0.501 under shuffling). This validates the loop where ground truth is observable.

Table 2. Known-key margin distribution at the final NTRU+ parameters ($q = 3457$), reference model, 12,000 honest ciphertexts per n. The last column is a fitted diagnostic tail, not a correctness estimate; official $\log_2 \delta' = (-379, -340, -260)$ [16].

n	$\min \mu/\lfloor q/2 \rfloor$	no-wrap	σ	diag. tail $\log_2$
768	0.76	0/12000	82.3	−308
864	0.75	0/12000	80.3	−324
1152	0.71	0/12000	96.0	−223

Table 3. FAB no-wrap margin-proxy boosting in a reduced measurable regime ($n = 64$, $q = 149$), pooled over 12 seeds; honest baseline over 1.8×10^6 trials. Diagnostic ratio B_{margin} with 95% log-risk-ratio interval. These are no-wrap violations, not observed base-decryption errors.

Method	no-wrap violation	B_{margin}	95% interval
honest baseline	3.7%	1.00	—
random search	21.0%	5.69	$[5.44, 5.96]$
surrogate-guided (FAB)	25.0%	6.79	$[6.52, 7.07]$
margin-oracle (upper bound)	67.0%	18.16	$[17.84, 18.49]$
signal-free control	4.0%	1.07	$[0.96, 1.20]$ (n.s.)

Exact rejection under the recovered-randomness check. Table 4 illustrates the mechanism of Theorem 2 in a separate 12-seed run. For honest plaintexts no-wrap-violation and equality-mismatch rates coincide; for non-FO-consistent plaintexts equality mismatch fires on every query. No wrong-key acceptance is observed with equality enabled. With equality disabled, the pooled descriptive adversarial/honest wrong-key-acceptance risk ratio is 2.20 ($[1.84, 2.56]$). Message search shows no exploitable structure (mean AUC 0.512), and the guard alone catches only ≈60% of the tested invalid cases, so equality is the essential residual rejection check in this experiment.

Thus equality retains a residual API-binding role in this separate reduced run; the deployed verdict still follows from correctness margins and bounds.

Modelled Observability and the Hold-Out Requirement. In the illustrative Gaussian model ($N = 30$, 12 seeds), closed-form, search, and hold-out results agree at $d' = 1.0$; at $d' = 0.02$, selection mildly inflates the near-chance search estimate.

Physical Measurements. Using a CW308 STM32F4 and Husky Plus, four KATs compare valid ciphertexts with a one-bit tamper at byte 0 or 575, over full flow and an instrumented `verify` window. Each of three reset-separated acquisition blocks has eight treatment and four A/A contrasts, with 100 flow and 300 equality traces/class (9,600 per block; 28,800 overall). These blocks provide temporal hold-out, not hardware replication. Orders are pseudorandom; Block 0 fixes POIs, directions, and thresholds; Blocks 1–2 are held out. Trigger-high duration is class-independent.

Table 4. Effect of the $FO^{\perp}$ equality check in a separate 12-seed reduced-regime run (means).

Plaintext	no-wrap viol.	eq.-mismatch	wrong-key, eq. on	wrong-key, eq. off
honest	3.4%	3.4%	0%	1.3%
adversarial	7.2%	100%	0%	2.9%

At $N = 1$, held-out full-flow accuracy is 1600/1600 in both blocks. Equality boundary POIs give 4782/4800 and 4789/4800; seven occur 1.125 μs after the trigger and enclose pre-`verify` instrumentation, so we do not attribute them to the comparison body. Excluding 512 μs per edge on Block 0 gives held-out fixed-input separation $|d'_{\text{fix}}| = 3.83/3.63$ and accuracy .985/.989. Across 8–512 μs exclusions, $|d'_{\text{fix}}|$ stays 3.63–4.15; the identical 8/32 μs POIs are at least 54.5 μs inside. All disjoint $N = 30$ treatment batches classify correctly (96/96 flow; 320/320 equality). Descriptive $N = 30$ A/A accuracies are $22/48 = 0.458$ and $83/160 = 0.519$. Fixed-vs-fixed POIs have three positive and five negative signs, precluding one global direction; each condition's direction agrees in both held-out blocks.

The frozen full-flow POIs lie at 122.191–123.115 ms. From the 157.462-ms complete trigger and separately measured 1.735-ms equality trigger, a conservative terminal reference starts at 155.728 ms, at least 32.613 ms later. A cycle-counted diagnostic places seven POIs in `poly_cbd1(r1)` and one at the following `poly_ntt(r1)` entry. `poly_cbd1` samples regenerated randomness from `H(msg)` before equality: dominant separation occurs during the generation of the compared value, although the equality check itself also exhibits non-negligible side-channel emissions. The final flow bin also contains fail-dependent key selection and is not equality-only. Figure 1 gives this post-hoc code localisation; it does not preregister or semantically identify a failure-dependent channel.

Different ciphertexts may affect all upstream processing; this localisation is neither a failure oracle nor an estimate of ε or B_{obs}.

Table 5. Matched in-situ held-out POIs: Block 1/2 mean $|d'_{\text{fix}}|$ and $N = 1$ accuracy. Single-trace values at Block-0-frozen POIs are averaged in absolute value over eight treatment contrasts (four A/A controls); d'_{fix} is not the F_{dec} sensitivity in (5). *Post-hoc interior fit: 512 μs excluded at each edge, Block 0 refit only, with held-out evaluation unchanged.

Measurement	evaluation data	$\lvert d'_{\text{fix}}\rvert$	accuracy
NTRU+768 flow	Blocks 1/2	58.8/75.9	1.000/1.000
NTRU+768 eq. interior*	Blocks 1/2	3.83/3.63	.985/.989
NTRU+768 flow A/A	Blocks 1/2	.177/.099	.505/.491
NTRU+768 eq. interior A/A*	Blocks 1/2	.055/.079	.503/.499

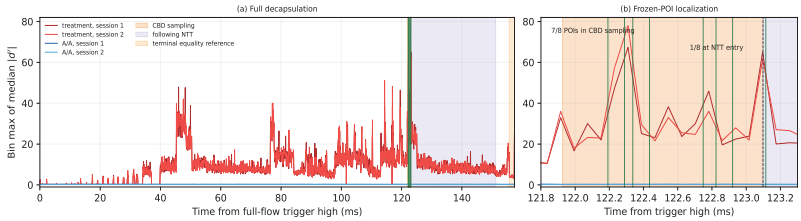


Fig. 1. Post-hoc held-out localisation. (a) Full-flow profile and terminal equality reference. (b) Seven full-flow POIs fall in `poly_cbd1(r1)` and one at the following NTT entry. This identifies code location, not a latent-base-error channel.

Table 6. Threat axes, literature-supported effects, and reference-model costs. Reaction mechanisms affect B_{obs} only when covering the observed path.

Mechanism	Axis	Coverage/effect	cost trend
constant-time implicit rejection	reaction	hides explicit API rejection	baseline
constant-time comparison	reaction	equality-local timing only	≈0
masked equality; randomized shuffling	reaction	equality-local; CBD/NTT uncovered	impl.-dep.
masked/CT re-encryption	reaction	N/A to re-encryption-free NTRU+	N/A
spot re-encryption	fault	detects injected corruption	+43%
check duplication	fault	transient faults only	≈0

6 Hardening: Threat-Axis Separation and Costs

Table 6 separates hardening axes. In the model, reaction mechanisms reduce B_{obs} by driving ε toward $\frac{1}{2}$ when they cover the complete observed path. Implicit rejection hides explicit rejection [12,13]; constant-time comparison closes timing leakage [11]; masking/shuffling can lower comparison leakage [3]. Protected re-encryption applies to FO KEMs that use it [20], not re-encryption-free NTRU+. For this fixed-input contrast, stronger separation occurs earlier; equality-only protection does not cover that CBD/NTT path and cannot bound the integrated trace. This localisation is not adversary advantage or empirical B_{obs}, and does not establish F_{dec}-labelled leakage in CBD/NTT.

Fault countermeasures address separate active corruption [5]. A fault may skip or corrupt a check, but does not create the reaction modelled in Theorem 3.

Reference-model branchless key selection, $K_{\text{out}} = (1-b)K_{\text{rand}} + bK_{\star}$ with $b = b_{\text{inv}} \wedge b_{\text{eq}} \wedge b_{\text{arith}}$, shares one accept/reject skeleton. The equality check and duplication incur an execution overhead of less than 5%, whereas the NTT/RRec self-check introduces a 43% overhead. Combining all countermeasures results in a 51% overall overhead (evaluated over 400 software decapsulations).

Proposition 1 (Spot re-encryption, exact). *If a corruption affects $t = \lceil \chi\ell \rceil$ of ℓ positions and the verifier re-encrypts a uniform k-subset without replacement, the false-acceptance probability is $\binom{\ell-t}{k}/\binom{\ell}{k} \le (1-\chi)^k$; hence $k \ge \lceil \ln(1/\eta)/\ln(1/(1-\chi)) \rceil$ meets target η.*

Proof. False acceptance occurs iff all sampled positions miss the corrupted set, whose probability is at most $((\ell - t)/\ell)^k = (1-\chi)^k$. Solving for k gives the stated threshold.

Proposition 2 (Independent-fault duplication). *Running a check R times with independent transient-bypass probability π gives all-bypass probability π^R; if the bypass is data-deterministic, the same event repeats and the probability remains π.*

7 Related Work and Conclusion

Decryption-failure and reaction attacks on NTRU date back to weakly padded variants [10,14]; failure boosting sharpened how such failures are found and amplified [4,7,8]. NTRU+ chosen-ciphertext key-recovery [17] targets the session key, which differs from the off-language rejection scenario formalized in Theorem 2; we do not explore that specific key-recovery vector in this work. Relevant FO-KEM leakage is documented in [3,11,20]; our verdict is NTRU+-specific.

We separated actual base-decryption error, API acceptance, and physical reaction for NTRU+ failure boosting. No-wrap margins provide a diagnostic rather than an error-rate measurement; together with published correctness bounds they settle the deployed feasibility question. The $\mathrm{FO}^{\perp}$ transform ensures that off-language internal acceptance is strictly excluded for the perturbations considered in this work. On full-size NTRU+768, held-out measurements show strong whole-flow valid-vs-tampered class separation, but do not identify a latent base-error oracle or adversary advantage. Under the same fixed-input contrast, the equality window is distinguishable, while all frozen dominant full-flow points occur at least 32.6 ms before the terminal equality reference. Equality-local protection therefore cannot be assumed to bound the integrated path; a claim about reaction observability still requires a matched latent-event design.

Acknowledgments. This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2026-25528270, DevOps-based Post Quantum Cryptography Switching Automation Technology and Open Platform Technology, 50%) and the Korea Planning & Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korean government (MOTIE) (No. RS-2025-02634277, Development of Human-AI Teaming-based Autonomous Security System for Future Cyber Threats, 50%).

References

1. Agresti, A.: Categorical Data Analysis. Wiley, 2nd edn. (2002)
2. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological) (1995)
3. Bhasin, S., D'Anvers, J.P., Heinz, D., Pöppelmann, T., Beirendonck, M.V.: Attacking and defending masked polynomial comparison for lattice-based cryptography. IACR Transactions on Cryptographic Hardware and Embedded Systems **2021**(3), 334–359 (2021)
4. Bindel, N., Schanck, J.M.: Decryption failure is more likely after success. In: Post-Quantum Cryptography (PQCrypto). LNCS, vol. 12100, pp. 206–225 (2020)
5. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults. In: Advances in Cryptology – EUROCRYPT. LNCS, vol. 1233, pp. 37–51 (1997). https://doi.org/10.1007/3-540-69053-0_18
6. Brown, L.D., Cai, T.T., DasGupta, A.: Interval estimation for a binomial proportion. Statistical Science (2001)
7. D'Anvers, J., Batsleer, A., Vercauteren, F.: Multitarget decryption failure attacks and their application to saber and kyber. In: Public Key Cryptography (PKC) (2022). https://doi.org/10.1007/978-3-030-97121-2_1
8. D'Anvers, J., Rossi, A., Virdia, M., Vercauteren, F.: (one) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In: Public Key Cryptography (PKC) (2020)
9. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Public Key Cryptography (PKC) (1999)
10. Gama, N., Nguyen, P.Q.: New chosen-ciphertext attacks on NTRU. In: Public Key Cryptography (PKC). LNCS, vol. 4450, pp. 89–106 (2007)
11. Guo, Q., Johansson, T., Nilsson, A.: A key-recovery timing attack on post-quantum primitives using the Fujisaki–Okamoto transformation and its application on FrodoKEM. In: Advances in Cryptology – CRYPTO. LNCS, vol. 12171, pp. 359–386 (2020)
12. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki–okamoto transformation. In: Theory of Cryptography Conference (TCC) (2017). https://doi.org/10.1007/978-3-319-70500-2_12
13. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the fo transform. In: ASIACRYPT (2022)
14. Jaulmes, É., Joux, A.: A chosen-ciphertext attack against NTRU. In: Advances in Cryptology – CRYPTO. LNCS, vol. 1880, pp. 20–35 (2000)
15. Kim, J., Park, J.H.: NTRU+: Compact construction of NTRU using simple encoding method. KpqC competition submission (and Cryptology ePrint Archive) (2022)
16. KpqC NTRU+ Team: NTRU+: Final specification. KpqC Competition, Final Specification Documents (2026), parameter sets $n \in \{768, 864, 1152\}$, $q = 3457$
17. Lee, J., Lee, M., Ryu, H., Park, J.: A novel CCA attack for NTRU+ KEM. Cryptology ePrint Archive, Paper 2023/1188 (2023)
18. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control (1992)
19. Standaert, F.X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Advances in Cryptology – EUROCRYPT. LNCS (2009)

20. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022**(1), 296–322 (2022)

Addressing the Zero-Correlation Issue and Enabling Cost-Based Key Enumeration for ZV-FA against ML-DSA

Riku Kuratani[1], Daiki Miyahara[1], Kazuo Sakiyama[1], and Yang Li[1]

[1]The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
Phone/FAX: +81-42-443-5000
E-mail: {kuratani, miyahara, sakiyama, liyang}@uec.ac.jp

Abstract. The Zero-Value Filtering Attack (ZV-FA) is a non-profiled CPA-based attack against incomplete-NTT implementations of ML-DSA. We identify a zero-correlation issue: when a targeted incomplete-NTT secret coefficient is 0 mod q, the Pearson correlation for the correct hypothesis is undefined, causing the zero-valued candidate to be ranked low in our reimplementation. We address this issue by using the rank-1 filtering score to prioritize the candidate value 0. We also apply score-based key enumeration to the filtering output using a heuristic relative score-drop cost and inverse incomplete-NTT verification. Simulated leakage experiments show that sufficiently accurate filtering can enable key recovery without the original ZV-FA scoring step, with greater benefits for zero-present keys. Practical invocation criteria remain future work.

Keywords: Side-channel attack · ML-DSA · Dilithium · incomplete NTT · zero-value filtering · cost-based key enumeration · CPA.

1 Introduction

The development of large-scale quantum computers threatens the discrete-logarithm and integer-factorization assumptions that underlie most of today's deployed public-key cryptography [1]. In response, the U.S. National Institute of Standards and Technology (NIST) has driven a multi-year standardization process for post-quantum cryptography (PQC), culminating in the selection of ML-DSA (originally Dilithium) [2–4] as the lattice-based digital-signature standard in 2024.

Beyond mathematical security, the implementation security of ML-DSA on resource-constrained embedded targets must be carefully assessed. ARM Cortex-M4-class devices are an important deployment platform, and the pqm4 library [5] ships highly optimized implementations of ML-DSA that rely on the *incomplete* number-theoretic transform (incomplete NTT) [6, 7]. The incomplete NTT stops the butterfly layers before full decomposition, leaving low-degree components whose products are computed by schoolbook multiplication. These intermediates are the attack target considered in this paper.

Side-channel attacks on Dilithium and related lattice-based implementations have also been studied in several settings [8–11]. Tosun and Savas [12] observed that incomplete-NTT-based schoolbook multiplications create an exploitable side-channel surface. Their Zero-Value Filtering Attack (ZV-FA) is a non-profiled CPA-based key-recovery attack. In its *filtering* stage, ZV-FA keeps traces where one known coefficient of the pairwise product is $0 \bmod q$, reducing a q^2-sized search per pair to two q-sized searches. In its *scoring* stage, it re-evaluates top-ranked candidate pairs using all traces and increases the search depth d whenever inverse-NTT verification fails (Section 2.3).

Problem. In this paper, we identify a structural problem in ZV-FA which we call the *zero-correlation issue*. In the multiplication targeted by ZV-FA, the intermediate value $r_{[i,j]}$ equals 0 identically whenever the true incomplete-NTT secret coefficient is 0 modulo the carrier prime, regardless of the known coefficient. Consequently, the hypothetical leakage for the correct hypothesis becomes constant, and the Pearson correlation is undefined. The original ZV-FA does not specify how to handle this case. In our reimplementation, we assign the undefined correlation a score of 0, which tends to push the true value toward the lower part of the rank list. In the scoring stage the same problem arises for pairs $(0, 0)$, in which case the scoring CPA cannot assign a meaningful score to the true pair, so the search must explore further before the key is recovered and may fail to identify the pair.

Contributions. Starting from the original ZV-FA filtering output, this paper makes three main contributions. First, we identify the zero-correlation issue of ZV-FA and show that it arises in both the filtering and scoring stages. We also show that the resulting low rank-1 filtering score in our reimplementation can be used to prioritize the candidate value 0 before the original ZV-FA scoring step (Section 3). Second, we apply score-based key enumeration to the ZV-FA filtering output without invoking the original scoring step. We transform each filtering score into a heuristic relative score-drop cost and use a priority-queue procedure to test candidate combinations from lower to higher total cost, while a one-directional rule avoids duplicates(Section 4). Third, we evaluate the algorithm on simulated leakage traces over many random keys. The results show cases where the key can be recovered without running the original ZV-FA scoring step, especially when the filtering output is sufficiently accurate (Sections 5–6).

2 Preliminaries

2.1 ML-DSA

ML-DSA is a lattice-based signature scheme whose security rests on the hardness of the Module-LWE [13] and Module-SIS problems. All operations are defined over the polynomial ring $\mathcal{R}_{q_{\mathrm{std}}} = \mathbb{Z}_{q_{\mathrm{std}}}[X]/(X^n + 1)$ with $n = 256$ and the standard modulus $q_{\mathrm{std}} = 8\,380\,417 = 2^{23} - 2^{13} + 1$. Throughout we write q_{std} for this standard prime and use q for the small *carrier prime* used in the small-NTT

Table 1: ML-DSA parameter sets.

NIST security level	2	3	5
τ (number of ± 1 in c)	39	49	60
(k, ℓ) (dimensions of $\mathbf{s}_1, \mathbf{s}_2$)	$(4, 4)$	$(6, 5)$	$(8, 7)$
η (coefficient bound of $\mathbf{s}_1, \mathbf{s}_2$)	2	4	2

computation; at NIST security level 3, $q = 769$. The targeted intermediate values are represented modulo this carrier prime. The secret vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ consist of short polynomials: $\mathbf{s}_1 \in S_\eta^\ell$ and $\mathbf{s}_2 \in S_\eta^k$, where $S_\eta = \{w \in \mathcal{R}_{q_{\text{std}}} : \|w\|_\infty \leq \eta\}$. Thus, every coefficient of each secret polynomial lies in $[-\eta, \eta]$, a property used later in the inverse-NTT verification step. The challenge c is a sparse ± 1 polynomial with τ non-zero coefficients. The parameters η, τ, k, ℓ depend on the security level (Table 1).

2.2 The Number-Theoretic Transform

Polynomial multiplication in $\mathcal{R}_{q_{\text{std}}}$ dominates the runtime of lattice-based schemes. The number-theoretic transform (NTT) is a finite-field analogue of the FFT that lowers the cost from $O(n^2)$ to $O(n \log n)$ and makes multiplication point-wise in the NTT domain, with an inverse transform at the end. For Cortex-M4-class implementations such as those in pqm4 [5], the NTT is *truncated* after $m < \log_2 n$ layers of Cooley-Tukey [14] butterflies to save instructions and memory accesses [15]. Instead of decomposing into n scalar coefficients, the polynomial is mapped to a collection of polynomials of degree $(n/2^m) - 1$, and multiplication in the transformed domain is performed pair-wise using the schoolbook algorithm. Throughout this paper we set $m = \log_2(n/2)$, so the polynomial $s = (s_0, \ldots, s_{n-1})$ is mapped to

$$\mathbf{s}' = \text{IncomNTT}(s) = \{(s'_{[0,0]}, s'_{[0,1]}), \ldots, (s'_{[n/2-1,0]}, s'_{[n/2-1,1]})\},$$

a sequence of $n/2$ coefficient pairs, each of degree 1. Multiplication of $\mathbf{s}'$ by $\mathbf{c}'$ in this domain computes, for each $i = 0, \ldots, n/2 - 1$,

$$\begin{aligned} r_{[i,0]} &= s'_{[i,0]} \cdot c'_{[i,0]} + s'_{[i,1]} \cdot c'_{[i,1]} \cdot \delta_i \pmod q, \\ r_{[i,1]} &= s'_{[i,0]} \cdot c'_{[i,1]} + s'_{[i,1]} \cdot c'_{[i,0]} \pmod q, \end{aligned} \tag{1}$$

where δ_i is a fixed twiddle constant.

Small NTT. Because each component polynomial of the secret vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ is short and c is sparse with ± 1 coefficients, the magnitude of the products targeted in the implementation is bounded by a quantity much smaller than the standard prime q_{std}. A so-called *carrier prime* [7] q, much smaller than q_{std}, can therefore be used to carry the intermediate values. For NIST security level 3, the typical carrier prime used in optimized Cortex-M4 implementations is $q = 769$, which we use throughout this paper.

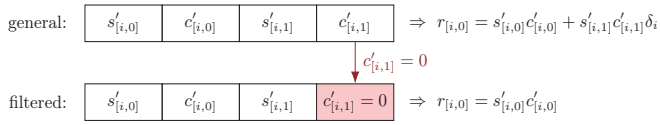


Fig. 1: Zero-value filtering of ZV-FA. Selecting traces with $c'_{[i,1]} = 0$ makes $r_{[i,0]}$ depend only on $s'_{[i,0]}$.

2.3 The Zero-Value Filtering Attack (ZV-FA)

The Zero-Value Filtering Attack [12] consists of three stages: filtering, scoring, and inverse-NTT verification. In what follows we describe these stages as they appear in our threat model.

Filtering. The attacker targets the intermediate value $r_{[i,0]} = s'_{[i,0]} \cdot c'_{[i,0]} + s'_{[i,1]} \cdot c'_{[i,1]} \cdot \delta_i \pmod q$. A direct CPA against $r_{[i,0]}$ would require enumerating q^2 hypotheses per pair. ZV-FA reduces this by selecting only traces that satisfy the zero-value condition $c'_{[i,1]} = 0 \pmod q$.Under this condition, the second summand vanishes and $r_{[i,0]} = s'_{[i,0]} \cdot c'_{[i,0]} \pmod q$, which depends on $s'_{[i,0]}$ alone. Figure 1 illustrates this zero-value filtering step. The attacker then mounts a CPA over the filtered traces against each candidate of $s'_{[i,0]}$. An analogous attack on the symmetric condition $c'_{[i,0]} = 0 \pmod q$ recovers $s'_{[i,1]}$ independently. The resulting correlation values from the two attacks are combined into a per-candidate score, and the candidates of each coefficient are sorted into rank lists $\mathcal{K}^0, \mathcal{K}^1$.

Scoring. For each coefficient pair, a search depth d is initialized to 1. From the top of $\mathcal{K}^0$ and $\mathcal{K}^1$, the attacker assembles $\mathcal{K} = \mathcal{K}^0_{[:d]} \times \mathcal{K}^1_{[:d]}$ and performs a second CPA over *all* traces, assuming the pair $(s'_{[i,0]}, s'_{[i,1]})$ jointly and evaluating $r_{[i,0]}, r_{[i,1]}$; the highest-scoring pair is the provisional answer. After collecting the answers for all $n/2 = 128$ pairs, the assembled key is verified; whenever verification fails, the pair with the lowest score is deemed responsible, d is doubled for that pair, and the procedure continues.

Inverse-NTT verification. The assembled $\mathbf{s}'$ is mapped back to the polynomial domain via the inverse incomplete NTT, $s = \mathrm{IncomNTT}^{-1}(\mathbf{s}')$, and the attacker checks $\|s\|_\infty \leq \eta$; because the genuine secret is short, any coefficient outside $[-\eta, \eta]$ signals an incorrect guess.

3 Zero-Correlation Issue and Zero-Value Prioritization

We first describe the zero-correlation issue and then introduce the resulting zero-value prioritization before scoring.

3.1 CPA Hypothesis and Leakage Vectors

Fix a coefficient $s'_{[i,0]}$ under attack. For each of the N retained filtering traces, the attacker computes the predicted intermediate value under a value hypothesis for $s'_{[i,0]}$, takes its Hamming weight, and forms the *hypothetical leakage vector* $H = (\mathrm{HW}(r_1), \ldots, \mathrm{HW}(r_N))$. Let $T = (T_1, \ldots, T_N)$ be the corresponding vector of *measured* leakage samples. A CPA scores the hypothesis by the Pearson correlation

$$\rho = \frac{\mathrm{Cov}(H, T)}{\sigma_H \sigma_T},$$

and ranks the candidates of $s'_{[i,0]}$ by this score.

3.2 Vanishing Variance and Uncomputable Correlation

Proposition 1 (Zero-correlation issue at filtering). *Consider the filtering CPA on $r_{[i,0]} = s'_{[i,0]} \cdot c'_{[i,0]}$ performed under the zero-value condition $c'_{[i,1]} = 0$. If the true value of $s'_{[i,0]}$ is $0 \bmod q$, then $r_{[i,0]} = 0$ identically across all retained traces, regardless of $c'_{[i,0]}$. Consequently the hypothetical leakage vector H is the constant zero vector, so $\sigma_H = 0$ and the denominator of the Pearson correlation $\rho = \mathrm{Cov}(H, T)/(\sigma_H \sigma_T)$ is zero, so ρ cannot be computed.*

The original ZV-FA does not specify how to handle this undefined correlation. In our reimplementation, we assign it a score of 0 to avoid unsupported overestimation. Consequently, the candidate value $s'_{[i,0]} = 0$ tends to fall toward the lower part of the rank list when it is the true value. The same undefined-correlation case arises in the scoring stage for the candidate pair $(0, 0)$. Under our score-zero handling, this can increase the search depth and, in the worst case, prevent key recovery.

Remark 1. The relevant event is not that a secret coefficient equals the integer 0, but that an incomplete-NTT coefficient satisfies $s'_{[i,j]} \equiv 0 \pmod q$ for $q = 769$. Treating residues modulo q as roughly uniform, the probability that at least one of the $n = 256$ coefficients is $0 \bmod q$ is $1 - (1 - 1/769)^{256} \approx 0.28$. Thus, the issue affects a non-negligible fraction of keys.

3.3 Zero-Value Prioritization via $\tau_{\mathbf{zero}}$

The zero-correlation issue also provides a useful signal before the original ZV-FA scoring step. Let λ_{rank1} denote the largest filtering score for a coefficient $s'_{[i,j]}$. When the true value is non-zero and the filtering quality is sufficient, λ_{rank1} tends to be large. In contrast, when the true value is $0 \bmod q$, its correlation is undefined and assigned a score of 0 in our reimplementation, while the rank-1 score tends to remain low. We use this behavior to prioritize the candidate value 0 before scoring. Specifically, when

$$\lambda_{\mathrm{rank1}} < \tau_{\mathrm{zero}},$$

the value 0 is treated as a high-priority candidate for $s'_{[i,j]}$. The threshold τ_{zero} s not derived from the true secret values; it must be chosen from observable filtering scores or prior experiments. Its selection involves a trade-off between missed zero-valued coefficients and falsely flagged non-zero coefficients, as discussed in Section 6.

4 Cost-Based Key Enumeration

Key enumeration from ranked candidate lists has been studied using probability-based optimal methods and practical score-based methods [16, 17]. We apply score-based key enumeration to the rank lists produced by ZV-FA filtering and attempt key recovery without invoking the original scoring step. Each candidate is assigned a relative score-drop cost, and full-key candidates are explored from lower to higher total cost.

4.1 Cost as Relative Score Decrease

For every candidate of every coefficient, we define a heuristic search cost as the fractional decrease of its filtering score relative to the rank-1 score:

$$\text{cost}(\text{cand}) = \frac{\lambda_{\text{rank1}} - \lambda_{\text{cand}}}{\lambda_{\text{rank1}}}.$$

The rank-1 candidate has cost 0, and candidates at worse ranks generally have larger costs. This relative normalization reduces the influence of coefficient-dependent score magnitudes. The cost is not a calibrated probability or likelihood, and therefore does not guarantee posterior-optimal key ordering. Since the cost is assigned to every candidate, the search can naturally consider lower-ranked candidates when needed. The candidate value 0 prioritized by τ_{zero} is also assigned cost 0.

4.2 Enumeration Algorithm

The search begins from the *baseline key* $\mathbf{s}'_{\text{base}}$, which selects the rank-1 candidate of every coefficient. Using a priority queue, the procedure verifies the available candidate with the lowest total cost, generates one-coefficient advances when verification fails, and stops when a candidate passes the inverse incomplete-NTT short-coefficient check or when the search cap is reached. Algorithm 1 states this formally, and Fig. 2 summarizes the flow. The queue is implemented as a binary heap. In our experiments, we use no queue-size limit and declare failure once 100,000 verifications have been performed, as discussed in Section 6.

4.3 One-Directional Expansion for Duplicate Avoidance

Naively generating successors from failed combinations can create duplicates. For example, from the baseline pair $(0, 0)$, both paths $(0, 0) \to (1, 0) \to (1, 1)$

Algorithm 1 Cost-based key enumeration used in this work.

Input: Per-coefficient candidate lists sorted by ascending cost; cap L
Output: Recovered secret key $\mathbf{s}'$, or FAILURE

```
s'_base ← rank-1 (cost-0) candidate of every coefficient
cost_base ← 0;    heap ← {(cost_base, s'_base)}
count ← 0
while heap not empty do
    (cost, s'_cur) ← pop-min(heap)
    count ← count + 1
    if InvNTT_Verify(s'_cur) then
        return s'_cur
    end if
    if count ≥ L then
        return FAILURE
    end if
    for each one-coefficient advance of s'_cur allowed by the one-directional rule do
        form s'_next and its total cost cost_next
        push (cost_next, s'_next) onto heap
    end for
end while
return FAILURE
```

and $(0,0) \to (0,1) \to (1,1)$ reach the same combination. We avoid this by allowing a new advance only on a coefficient whose index is at least that of the most-recently advanced coefficient. Thus $(1,1)$ is generated only from $(1,0)$, not from $(0,1)$. This rule removes duplicates while preserving the reachable set, as illustrated in Figure 3.

5 Simulation Evaluation

5.1 Setup

Our experiments are simulation-based: physical measurements on a Cortex-M4 board are deferred to future work. We implemented the ZV-FA baseline and the proposed method based on the attack description in [12]. We used an incomplete NTT truncated at the $\log_2(n/2)$-th layer and the carrier prime $q = 769$ to reproduce small-NTT Cortex-M4 implementations at NIST security level 3. We modeled the leakage of the intermediate value $r_{[i,j]}$ in Eq. (1) as

$$T = \mathrm{HW}(r_{[i,j]}) + \mathcal{N}(0, \sigma^2),$$

with $\sigma = 10$. Since the original ZV-FA does not specify the word size, we compute the Hamming weight using the 32-bit two's-complement representation of the centered intermediate value. When the hypothetical leakage vector has zero variance, the Pearson correlation is undefined; in our reimplementation, we assign the corresponding hypothesis a score of 0. The experiments were implemented in Python 3.13.5 and executed on macOS 15.7.3 using a 10-core Apple M2 Pro processor with 16 GB of RAM.

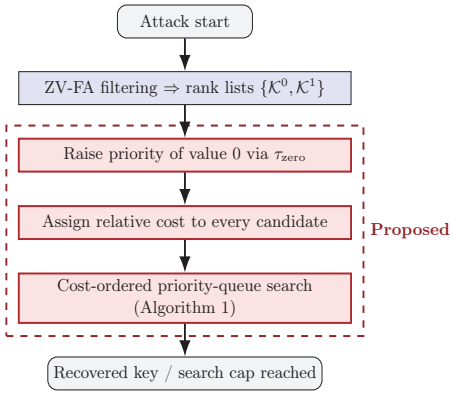


Fig. 2: Overall flow from the ZV-FA filtering output to cost-based enumeration. After zero-value prioritization, relative score-drop costs are assigned to the candidates, and lower-cost combinations are tested first using Algorithm 1.

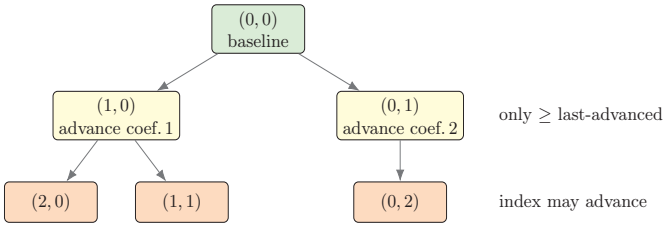


Fig. 3: Duplicate avoidance by one-directional expansion. The combination $(1,1)$ is generated only from $(1,0)$.

5.2 Choosing the Filtering-Trace Range

To choose the trace-count range, we first studied the rank-of-truth distribution on a trace dataset in which one coefficient's true value is zero (Figure 4). As N_f grows, true values concentrate at the top ranks; for $N_f \gtrsim 15{,}000$, nearly every coefficient is ranked near the top, except for a few low-ranked cases including the zero-valued coefficient. Based on this observation, we evaluate $N_f \in \{15{,}000, \ldots, 19{,}000\}$.

In the experiments below, we use $\tau_{\text{zero}} = 0.4$, chosen empirically from the observed rank-1 score distribution. The risks of missed zero-valued coefficients and falsely flagged non-zero coefficients are discussed in Section 6.

5.3 Multi-Key Evaluation

We evaluated 50 randomly generated secret keys, which were classified into 12 zero-present keys whose $\mathbf{s}'$ contains at least one incomplete-NTT coefficient equal to $0 \bmod q$, and 38 zero-absent keys with no such coefficient. For each $N_f \in \{15{,}000, \ldots, 19{,}000\}$, we recorded the success rate and execution time, treating

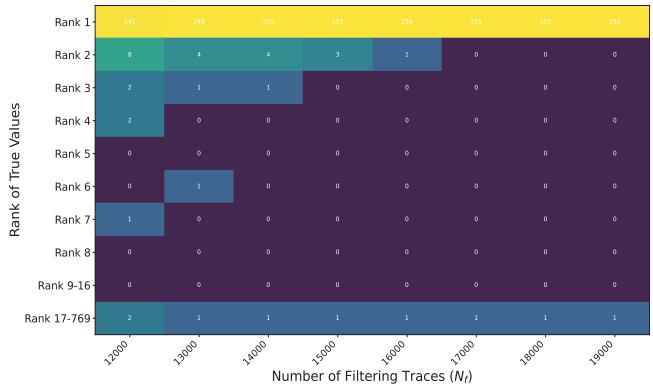


Fig. 4: Distribution of the rank of the true value across the 256 coefficients, as a function of N_f. For $N_f \gtrsim 15{,}000$ the truth is at the top ranks for essentially every coefficient; the bottom row collects the few coefficients (including the zero-valued one) whose true values are ranked low.

100,000 inverse incomplete-NTT verifications as failure; reaching this cap takes about 60 s.

Success rate. Figure 5 shows that the success rate of the proposed enumeration increases with N_f, from about 0.42 at $N_f = 15{,}000$ to 1.0 at $N_f = 19{,}000$. This indicates that, as the filtering output becomes more accurate, more true values appear among low-rank or low-cost candidates and fewer runs hit the verification cap.

Mean time with failed runs. Figure 6 plots the mean execution time with failed runs. A failed run contributes only the time until the search is aborted at the verification cap, so these averages are lower bounds on the true mean recovery time. With this caveat, for zero-present keys the proposed enumeration shows shorter mean execution time than ZV-FA once N_f is large, while for zero-absent keys it starts slower at small N_f and approaches ZV-FA as N_f grows.

Mean time for successful runs. Table 2 reports the success-only comparison, timing both schemes on the datasets where the proposed enumeration succeeded. For zero-present keys, the proposed method completes in about 9–14 s, compared with 48–51 s for ZV-FA. In contrast, ZV-FA completes in about 9 s for zero-absent keys, showing a clear association between zero-valued coefficients and increased ZV-FA recovery time. For zero-absent keys, the proposed method is slower at small N_f and matches ZV-FA once N_f is large.

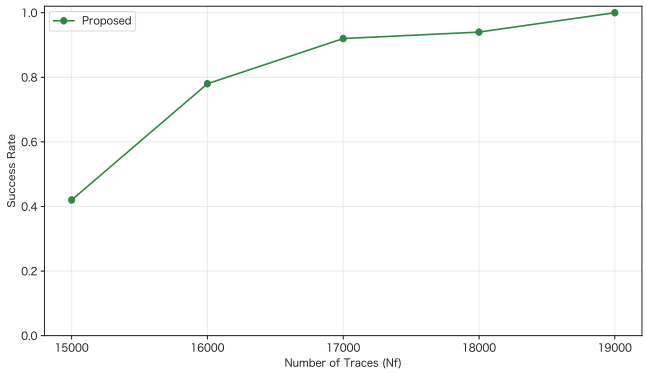


Fig. 5: Success rate of the proposed enumeration as a function of N_f, computed over 50 randomly drawn keys. Runs reaching the 100,000-verification cap are counted as failures.

Table 2: Success rate and mean time for successful runs as a function of N_f. Means are computed over datasets where the proposed enumeration succeeded. The zero-present and zero-absent sets contain 12 and 38 keys.

Population	N_f	Success rate	ZV-FA mean (s)	Proposed mean (s)
Zero present	15,000	0.750	51.35	14.13
	16,000	1.000	48.41	12.87
	17,000	1.000	49.42	12.02
	18,000	1.000	48.46	11.74
	19,000	1.000	49.22	9.28
Zero absent	15,000	0.316	8.89	17.12
	16,000	0.711	8.98	10.72
	17,000	0.895	9.13	10.76
	18,000	0.921	9.13	9.22
	19,000	1.000	9.22	9.24

6 Discussion

6.1 Behavior of the Proposed Enumeration

The enumeration procedure applies score-based key enumeration and inverse incomplete-NTT verification to the ZV-FA filtering output to attempt key recovery without running the original scoring CPA. On successful runs, it is faster than ZV-FA for zero-present keys at sufficiently large N_f, but shows no clear runtime advantage for zero-absent keys. Its success depends on filtering quality: low-rank or low-cost true values facilitate recovery within the verification budget, whereas inaccurate filtering increases the risk of reaching the cap. The fixed choices of $\tau_{\text{zero}} = 0.4$ and 100,000 verifications require further study, particularly regarding the trade-off between missed and falsely prioritized coefficients.

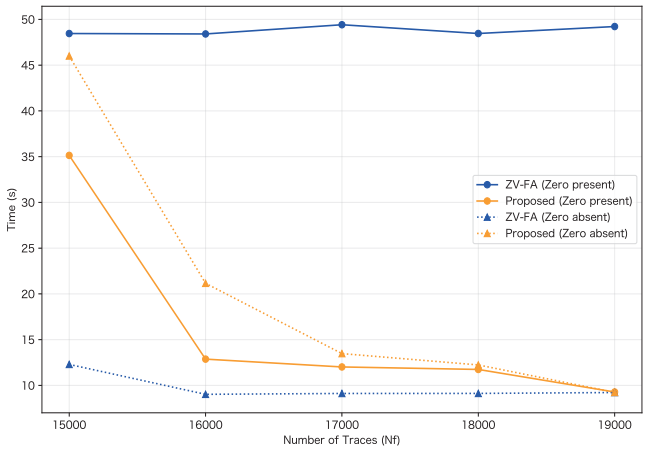


Fig. 6: Mean execution time with failed runs. The zero-present and zero-absent sets contain 12 and 38 keys. Failed runs are timed only until the verification cap, so the means are lower bounds.

6.2 Future Work

Our evaluation is simulation-based with an additive Gaussian noise model. Future work will evaluate the attack on physical traces from a Cortex-M4 implementation to validate the leakage model under realistic conditions. Another direction is to combine the enumeration procedure with the original ZV-FA scoring step: enumeration is attempted first, followed by ZV-FA scoring when necessary. Making this hybrid practical requires rules for deciding, from observable filtering information, when enumeration should be attempted and when it should be handed over to scoring.

7 Conclusion

We identified a zero-correlation issue in ZV-FA against incomplete-NTT ML-DSA and prioritized zero-valued candidates using the rank-1 filtering score. We also applied score-based key enumeration to recover keys without the original scoring CPA. Simulations showed that the method succeeds when the filtering output is sufficiently accurate, with greater benefits for zero-present keys. Combining enumeration with ZV-FA scoring remains future work.

Acknowledgements

This work was supported by JST K Program, Grant Number JPMJKP24U2, Japan.

References

1. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proc. 35th Annual Symposium on Foundations of Computer Science (FOCS), pp. 124–134. IEEE (1994)
2. National Institute of Standards and Technology: Module-Lattice-Based Digital Signature Standard. FIPS 204 (2024)
3. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018(1), 238–268 (2018)
4. 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 (2021)
5. Kannwischer, M.J., Rijneveld, J., Schwabe, P., Stoffelen, K.: pqm4: Testing and benchmarking NIST PQC on ARM Cortex-M4. In: Second PQC Standardization Conference (2019)
6. Alkim, E., Bilgin, Y.A., Cenk, M., Gérard, F.: Cortex-M4 optimizations for {R,M}LWE schemes. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020(3), 336–357 (2020)
7. Abdulrahman, A., Hwang, V., Kannwischer, M.J., Sprenkels, D.: Faster Kyber and Dilithium on the Cortex-M4. In: Ateniese, G., Venturi, D. (eds.) ACNS 2022. LNCS, vol. 13269, pp. 853–871. Springer (2022)
8. Chen, Z., Karabulut, E., Aysu, A., Ma, Y., Jing, J.: An efficient non-profiled side-channel attack on the CRYSTALS-Dilithium post-quantum signature. In: Proc. IEEE 39th International Conference on Computer Design (ICCD), pp. 583–590. IEEE (2021)
9. Steffen, H., Land, G., Kogelheide, L., Güneysu, T.: Breaking and protecting the crystal: Side-channel analysis of Dilithium in hardware. In: Post-Quantum Cryptography – PQCrypto 2023. LNCS, vol. 14154, pp. 688–711. Springer (2023)
10. Qiao, Z., Liu, Y., Zhou, Y., Shao, M., Sun, S.: When NTT meets SIS: Efficient side-channel attacks on Dilithium and Kyber. IACR Cryptol. ePrint Arch., Report 2023/1866 (2023)
11. Tosun, T., Moradi, A., Savas, E.: Exploiting the central reduction in lattice-based cryptography. IEEE Access 12, 166814–166833 (2024)
12. Tosun, T., Savas, E.: Zero-value filtering for accelerating non-profiled side-channel attack on incomplete NTT-based implementations of lattice-based cryptography. IEEE Trans. Inf. Forensics Secur. 19, 3353–3365 (2024)
13. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6), 34:1–34:40 (2009)
14. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Math. Comput. 19(90), 297–301 (1965)
15. Lyubashevsky, V., Seiler, G.: NTTRU: Truly fast NTRU using NTT. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019(3), 180–201 (2019)
16. Veyrat-Charvillon, N., Gérard, B., Renauld, M., Standaert, F.-X.: An optimal key enumeration algorithm and its application to side-channel attacks. In: Selected Areas in Cryptography – SAC 2012. LNCS, vol. 7707, pp. 390–406. Springer (2013)
17. Bogdanov, A., Kizhvatov, I., Manzoor, K., Tischhauser, E., Witteman, M.: Fast and memory-efficient key recovery in side-channel attacks. In: Selected Areas in Cryptography – SAC 2015. LNCS, vol. 9566, pp. 310–327. Springer (2016)

Differential Cryptanalysis of the Block Cipher LTLBC

Ryotaro Kai [1] and Yasutaka Igarashi [1] and Naoki Shibayama [1]

[1] Tokyo University of Science 2641 Yamazaki, Noda, Chiba, 278-8510, Japan
{7325539@ed, yasutaka@rs.noda,7323703@ed,}.tus.ac.jp

Abstract. This paper evaluates the security of LTLBC, a 64-bit lightweight block cipher proposed by Sun et al. in 2024 with a 128-bit key and 14-round, against differential cryptanalysis. Building on our previous STP-based differential analysis, which identified two 6-round differential characteristics with probability $2^{-49}$and established that the maximum 7-round single-characteristic probability is 2^{-64}, we construct a key-recovery attack on 8-round reduced LTLBC with a success probability of about 96.5%. The attack requires 2^{52}chosen plaintexts and approximately $2^{52.1}$equivalent 8-round encryptions. This result does not threaten the full 14-round LTLBC, but it provides a concrete third-party evaluation of its reduced-round security.

Keywords: Block Cipher, LTLBC, Differential Cryptanalysis

1 Introduction

LTLBC [1] is a low-latency lightweight block cipher proposed in 2024 for resource-constrained environments. It has a 64-bit block size, a 128-bit master key, and 14-round. The original proposal mainly evaluated resistance to differential cryptanalysis using lower bounds on the number of active S-boxes for up to 6-round. However, active S-box counts alone do not determine the maximum single differential characteristic probability or the feasibility of a reduced-round key-recovery attack. In particular, the positions of active S-boxes near the boundary rounds determine which round-key parts can be estimated from the ciphertext side. Das [2] independently reassessed LTLBC using MILP-based integral, linear, and differential cryptanalys. That work revised the active S-box counts through 9-round and presented a concrete 6-round differential trail with 19 active S-boxes and probability 2^{-55}. However, it did not establish the maximum single differential characteristic probability or construct a key-recovery attack. In our previous work [3], we modeled the exact differential weights of LTLBC as an SMT problem and solved it using STP [4]. We determined the maximum single differential characteristic probabilities for 1 to 7-round, reported two distinct 6-round characteristics with probability 2^{-49}, and established that the maximum 7-round single-characteristic probability is 2^{-64}. The present paper substantially extends [3] by using these two characteristics to construct a key-recovery attack on 8-round reduced LTLBC. The main contributions of this paper are as follows.

1. Using the two 6-round differential characteristics reported in [3], we exploit their active S-box positions in 7-round and 8-round to construct a key-recovery attack on 8-round reduced LTLBC.
2. By combining the two characteristics, differential filtering determines a total of 108 bits of $RK_7$and RK_8. The remaining 20 bits are exhaustively searched under the key-schedule constraints.
3. A four-plaintext structure shares the data required by the two input differences. The attack requires 2^{52}chosen plaintexts and approximately $2^{52.1}$equivalent 8-round encryptions, with a success probability of approximately 96.5%.

2 LTLBC

LTLBC is a lightweight block cipher proposed for low-latency IoT applications. It has a 64-bit block size, a 128-bit master key, and 14-round. Fig. 1 shows the overall encryption structure of LTLBC, including the key schedule. The master key $\boldsymbol{K}$ is divided into two 64-bit halves. In this paper, K_0denotes the upper 64 bits of $\boldsymbol{K}$, which is used as the initial whitening key before the first round. The round keys used inside the round function are denoted by RK_i for $i = 1, \ldots, 14$. Each round consists of PermuteBits, MixWord, SubCell, AddRoundKey, and AddRoundConstant, where MixWord is omitted in the final round. Let RK_i and RC_i denote the 64-bit round key and the round constant in i-round, respectively. The 64-bit plaintext is processed by PermuteBits, MixWord, and SubCell, where PermuteBits is a bit-wise permutation layer, MixWord is a word-wise linear mixing layer on four 16-bit branches, and SubCell applies the 4-bit S-box to sixteen 4-bit blocks in parallel. AddRoundKey XORs $RK_i$to the state, and AddRoundConstant XORs $RC_i$to the state. According to the key schedule in [1], the key update is invertible and preserves the two 64-bit branches of the key state. Therefore, strong dependencies exist among round keys of the same parity, and two consecutive 64-bit round keys determine the corresponding 128-bit key state. The full specification of LTLBC is given in [1].

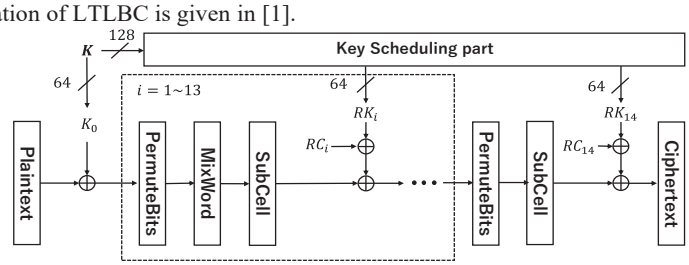


Fig. 1. Overall encryption structure of LTLBC

3 Differential Characteristics of LTLBC

This section summarizes the differential characteristic search results reported in our previous work [3]. Table 1 is reproduced from [3], and Table 2 adapts the values required for the key-recovery attack.

3.1 Differential Modeling

The differential characteristic search reported in [3] was formulated as a Satisfiability Modulo Theories (SMT)-based model. SMT is a framework for deciding the satisfiability of logical formulas with respect to background theories, including fixed-size bit-vectors. STP is an SMT solver specialized in fixed-size bit-vectors and arrays, enabling the bit-level operations of a block cipher to be represented exactly. In our previous work [3], we used the STP implementation distributed through its GitHub repository [4]. Since AddRoundKey and AddRoundConstant are XOR operations, they do not affect differential characteristics. Thus, the differential propagation is determined by SubCell, PermuteBits, and MixWord. The S-box constraints are derived from the differential distribution table (DDT), which was generated using SageMath [5]. PermuteBits and MixWord are modeled as deterministic bit-vector constraints.

For each S-box transition with probability p, its differential weight is defined as

$$w = -\log_2 p \tag{1}$$

For an i-round characteristic, the total differential weight was defined as

$$W_i = \sum_{1}^{i} \sum_{r=0}^{15} w_{i,r} \tag{2}$$

where the sum is taken over all S-box transitions in the characteristic. The objective of the search was to minimize W_i. The complete implementation details of the model are given in [3]. To determine the optimum for each round number, we imposed a nonzero input difference and queried STP under the constraint

$$W_i \leq t \tag{3}$$

The threshold t was located by binary search, using the optimum for the preceding round as the lower bound and three times that value as the initial upper bound. For 7-round, STP returned

$$W_7 \leq 63: UNSAT, W_7 \leq 64: SAT \tag{4}$$

The satisfiable instance returned a concrete characteristic of weight 64. Consequently, the minimum 7-round differential weight is 64, and the maximum 7-round single differential characteristic probability is 2^{-64}. We evaluate single differential characteristics, not full differentials. Therefore, the reported characteristic probability is a lower bound on the actual differential probability, and trail-clustering effects are not considered.

3.2 Search Results

Table 1 shows the maximum single differential characteristic probability for each round number. In this paper, DCP_i denotes the maximum single differential characteristic probability for i-round LTLBC. These values were reported in [3] and are reproduced here for use in the subsequent attack.

Table 1. Maximum single differential characteristic probabilities of LTLBC from 1 to 7-round (reproduced from [3])

Round	1	2	3	4	5	6	7
DCP_i	2^{-2}	2^{-8}	2^{-14}	2^{-28}	2^{-39}	2^{-49}	2^{-64}

The 7-round probability 2^{-64}matches the random-permutation baseline for a fixed difference transition and thus provides no single-characteristic distinguishing advantage, although characteristic clustering may yield a higher-probability differential. Our previous work identified two distinct 6-round differential characteristics, each with probability 2^{-49}, hereafter referred to as Differential Characteristic 1 (DC1) and Differential Characteristic 2 (DC2) and used in the key-recovery attack. Das [2] reported a 6-round trail with 19 active S-boxes and probability 2^{-55}, which is consistent with Table 1 because minimizing the number of active S-boxes does not necessarily minimize the exact DDT weight. In each nonfinal round, MixWord is immediately followed by SubCell; hence, each nonzero 4-bit cell in the post-MixWord difference directly identifies an active S-box. The 7-round post-MixWord differences from [3] determine the 7-round active positions, while the 8-round positions are obtained by propagating each characteristic through the remainder of 7-round and the 8-round PermuteBits layer; MixWord is omitted in the final round. Table 2 summarizes the differences and active S-box positions required for the attack. The S-boxes are indexed from 0 to 15, from the leftmost to the rightmost 4-bit cell.

Table 2. Input difference, 6-round Output Differences, 7-round post-MixWord Differences, and Active S-box positions in 7-round for DC1 and DC2 (all values reproduced from [3])

Differential Characteristic	Input difference	6-round Output Differences	7-round post-MixWord Difference	Active S-box positions in 7-round
DC1	0x1000400018004200	0x0000060000000000	0x0000000001104026	9,10,12,14,15
DC2	0xA9108000AC008400	0x5600060000000000	0x0909080840D93012	1,3,5,7,8,10,11,12,14,15

4 Key Recovery Attack on 8-Round LTLBC

This section constructs a key-recovery attack on 8-round reduced LTLBC using DC1 and DC2, which were reported in [3] and summarized in Table 2. The key-recovery construction itself is new to the present paper. In the LTLBC specification, MixWord is omitted in the final round. Therefore, when the 8th round is regarded as the final round in the 8-round reduced variant, MixWord is not applied in 8-round. The attack first estimates the 7 and 8-round key pair (RK_7, RK_8), and then reconstructs the 128-bit master key through the inverse key schedule. This reconstruction is possible because the LTLBC key update is invertible, and two consecutive 64-bit round keys determine the corresponding 128-bit key state. We first describe the attack using DC1, then use DC2 to estimate additional round-key bits and finally evaluate the data and time complexities.

4.1 Key Recovery using DC1

Let $P$and P'be a plaintext pair satisfying the input difference of the corresponding differential characteristic and let C and C'be the corresponding ciphertexts. From Table2, DC1 uses plaintext pairs with the following input difference:

$$P \oplus P' = 0\text{x}1000400018004200 \tag{5}$$

In this attack, the 64-bit round key $RK_i$is divided into sixteen 4-bit subkey parts as

$$RK_i = (RK_{i,0}, RK_{i,1}, \dots, RK_{i,15}).$$

Here, $RK_{i,r}$denotes the r-th 4-bit subkey part of the i-th round key, corresponding to the r-th S-box position, where $0 \le r \le 15$. Thus, "round key" refers to the 64-bit key RK_i, whereas "subkey part" refers to one of its 4-bit components $RK_{i,r}$. For $n \in \{1,2\}$, $\Delta a_{n,7}$denotes the post-MixWord difference in 7-round for DC n. If 2^{49} plaintext pairs with this input difference are prepared, then one right pair is expected because the 6-round differential characteristic probability is 2^{-49}. For DC1, the post-MixWord difference in 7-round is given from Table 2 as

$$\Delta a_{1,7} = 0x0000000001104026 \tag{6}$$

In Figs. 2, only the 4-bit round-key parts related to nonzero differences are shown. Although the round key is applied to all state bits, zero-difference positions do not provide key-dependent filtering conditions. Therefore, 4-bit round-key parts associated only with zero-difference positions are omitted for clarity. From $\Delta a_{1,7}$, the active S-box positions in 8-round are obtained by applying the PermuteBits layer to the nonzero 4-bit blocks after the 7-round MixWord. Thus, the active S-box positions in 8-round are determined only by PermuteBits, as shown in Fig. 2. In Fig. 2, the leftmost bit is indexed as 0 and the rightmost bit as 63. Blue positions indicate zero difference, while red positions indicate nonzero difference. From Fig. 2, five S-boxes are active in 7-round, and eight S-boxes are active in 8-round. Therefore, DC1 determines $(5+8) \times 4 = 52$ round-key bits of RK_7 and RK_8.

Next, we describe the ciphertext-pair filtering. The ciphertext difference is defined as

$$\Delta C = C \oplus C' \tag{7}$$

According to Fig. 2, the ciphertext pairs used in the attack must satisfy the following 4-bit-wise difference pattern:

$$\Delta C = (\alpha, \alpha, \alpha, \alpha, 0_1, 0_2, 0_3, 0_4, \alpha, \alpha, \alpha, \alpha, 0_5, 0_6, 0_7, 0_8), 0_i = 0 \ (1 \le i \le 8) \tag{8}$$

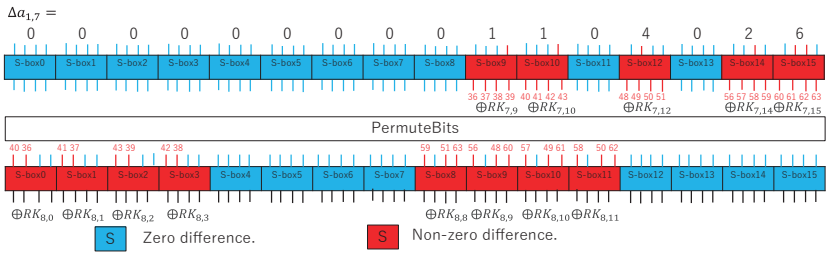


Fig. 2. Boundary-round active positions and estimated 4-bit subkey parts for DC1

Here, α denotes an arbitrary 4-bit difference value and 0_i denotes the i-th 4-bit zero-difference condition. Since eight 4-bit positions must be zero, each zero condition holds with probability 2^{-4} under the uniformity assumption.
Therefore, the probability that all eight zero-difference conditions are satisfied is

$$(2^{-4})^8 = 2^{-32} \tag{9}$$

Since 2^{49}plaintext pairs are prepared, the expected number of ciphertext pairs satisfying the above condition is

$$2^{49} \times 2^{-32} = 2^{17} \tag{10}$$

In this attack, the eight zero-difference conditions are not checked simultaneously. Instead, they are checked sequentially. Specifically, we first test whether the position labeled $0_1$is zero and discard the ciphertext pairs that do not satisfy this condition. Then, for the remaining pairs, we test 0_2, and this procedure continues up to 0_8. At the k-th zero-difference check, the expected number of candidate ciphertext pairs is

$$2^{49} \times 2^{-4(k-1)} \tag{11}$$

Hence, the total number of zero-difference checks is

$$\sum_{k=1}^{8} 2^{49-4(k-1)} = 2^{49}(1 + 2^{-4} + 2^{-8} + 2^{-12} + 2^{-16} + 2^{-20} + 2^{-24} + 2^{-28}) \cong 2^{49.1} \tag{12}$$

In this paper, we assume that one zero-difference check has the same cost as one S-box operation. In LTLBC, one round uses 16 S-boxes in the data path and one additional S-box in the key schedule. Therefore, one 8-round encryption requires $(16 + 1) \times 8 = 136$ S-box operations. Thus, the computational complexity of the sequential zero-difference filtering, expressed in equivalent 8-round encryptions, is

$$\frac{2^{49.1}}{136} \cong 2^{42} \tag{13}$$

After this filtering, 2^{17}ciphertext pairs remain on average. For the remaining ciphertext pairs, the active 8-round subkey parts are guessed at the 4-bit level, and the corresponding S-boxes are partially inverted from the ciphertext side. This partial decryption recovers the necessary bits of the 7-round output difference. The recovered difference is then used to guess the active 7-round keys and to check whether the required S-box input-difference conditions of DC1 are satisfied. For compactness, Table 3 uses the following abbreviations. “Zero-bit check” means that the unrecovered bit positions must be zero according to the expected 7-round output difference of DC1. “S-box input check” means that the computed input differences of the corresponding 7th-round S-boxes must match the expected differences of DC1. The column “Computed bits” lists the bit positions of the 7-round output difference obtained by partial decryption. In Tables 3 and 4, $S_r$denotes the input difference of the r-th S-box in the 7th round after partial decryption from the ciphertext side. As shown in Table 3, DC1 estimates five 7th-round 4-bit subkey parts and eight 8-round 4-bit subkey parts. Therefore, DC1 determines 52 bits round-key in total. For a wrong round-key candidate, the expected number of remaining pairs after one filtering trial is 2^{-15}, as shown in the final

"Remaining pairs" column of Table 3. We use this expected value as the passing probability of a wrong round-key candidate. Therefore, the probability that a wrong round-key candidate passes one filtering trial is 2^{-15}.

Table 3. Filtering process for ciphertext pairs in DC1

step	Operation	Guess RK	Condition/Computed bits	Check	Remaining pairs
1	Zero-difference filtering	—	—	Eight 4-bit zero-difference conditions	2^{17}
2	Partial decryption of $S_8, \dots, S_{11}$ in 8-round	$RK_{8,8}$, $RK_{8,9}$, $RK_{8,10}$, $RK_{8,11}$	(59,51,63), (56,48,60), (57,49,61), (58,50,62)	Zero-bit check: remaining 1 bit	$2^{17} \times (2^{-1})^4 = 2^{13}$
3	7-round S-box check	$RK_{7,12}$, $RK_{7,14}$, $RK_{7,15}$	input differences of S_{12}, S_{14}, S_{15}	S-box input check	$2^{13} \times (2^{-4})^3 = 2^1$
4	Partial decryption of $S_0, \dots, S_3$in 8-round	$RK_{8,0}$, $RK_{8,1}$, $RK_{8,2}$, $RK_{8,3}$	(40,36), (41,37), (43,39), (42,38)	Zero-bit check: remaining 2 bits	$2^1 \times (2^{-2})^4 = 2^{-7}$
5	7-round S-box check	$RK_{7,9}$, $RK_{7,10}$	Input differences of S_9, S_{10}	S-box input check	$2^{-7} \times (2^{-4})^2 = 2^{-15}$

Therefore, if the above filtering is repeated independently four times, the probability that a wrong round-key candidate passes all four trials is

$$(2^{-15})^4 = 2^{-60} \tag{14}$$

The number of round-key candidates considered in DC1 is 2^{52}. Hence, the expected number of wrong round-key candidates remaining after four filtering trials is

$$2^{52} \times 2^{-60} = 2^{-8} \tag{15}$$

Since this value is sufficiently small, DC1 determines the target 52 bits round-key uniquely in the expected sense. Next, the sequential zero-difference filtering in Step 1 is repeated four times. Therefore, its complexity is

$$2^{42} \times 4 = 2^{44} \tag{16}$$

Let T_1denote the computational cost of the subkey-guessing steps in the first filtering trial shown in Table 3. In DC1, thirteen 4-bit subkey parts are guessed. In the j-th step, $2^{4(j+1)}$represents the number of guessed subkey candidates after guessing $j+1$subkey parts, and 2^{e_j}represents the expected number of remaining ciphertext pairs before the corresponding filtering condition. Thus, $T_1$is evaluated as

$$T_1 = \frac{2}{136} \times \sum_{j=0}^{12} 2^{4(j+1)} \times 2^{e_j} \cong 2^{36.1} \tag{17}$$

where

$$(e_0, \dots, e_{12}) = (17,16,15,14,13,9,5,1,-1,-3,-5,-7,-11)$$

Here, factor 2 accounts for the two ciphertexts in each pair, and 136 is used to convert S-box operations into equivalent 8-round encryptions. Since each additional filtering trial reduces the number of wrong round-key candidates by a factor of 2^{-15}, the costs of the second, third, and fourth trials are negligible compared with T_1. Hence, the dominant cost of round-key estimation in DC1 is $T_1 \simeq 2^{36.1}$.

Therefore, the computational complexity of DC1, measured in equivalent 8-round encryptions, is

$$2^{44} + 2^{36.1} \cong 2^{44} \tag{18}$$

DC1 alone leaves 76 bits of (RK_7, RK_8) unknown, and the remaining exhaustive search costs 2^{76}trials. Therefore, DC1 alone is not efficient, and DC2 is combined to reduce the remaining key space.

4.2 Key Recovery using DC2

DC2 also has a 6-round differential characteristic probability of 2^{-49}, but its active S-box positions near the final rounds are different from those of DC1. These different active positions expose different 4-bit subkey parts of $RK_7$and RK_8in the ciphertext-side filtering. From Table 2, DC2 uses plaintext pairs with the following input difference:

$$P \oplus P' = \text{0xA9108000AC008400} \tag{19}$$

From Table 2, the post-MixWord difference in 7-round is

$$\Delta a_{2,7} = 0x0909080840D93012 \tag{20}$$

In DC2, the 4-bit round-key part already determined by DC1 are first used for pre-filtering. Although several 7-round subkeys are known from DC1, a 7-round S-box input-difference condition can be checked only when the corresponding 8-round subkey required for partial decryption is also known. At this stage, only one such 4-bit S-box input-difference condition is available. Under the uniformity assumption, satisfying this 4-bit condition has probability 2^{-4}. Therefore, the first filtering stage reduces the candidate ciphertext pairs from 2^{49} to 2^{45}. The remaining pre-filtering checks are zero-difference checks using the known 8-round subkeys. They are performed sequentially on the surviving pairs. The exponent set {0,1,2,3,4,8,12,16,17,18,19,20} represents the cumulative filtering exponents before each check. In other words, a check with exponent $e$is applied only to pairs that have survived the previous checks with probability 2^{-e}. The entries 0,1,2,3,4 correspond to the first stage ending at 2^{45}remaining pairs, and the final check after exponent 20 imposes an additional 4-bit condition, reducing the remaining pairs to 2^{25}. Therefore, the pre-filtering cost is computed as

$$C_{pre} = \frac{2^{49} \times 2 \times 3}{136} \sum_{e \in \{0,1,2,3,4,8,12,16,17,18,19,20\}} 2^{-e} \cong 2^{45.5} \tag{21}$$

Factors 2 and 3 account for the two ciphertexts in each pair and the partial S-box operations, respectively. After the pre-filtering, the remaining ciphertext pairs are used to estimate the unknown round-key parts. Table 4 summarizes this process for DC2, using the active positions shown in Fig. 3. This phase newly estimates six 7-round 4-bit subkey parts, $RK_{7,1}, RK_{7,3}, RK_{7,5}, RK_{7,7}, RK_{7,8}, RK_{7,11}$, and eight 8-round 4-bit subkey parts, $RK_{8,4}, RK_{8,5}, RK_{8,6}, RK_{8,7}, RK_{8,12}, RK_{8,13}, RK_{8,14}, RK_{8,15}$. Thus, the attack newly determines $(6+8) \times 4 = 56$ bits round-key. For wrong round-key candidates, the expected number of remaining pairs after one filtering trial is 2^{-15}, as shown in the final "Remaining pairs" column of Table 4. We use this expected value as the passing probability of wrong round-key candidates. The probability that a wrong round-key candidate passes one filtering trial is 2^{-15}. Therefore, if the filtering is repeated independently four times, the probability that a wrong round-key candidate passes all four trials is

$$(2^{-15})^4 = 2^{-60} \tag{22}$$

Since the number of newly considered round-key candidates is 2^{56}, the expected number of wrong candidates remaining after four trials is

$$2^{56} \times 2^{-60} = 2^{-4} \tag{23}$$

Since this value is sufficiently small, DC2 determines the target 56 bits round-key uniquely in the expected sense.

Next, we evaluate the complexity of DC2. The pre-filtering is performed four times, and thus its complexity is

$$2^{45.5} \times 4 = 2^{47.5} \tag{24}$$

Let U_1 denote the computational cost of the new round-subkey estimation in the first filtering trial for DC2. In DC2, fourteen 4-bit subkey parts are newly guessed. In the j-th step, $2^{4(j+1)}$ represents the number of guessed subkey candidates after guessing $j+1$ subkey parts, and 2^{f_j} represents the expected number of remaining ciphertext pairs before the corresponding filtering condition.

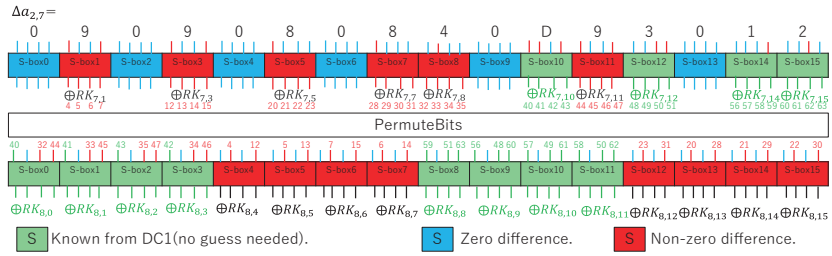


Fig. 3. Boundary-round active positions and estimated 4-bit subkey parts for DC2

Table 4. Filtering process for ciphertext pairs after pre-filtering in DC2

step	Operation	Guess RK	Computed bits	Check	Remaining pairs
1	7-round S-box check	$RK_{7,8}, RK_{7,11}$	$RK_{7,8}$:bit 32-35; $RK_{7,11}$:bit 44-47	S-box input check: $S_8 = 4, S_{11} = 9$	$2^{25} \times (2^{-4})^2 = 2^{17}$
2	Partial decryption in 8-round	$RK_{8,4}, RK_{8,5}, RK_{8,6}, RK_{8,7}$	(4,12), (5,13), (7,15), (6,14)	Zero-bit check: remaining 2 bits	$2^{17} \times (2^{-2})^4 = 2^9$
3	7-round S-box check	$RK_{7,1}, RK_{7,3}$	$RK_{7,1}$:bit 4-7; $RK_{7,3}$:bit 12-15	S-box input check: $S_1 = 9, S_3 = 9$	$2^9 \times (2^{-4})^2 = 2^1$
4	Partial decryption in 8-round	$RK_{8,12}, RK_{8,1}, RK_{8,14}, RK_{8,1}$	(23,31), (20,28), (21,29), (22,30)	Zero-bit check: remaining 2 bits	$2^1 \times (2^{-2})^4 = 2^{-7}$
5	7-round S-box check	$RK_{7,5}, RK_{7,7}$	$RK_{7,5}$:bit 20-23; $RK_{7,7}$:bit 28-31	S-box input check: $S_5 = 8, S_7 = 8$	$2^{-7} \times (2^{-4})^2 = 2^{-15}$

Thus, U_1 is evaluated as

$$U_1 = \frac{2}{136} \times \sum_{j=0}^{13} 2^{4(j+1)} \cdot 2^{f_j} \cong 2^{40.1} \tag{25}$$

where$(f_0, \dots, f_{13}) = (25,21,17,15,13,11,9,5,1,-1,-3,-5,-7,-11)$
Since each additional filtering trial reduces the number of wrong round-key candidates by a factor of 2^{-15}, the costs of the second, third, and fourth trials are negligible compared with U_1. Hence, the dominant cost of the new round-key estimation in DC2 is $U_1 \simeq 2^{40.1}$. Therefore, the computational complexity of DC2, measured in equivalent 8-round encryptions, is

$$2^{47.5} + 2^{40.1} \simeq 2^{47.5} \tag{26}$$

Together with the 52 bits obtained from DC1, the additional 56 bits obtained from DC2 determine 108 bits of RK_7, RK_8 by differential filtering.

4.3 Key Recovery Using DC1 and DC2

This subsection describes how the two differential characteristics are combined to reduce the required number of chosen plaintexts. From Tabel 2, let the differences of DC1 and DC2 be

$$\Delta P_1 = 0\text{x}1000400018004200, \Delta P_2 = 0\text{xA}9108000\text{AC}008400 \tag{27}$$

respectively. If the two attacks are performed independently, four data sets are required for each characteristic. Since each data set contains 2^{49}plaintext pairs, the required number of chosen plaintexts is

$$D_{\text{naive}} = 2^{49} \times 2 \times (4 + 4) = 2^{53} \tag{28}$$

To reduce this data complexity, we use the following four-plaintext structure for each base plaintext x:

$$x, x \oplus \Delta P_1, x \oplus \Delta P_2, x \oplus \Delta P_1 \oplus \Delta P_2 \tag{29}$$

This structure simultaneously provides plaintext pairs with input differences ΔP_1 and ΔP_2. Therefore, 2^{49} pairs for each differential characteristic can be obtained from 2^{48} such structures. Since each structure contains four plaintexts, one shared data set requires

$$4 \times 2^{48} = 2^{50} \tag{30}$$

chosen plaintexts. Repeating this for four independent data sets gives

$$D = 4 \times 2^{50} = 2^{52} \tag{31}$$

Thus, the proposed data-sharing method reduces the required number of chosen plaintexts from 2^{53} to 2^{52}. The value 2^{52} comes from four independent data sets, each of which consists of 2^{50} chosen plaintexts generated by the four-plaintext structure. Each shared data set provides approximately 2^{49} plaintext pairs for DC1 and 2^{49} plaintext pairs for DC2. However, this only means that the expected number of right pairs is one for each characteristic; it does not guarantee that a right pair always exists. For one differential characteristic, the probability that no right pair appears in one data set is

$$(1 - 2^{-49})^{2^{49}} \cong e^{-1} \tag{32}$$

Thus, one data set contains at least one right pair with probability approximately $1 - e^{-1}$. Since the proposed attack uses four independent data sets, the probability that at least one right pair is obtained for one differential characteristic is

$$1 - \{(1 - 2^{-49})^{2^{49}}\}^4 \cong 1 - e^{-4} \cong 0.982 \tag{33}$$

The proposed key-recovery attack requires right pairs for both DC1 and DC2. Assuming that the right-pair generation events for the two characteristics are independent, the probability that both characteristics provide at least one right pair is

$$(1 - e^{-4})^2 \cong 0.965 \tag{34}$$

Therefore, the overall success probability due to right-pair generation is about 96.5%.

4.4 Total Complexity

From Section 4.1, the computational cost of DC1 is 2^{44} equivalent 8-round encryptions. From Section 4.2, the computational cost of DC2 is $2^{47.5}$ equivalent 8-round encryptions. From Section 4.3, the data-generation cost is 2^{52} encryptions. After these filtering steps, the complete RK_8 and 44 bits of the RK_7 are determined. The remaining unknown 7-round 4-bit subkey parts are $RK_{7,0}$, $RK_{7,2}$, $RK_{7,4}$, $RK_{7,6}$, $RK_{7,13}$. Since these are five 4-bit subkey parts, the remaining exhaustive search requires 2^{20}.

Therefore, the total time complexity is

$$2^{44} + 2^{47.5} + 2^{52} + 2^{20} \simeq 2^{52.06} \tag{35}$$

Therefore, the total computational complexity of the 128-bit master-key recovery attack on 8-round reduced LTLBC, including data generation, is approximately $2^{52.1}$equivalent 8-round encryptions. The memory complexity is dominated by the storage of candidate ciphertext pairs after the initial ciphertext-difference filtering. For each characteristic and each data set, approximately 2^{49} plaintext pairs are generated, and the 32-bit zero-difference condition reduces them to about 2^{17} candidate ciphertext pairs. Therefore, the memory complexity for the filtering phase is about 2^{17}candidate ciphertext pairs when the data are processed in a streaming manner. The attack determines the complete $RK_8$and 44 bits of RK_7, for a total of 108 bits round-key. The remaining 20 bits are determined by exhaustive search. Using these two consecutive round keys (RK_7, RK_8), the 128-bit master key is reconstructed through the inverse key schedule. This reconstruction is possible because the key update of LTLBC is invertible, and the complete pair of 7- and 8-round keys determines the corresponding 128-bit key state. However, the attack applies only to 8-round LTLBC and does not threaten the full 14-round LTLBC.

5 Conclusion

In this paper, we re-evaluated the security of reduced-round LTLBC against differential cryptanalysis. We confirmed that the maximum differential characteristic probability for 7-round is 2^{-64}, which is equal to the uniform probability for a 64-bit block. We also found two 6-round differential characteristics with probability 2^{-49} and used them to construct a key-recovery attack on 8-round LTLBC with success probability of about 96.5%. The attack requires 2^{52} chosen plaintexts and approximately $2^{52.1}$ equivalent 8-round encryptions. This result does not threaten the full 14-round LTLBC. Future work includes investigating whether the attack can be extended to 9-round reduced LTLBC using additional filtering conditions or other differential characteristics.

References

1. W. Sun, L. Li, and X. Huang, “LTLBC: a low-latency lightweight block cipher for internet of things,” Cluster Computing, 2024.
2. A. K. Das, "New Integral Distinguishers and Security Reassessment of LTLBC," in Proceedings of the 22nd International Conference on Security and Cryptography
3. R.Kai, N. Shibayama, and Y. Igarashi, “Searching for the Maximum Differential Characteristic Probability of the Block Cipher LTLBC,” Technical Report of IEICE, Vol. 126, No. 40, ISEC2026-6, pp. 6-13, 2026 (in Japanese)
4. STP, “STP constraint solver,” available at: https://github.com/stp/stp, accessed May 2026.
5. SageMath, “SageMath, theSage Mathematics Software System,” available at: https://www.sagemath.org/, accessed May 2026.

Evaluating the Syzygy Distinguisher Cost for PALOMA and Its Comparison with Classic McEliece

Kim Dong Hyeon, Kim Ye Jin, and Kim Dong Chan

Kookmin University, Seoul, Republic of Korea
{wlswudpdlf31,alice1225kim,dckim}@kookmin.ac.kr

Abstract. The security of McEliece rests on the hardness of decoding generic linear codes and on the indistinguishability of Goppa codes from random linear codes. The syzygy distinguisher, introduced at Eurocrypt 2025, targets the second assumption in subexponential time, and, unlike earlier distinguishers, applies at code-based PQC parameter sizes without parameter restrictions. Such estimates have already been reported for Classic McEliece, so we evaluate the cost of this distinguisher on the parameter sets of PALOMA, for which none has been reported. At the same parameters, we check the conditions under which these estimates are justified: one condition fails for mceliece460896 and PALOMA-192, where the iterative cost underestimates the true operation count, while the Koszul–Wiedemann cost is unaffected. The distinguishing degree relies on an unproved lower bound, so we also recompute the Koszul–Wiedemann cost under the two proven bounds. The comparison uses the cost of information-set decoding (ISD), obtained for these parameter sets from the Esser–Bellini syndrome decoding estimator. The distinguishing cost, in both its iterative and Koszul–Wiedemann variants, exceeds the ISD cost for every parameter set considered; as the two attacks pursue different goals, this gap measures the cost margin the parameters provide between them, not a relative speed.

1 Introduction

The McEliece cryptosystem was proposed in 1978 [9]. It currently underlies the post-quantum cryptography (PQC) schemes Classic McEliece [2] and PALOMA [6, 5]. Both adopt the Niederreiter cryptosystem [11], a variant whose public key is built from the parity-check matrix of the secret code rather than from its generator matrix. The two cryptosystems are equivalent in security [8], and both admit semantic security under standard assumptions [12]. In the Niederreiter cryptosystem an error vector $\mathbf{e}$ of weight t is encrypted under the public key $\mathbf{H}_{\text{public}} = \mathbf{S}\mathbf{H}_{\text{secret}}\mathbf{P}$ to produce a syndrome $\mathbf{s} = \mathbf{H}_{\text{public}}\mathbf{e}$, where $\mathbf{H}_{\text{secret}}$ is the parity-check matrix of the secret code, $\mathbf{S}$ a random invertible matrix and $\mathbf{P}$ a random permutation matrix. Classic McEliece and PALOMA use this syndrome computation inside a key-encapsulation mechanism, drawing $\mathbf{e}$ uniformly at random among the vectors of weight t and deriving the session key from it

by hashing. A legitimate receiver, knowing the algebraic structure hidden within the code, decodes the syndrome to recover the error vector. An attacker, by contrast, is assumed to be unable to discern that structure from the public key alone; both schemes publish the systematic form of the public parity-check matrix rather than $\mathbf{SH}_{\text{secret}}\mathbf{P}$ itself, which does not affect the analysis below.

The security of McEliece rests on two assumptions. The first is that the decoding problem for a general linear code (the syndrome decoding problem, SDP) is computationally hard; the representative attack targeting this assumption is Information Set Decoding (ISD). The second is that Goppa codes are computationally indistinguishable from general linear codes (the Goppa distinguishing problem, GD). The two assumptions address different attack goals: SDP corresponds to a message-recovery attack, while GD corresponds to the decisional variant of a key-recovery attack. OW-CPA security holds as long as both assumptions are hard. Breaking GD is a decisional statement and removes the theoretical justification for the assumption without by itself yielding a key recovery or lowering the security level, so its cost is read as a margin rather than as an attack cost.

This paper concerns the second assumption. The relevant attack for this assumption is a distinguisher, which determines whether a given public key comes from a code with a particular structure or from a random code. The existence of a distinguisher does not immediately yield a key-recovery attack. The starting point of this line of research is the distinguisher for alternant and Goppa codes presented in [4], later reinterpreted as a statement about square codes in [13], which related the dimension it observes to the dimension of the square of the dual of the public code, and given a first rigorous analysis in [10], which proved a tight upper bound on that dimension. It was effective only in a limited parameter range, however, and did not reach parameters at the Classic McEliece level.

The syzygy distinguisher, introduced at Eurocrypt 2025, can be viewed as an extension of the square-code distinguisher [14]. Whereas the square-code distinguisher observes the product structure of the code and its relation space at the quadratic level, the syzygy distinguisher examines the higher-degree structure above that relation space. It distinguishes in sub-exponential time, and applies to parameters at the Classic McEliece level.

Classic McEliece and PALOMA have so far used the ISD cost as the baseline for choosing parameters. The syzygy distinguisher makes it necessary to weigh the cost of attacking the Goppa indistinguishability assumption as well, together with how much margin the current parameters leave between that cost and the ISD cost. To our knowledge no such cost has been reported for PALOMA, a code-based KpqC scheme [7], and the ISD cost and the syzygy distinguisher cost have not been compared on a common basis at the parameters of either scheme; [14] reports distinguishing costs for Classic McEliece, but relates them to the ISD cost only asymptotically.

This paper evaluates the syzygy distinguisher cost at the parameters of Classic McEliece and PALOMA. We first check the two conditions under which the distinguisher's heuristic for a random code is justified; the second fails for

mceliece460896 and PALOMA-192, where the iterative cost underestimates the true operation count, while the Koszul–Wiedemann cost, which does not use that condition, is unaffected. We then compute the distinguishing cost for both schemes from the same cost expressions and the same choice of lower bound on the distinguishing degree; the PALOMA costs are reported here for the first time. The same computation shows how the parameters n, m and t determine the cost through the extent to which the code can be shortened. We also recompute the cost under the two proven lower bounds, so that the evaluation does not rest on the tight bound, which has no complete proof. Finally, we set the distinguishing cost against the cost of the generic decoding attack, obtained for both schemes from the same syndrome decoding estimator.

2 Preliminaries

This section fixes the notation and states the background required for the syzygy distinguisher.

2.1 Linear Codes

Throughout this paper we use the column-vector convention. An $[n, k]$-code $\mathsf{C} \subseteq \mathbb{F}^n$ over a finite field $\mathbb{F}$ is a k-dimensional subspace, written as an $[n, k]_q$-code when $\mathbb{F} = \mathbb{F}_q$; its dual code is denoted D and has dimension $k^\perp = n - k$. A parity-check matrix of C is a matrix $\mathbf{H} \in \mathbb{F}^{k^\perp \times n}$ with $\mathsf{C} = \{\mathbf{c} \in \mathbb{F}^n : \mathbf{Hc} = \mathbf{0}\}$; its transpose $\mathbf{H}^\top \in \mathbb{F}^{n \times k^\perp}$ is a generator matrix of D, that is, its columns form a basis of D. The Schur product $\mathbf{ab}$ of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$ is their componentwise product, and the powers of C are defined inductively by $\mathsf{C}^{\langle 0 \rangle} = \mathbb{F} \cdot \mathbf{1}$ and $\mathsf{C}^{\langle r \rangle} = \langle \mathbf{cc}' : \mathbf{c} \in \mathsf{C}^{\langle r-1 \rangle},\ \mathbf{c}' \in \mathsf{C} \rangle_{\mathbb{F}}$.

Fix a finite field extension $\mathbb{F}_{q^m}$, a support vector $\mathbf{x} = (\alpha_0, \ldots, \alpha_{n-1}) \in (\mathbb{F}_{q^m})^n$ whose components are pairwise distinct, and a multiplier vector $\mathbf{y} = (y_0, \ldots, y_{n-1}) \in (\mathbb{F}_{q^m})^n$ whose components are all nonzero. The generalized Reed–Solomon code (GRS code) of order $t \leq n$ is $\mathsf{GRS}_t(\mathbf{x}, \mathbf{y}) = \{\mathbf{y} f(\mathbf{x}) : f(X) \in \mathbb{F}_{q^m}[X]_{<t}\}$, an $[n, t]$-code over $\mathbb{F}_{q^m}$. The alternant code of order t over $\mathbb{F}_q$ is $\mathsf{Alt}_t(\mathbf{x}, \mathbf{y}) = \mathsf{GRS}_t(\mathbf{x}, \mathbf{y})^\perp \cap (\mathbb{F}_q)^n$, with parameters $[n, \geq n - mt]_q$ in general [14]. When $g(X) \in \mathbb{F}_{q^m}[X]$ does not vanish at any component of $\mathbf{x}$, the Goppa code is $\mathsf{Gop}(\mathbf{x}, g) = \mathsf{Alt}_{\deg(g)}(\mathbf{x}, g(\mathbf{x})^{-1})$.

We primarily work with the dual alternant family $\mathsf{Alt}^\perp_{q,m,n,t}$ and the dual Goppa family $\mathsf{Gop}^\perp_{q,m,n,t}$. When $\mathsf{D} \in \mathsf{Alt}^\perp_{q,m,n,t}$ is proper, that is when $k^\perp = mt$, its scalar extension decomposes as [14]

$$\mathsf{D}_{\mathbb{F}_{q^m}} = \mathsf{GRS}_t(\mathbf{x}, \mathbf{y}) \oplus \mathsf{GRS}_t(\mathbf{x}^q, \mathbf{y}^q) \oplus \cdots \oplus \mathsf{GRS}_t(\mathbf{x}^{q^{m-1}}, \mathbf{y}^{q^{m-1}}).$$

For $s \geq 0$, an s-shortened subcode D_s of D consists of the codewords vanishing on s chosen coordinates, with those coordinates deleted; in general it is an $[n_s, k_s^\perp]$-code with $n_s = n - s$ and $k_s^\perp = k^\perp - s$.

2.2 Graded Betti Numbers of a Code

Regard the polynomial ring $S = \mathbb{F}[X_1, \ldots, X_{k^\perp}]$ as graded by total degree, and let $\mathbf{c}_1, \ldots, \mathbf{c}_{k^\perp}$ be the columns of $\mathbf{H}^\top$, that is, the basis of D carried by the rows of $\mathbf{H}$. Sending X_i to $\mathbf{c}_i$ gives the evaluation map $\mathrm{ev}_{\mathbf{H}} : S \to \mathbb{F}^n$, $f \mapsto f(\mathbf{c}_1, \ldots, \mathbf{c}_{k^\perp}) = (f(\mathbf{p}_1), \ldots, f(\mathbf{p}_n))$, where $\mathbf{p}_1, \ldots, \mathbf{p}_n \in \mathbb{F}^{k^\perp}$ are the columns of $\mathbf{H}$. Writing S_r for the space of homogeneous polynomials of degree r in S, the restriction of $\mathrm{ev}_{\mathbf{H}}$ to S_r yields the following map and its kernel:

$$S_r \longrightarrow \mathsf{D}^{\langle r \rangle}, \qquad I_r(\mathsf{D}) = \mathrm{Ker}(S_r \to \mathsf{D}^{\langle r \rangle}).$$

For a finitely generated graded S-module M_0, write F_i for the free S-module on a minimal homogeneous generating set of M_i and let $M_{i+1} = \mathrm{Ker}(F_i \to M_i)$ be the $(i+1)$-st syzygy module; iterating yields the minimal free resolution $\cdots \to F_1 \to F_0$ of M_0. In this paper we set M_0 to be the homogeneous coordinate ring $\mathsf{D}^{\langle \cdot \rangle} = \bigoplus_{r \geq 0} \mathsf{D}^{\langle r \rangle}$ of the dual code D. The number of degree-j elements in the minimal generating set of M_i is called the (i, j)-th graded Betti number, written $\beta_{i,j}$. These numbers are usually displayed in the Betti diagram, the table whose (i, j)-th entry is $\beta_{i,i+j}$.

3 The Syzygy Distinguisher

In this section we recall how the Betti numbers of the linear strand are computed, what values they take for alternant, Goppa, and random codes, and how the distinguisher separates the two families by comparing those values. Dual alternant and dual Goppa codes satisfy more algebraic relations than random codes, and these extra relations persist in the higher-order syzygies that the Betti numbers count.

3.1 Computing Linear-Strand Betti Numbers

Only the first row of the Betti diagram is needed for this distinguisher, and elementary linear algebra suffices to obtain it. The syzygy distinguisher computes $M_{r-1,r}(\mathsf{D})$ iteratively, where M_i is the i-th syzygy module of the minimal free resolution and $M_{r-1,r}$ is its degree-r component.

Let $\mathbf{c}_1, \ldots, \mathbf{c}_{k^\perp}$ be the rows of the parity-check matrix $\mathbf{H} \in \mathbb{F}^{k^\perp \times n}$. The map sending the degree-2 monomial $X_a X_b$ to $\mathbf{c}_a \mathbf{c}_b$ has left kernel $M_{1,2} = I_2(\mathsf{D})$, and a basis B_2 of that kernel satisfies $|B_2| = \beta_{1,2}$. The procedure extends in the same way to $r \geq 3$: given a basis B_{r-1} of $M_{r-2,r-1}$, multiplying each of its elements σ by all variables $X_1, \ldots, X_{k^\perp}$ gives the linear map

$$\begin{aligned} \varphi_r : M_{r-2,r-1} \otimes S_1 &\longrightarrow M_{r-2,r} \\ \sigma \otimes X_a &\longmapsto X_a \sigma, \end{aligned}$$

whose kernel is $M_{r-1,r}$, of dimension $\beta_{r-1,r}$. For $r \geq 4$ the same kernel is cut out by the map $\psi_r : M_{r-2,r-1} \otimes S_1 \to M_{r-3,r-2} \otimes S_2$, obtained by tensoring the

inclusion $M_{r-2,r-1} \subseteq M_{r-3,r-2} \otimes S_1$ with S_1 and composing with the multiplication $S_1 \otimes S_1 \to S_2$. Since $\ker(\varphi_r) = \ker(\psi_r)$, either map yields $\beta_{r-1,r}$, and it is ψ_r that is amenable to computation.

These computations are carried out on matrices: $\mathbf{M}_2$ is the squared matrix of $\mathbf{H}$, $\mathbf{M}_3$ the degree-3 Macaulay matrix of B_2, and $\mathbf{M}_r$ for $r \geq 4$ the blockwise Macaulay matrix realizing ψ_r. Each Betti number is the left kernel dimension of one matrix and is obtained from its rank. The matrix $\mathbf{S}$ acts as a linear change of coordinates in $X_1, \ldots, X_{k^\perp}$, and $\mathbf{P}$ reorders the coordinates. The Betti numbers are therefore unchanged, so the syzygy distinguisher remains valid when given $\mathbf{H}_{\text{public}} = \mathbf{S}\mathbf{H}_{\text{secret}}\mathbf{P}$.

3.2 Betti Numbers of Alternant and Goppa Codes

A proper dual alternant code $\mathsf{D} \in \mathsf{Alt}^{\perp}_{q,m,n,t}$ extended to $\mathbb{F}_{q^m}$ gives a code $\mathsf{D}_{\mathbb{F}_{q^m}}$ that decomposes as a direct sum of GRS codes. Its $k^\perp = mt$ basis vectors are $(\mathbf{y}\mathbf{x}^a)^{q^u}$ for $0 \leq a \leq t-1$ and $0 \leq u \leq m-1$. The space $I_2(\mathsf{D}_{\mathbb{F}_{q^m}})$ of quadratic relations contains the 2×2 minors of a $2 \times f$ matrix in the variables of S [14], where

$$f = (e+1)t - (q^{e+1} - 1)/(q-1), \qquad e = \lfloor \log_q(t-1) \rfloor.$$

For its s-shortened subcode D_s, the following inequality and lower bound hold for all $r \geq 2$:

$$\beta_{r-1,r}(\mathsf{D}_s) \geq (r-1)\binom{f-s}{r}, \qquad r_{\max}(\mathsf{D}_s) \geq f - s,$$

where $r_{\max}(\mathsf{C}) = \max\{r : \beta_{r-1,r}(\mathsf{C}) > 0\}$. Goppa codes are a special case of alternant codes, so this bound applies to them directly. Binary separable Goppa codes carry more quadratic relations and admit a stronger bound, with the following values in place of f and e:

$$\widehat{f} = (2\widehat{e}+2)t - (4^{\widehat{e}+1} - 1)/3, \qquad \widehat{e} = \lfloor \log_4(2t-1) \rfloor.$$

For binary separable Goppa codes the syzygy distinguisher improves the bound once more, replacing $\widehat{f}$ by

$$r^* = (a+2)t - 2^a, \qquad a = \lceil \log_2 t \rceil,$$

so that $r_{\max}(\mathsf{D}_s) \geq r^* - s$. The original paper states this bound without proof and defers the proof to future work. It further reports the bound to be tight in most of its examples, that is $r_{\max}(\mathsf{D}_s) = r^* - s$, as an experimental observation.

3.3 Betti Numbers of Random Codes

A code C has regularity 2 when $\mathsf{C}^{\langle 2 \rangle} = \mathbb{F}^n$, a condition the codes considered by the syzygy distinguisher satisfy. Their Betti numbers are computed with the maps φ_r, whose index is

$$\operatorname{ind}(\varphi_r) = \dim\ker(\varphi_r) - \dim\operatorname{coker}(\varphi_r) = \left(\frac{k(k+1)}{r} - n\right)\binom{k-1}{r-2}.$$

The critical value at which the sign of the index changes is $r_{\mathrm{crit}}(\mathsf{C}) = k(k+1)/n$. If $r < r_{\mathrm{crit}}$, then $\beta_{r-1,r} \geq \mathrm{ind}(\varphi_r) > 0$ is forced for any code of regularity 2, so $\beta_{r-1,r} = 0$ becomes possible only when $r > r_{\mathrm{crit}}$. For a random code the syzygy distinguisher expects φ_r to have zero rank defect, on the heuristic that a linear map with no special structure attains the largest rank its dimensions allow. Below r_{crit} this makes $\beta_{r-1,r}$ equal to $\mathrm{ind}(\varphi_r)$, and above it gives $\ker(\varphi_r) = 0$. The original paper does not fully prove this assumption but reports experimental evidence in its support.

The heuristic does not hold for all r. The minimum distance $d = \mathrm{d}_{\min}(\mathsf{C})$ forces $\beta_{r-1,r} > 0$ for $r \leq k+1-d$, and the dual minimum distance $d^{\perp} = \mathrm{d}_{\min}(\mathsf{C}^{\perp})$ is observed experimentally to give $\beta_{r-2,r} > 0$ for $r \geq d^{\perp}$ [14]. The heuristic is then stated for an $[n,k]_q$-code whose length satisfies the premise $k + \sqrt{2k} + 2 < n < \binom{k+1}{2}$; the lower bound keeps n clear of the range $n = k + O(\sqrt{k})$ in which the minimal resolution conjecture is known to fail, and the upper bound, $\binom{k+1}{2}$ being the number $\dim S_2$ of degree-2 monomials in k variables, is what secures regularity 2. Under this premise there are two conditions, each supporting one conclusion.

1. $d > k + 1 - r_{\mathrm{crit}}(\mathsf{C})$ makes a random code satisfy $\beta_{r-1,r} = 0$ with high probability for $r > r_{\mathrm{crit}}(\mathsf{C})$.
2. $d^{\perp} > r_{\mathrm{crit}}(\mathsf{C})$ makes a random code satisfy $\beta_{r-1,r} = \mathrm{ind}(\varphi_r)$ with high probability for $r < r_{\mathrm{crit}}(\mathsf{C})$.

In the asymptotic analysis and in the parameter tables, d and $d^{\perp}$ are replaced by the typical distances $\mathrm{d}_{GV}(q,n,k)$ and $\mathrm{d}_{GV}(q,n,n-k)$ of random codes. If the premise fails, the heuristic no longer supports either conclusion. Condition 1 justifies the decision between the two code types and truncates the sum that defines κ_{iter}, and condition 2 justifies the matrix sizes that κ_{iter} assumes.

3.4 The Distinguishing Procedure

For a shortened code D_s, the distinguisher operates on the following principle. Suppose that for alternant or Goppa codes there exists an r^* such that $r_{\max}(\mathsf{D}_s) \geq r^* - s$ holds for all s. The distinguisher chooses s to satisfy

$$r_{\mathrm{crit}}(\mathsf{D}_s) = \frac{k_s^{\perp}(k_s^{\perp}+1)}{n_s} < r^* - s,$$

and sets $r = r^* - s$. An alternant or Goppa code then has $\beta_{r-1,r}(\mathsf{D}_s) > 0$. A random code with the same parameters has $r > r_{\mathrm{crit}}$ and, if the distance conditions hold, is expected by the heuristic to have $\beta_{r-1,r} = 0$. The distinguisher therefore decides between the two code types by checking whether $\beta_{r-1,r}(\mathsf{D}_s)$ is zero.

The Betti number $\beta_{r-1,r}(\mathsf{D}_s)$ is obtained as the dimension of the left kernel of the matrix $\mathbf{M}_r$, which is computed by building $\mathbf{M}_r$ and determining its rank

by Gaussian elimination. The complexity of the syzygy distinguisher using this iterative approach is

$$\kappa_{\text{iter}} = \sum_{i=4}^{\lfloor r_{\text{crit}}(\mathsf{D}_s)\rfloor+1} \max\left(k_s^{\perp}\,\text{ind}(\varphi_{i-1}), \binom{k_s^{\perp}+1}{2}\,\text{ind}(\varphi_{i-2})\right)^{\omega},$$

where $\omega \approx 2.372$ is the linear algebra complexity exponent.

The iterative approach computes every intermediate Betti number, whereas the distinguisher needs only to know whether $\beta_{r-1,r}$ is zero at the target r. Koszul cohomology expresses $M_{r-1,r}(\mathsf{D})$ directly as the kernel of a single map induced by the Koszul complex. The decision therefore reduces to testing that kernel for a nontrivial element, a task the Wiedemann algorithm performs. As shown above, an alternant or Goppa code has $\beta_{r-1,r}(\mathsf{D}_s) > 0$ and therefore a nontrivial element, while a random code is expected to have none. This is the Koszul–Wiedemann method, and its complexity is

$$\kappa_{\text{KW}} = O\left(n_s^2\binom{k_s^{\perp}}{r-2}\left((r-1)^2\binom{k_s^{\perp}+1}{r} + \binom{k_s^{\perp}}{r-2}\right)\right).$$

4 Evaluation Results and Discussion

This section evaluates the syzygy distinguisher on the Classic McEliece and PALOMA parameter sets. We first check the conditions under which the distinguisher applies, and identify the sets for which they fail. We then compute the distinguishing cost under the heuristic bound r^* and under the two proven bounds f and $\widehat{f}$, and explain how the parameters n, m and t determine it. Finally, we compare that cost with the ISD decoding cost. The PALOMA costs are reported here for the first time. The Classic McEliece distinguisher costs already appear in [14]; we recompute them with the same procedure we apply to PALOMA, so that the two families can be compared directly.

Table 1 shows the parameters used in the evaluation, taken from the Classic McEliece [2] and PALOMA [6] specifications. For Classic McEliece, the f-variants use the same parameters (m, n, t, k), so only the non-f parameter sets are considered. Here $k = n - mt$ is the code dimension and $R = k/n$ its rate. The distinguisher takes a generator matrix of the dual code as its input, which is exactly the public key $\mathbf{H}_{\text{public}}$, so the table also lists the dual dimension $k^{\perp} = mt$ and the dual rate $R^{\perp} = k^{\perp}/n$.

4.1 Validity of the Distinguishing Conditions

Section 3.3 states two conditions under which the heuristic $\beta_{r-1,r} = 0$ for a random code is justified. Both rest on the premise $k_s^{\perp} + \sqrt{2k_s^{\perp}} + 2 < n_s < \binom{k_s^{\perp}+1}{2}$, which the values n_s and $k_s^{\perp}$ of Table 3 satisfy for all parameter sets. Table 2 evaluates the two conditions themselves, where $\mathrm{d}_{GV}(q, n, k)$ is the smallest d satisfying $\sum_{i=0}^{d}\binom{n}{i}(q-1)^i > q^{n-k}$.

Table 1. Classic McEliece/PALOMA parameter sets

Scheme	Security level	m	n	t	k	R	$k^{\perp}$	$R^{\perp}$
mceliece348864	L1	12	3488	64	2720	0.780	768	0.220
PALOMA-128	L1	13	3904	64	3072	0.787	832	0.213
mceliece460896	L3	13	4608	96	3360	0.729	1248	0.271
PALOMA-192	L3	13	5568	128	3904	0.701	1664	0.299
mceliece6688128	L5	13	6688	128	5024	0.751	1664	0.249
mceliece6960119	L5	13	6960	119	5413	0.778	1547	0.222
mceliece8192128	L5	13	8192	128	6528	0.797	1664	0.203
PALOMA-256	L5	13	6592	128	4928	0.748	1664	0.252

Table 2. Distance conditions at the parameters of Table 3

Parameter set	Condition 1 $\mathrm{d}_{GV}(q, n_s, k_s^{\perp}) > k_s^{\perp} + 1 - r_{\mathrm{crit}}$	Condition 2 $\mathrm{d}_{GV}(q, n_s, n_s - k_s^{\perp}) > r_{\mathrm{crit}}$
mceliece348864	$932 > 321.1$	$50 > 42.86$
PALOMA-128	$1040 > 385.7$	$62 > 55.25$
mceliece460896	$1092 > 513.5$	$89 < 92.47$
PALOMA-192	$1248 > 641.2$	$114 < 125.81$
mceliece6688128	$1692 > 641.3$	$102 > 93.69$
mceliece6960119	$1869 > 605.8$	$92 > 76.21$
mceliece8192128	$2306 > 641.3$	$93 > 69.74$
PALOMA-256	$1653 > 641.2$	$103 > 95.77$

Condition 1 holds for all parameter sets, so the distinguisher applies to every one of them. Condition 2 fails for mceliece460896 and PALOMA-192, and holds for the remaining sets.

Written in terms of the dual rate $R_s^{\perp} = k_s^{\perp}/n_s$, condition 1 holds when $R_s^{\perp} < 0.277$ and condition 2 when $R_s^{\perp} < 0.141$. Both boundaries follow from the q-ary entropy approximation of the Gilbert–Varshamov distance [14]. Before shortening, mceliece460896 and PALOMA-192 have the two largest dual rates $R^{\perp}$ among all sets, namely 0.271 and 0.299 against at most 0.252 for the others. Shortening lowers the dual rate, but for these two it stops at 0.153 and 0.164, whereas every other set falls to at most 0.130.

The consequence is confined to κ_{iter}, which uses condition 2 to substitute $\mathrm{ind}(\varphi_r)$ for the graded Betti number in sizing the matrices; κ_{KW} does not use the condition at all. Since a graded Betti number is the kernel dimension of φ_r, the bound $\beta_{r-1,r}(\mathsf{D}_s) \geq \mathrm{ind}(\varphi_r)^+$ holds for every code and every $r \geq 2$ without any heuristic, so where condition 2 fails, κ_{iter} underestimates the true operation count rather than overestimating it. Only the matrix sizes are affected, as the range of degrees summed is fixed by condition 1, which holds for both sets.

4.2 Computing Syzygy Distinguisher Costs

The syzygy distinguisher cost is measured by $\log_2 \kappa_{\text{iter}}$ and $\log_2 \kappa_{\text{KW}}$. Table 3 shows the syzygy costs for the Classic McEliece and PALOMA parameter sets. Among the lower bounds we take r^* to be the tight heuristic one, not the proven bounds f and $\widehat{f}$. In the table, $\log_2 \kappa_{\text{iter}}$ has to be read with the minimum distance condition in mind; costs for which the condition is not satisfied are enclosed in parentheses. Up to rounding, the Classic McEliece rows of Table 3 agree with the estimates reported in [14], which applies the same tight lower bound on r^*. We restate them here so that the Classic McEliece and PALOMA costs come from the same formulas and the same choice of r^*; the PALOMA rows are not reported in [14].

Table 3. Syzygy distinguisher costs for Classic McEliece/PALOMA parameter sets

Parameter set	r^*	s	r	$[n_s, k_s^\perp]$	r_{crit}	$\log_2 \kappa_{\text{iter}}$	$\log_2 \kappa_{\text{KW}}$
mceliece348864	448	405	43	$[3083, 363]$	42.86	469.0	401.5
PALOMA-128	448	392	56	$[3512, 440]$	55.25	591.3	505.4
mceliece460896	736	643	93	$[3965, 605]$	92.47	(909.4)	772.4
PALOMA-192	1024	898	126	$[4670, 766]$	125.81	(1195.3)	1012.5
mceliece6688128	1024	930	94	$[5758, 734]$	93.69	982.4	834.2
mceliece6960119	943	866	77	$[6094, 681]$	76.21	841.1	716.5
mceliece8192128	1024	954	70	$[7238, 710]$	69.74	801.5	682.6
PALOMA-256	1024	928	96	$[5664, 736]$	95.77	996.7	846.0

Both syzygy distinguisher costs grow with n_s, $k_s^\perp$ and r, and the parameters n, m, t act on these three quantities through the dual dimension $k^\perp = mt$ and the bound r^*. Raising one parameter while holding the other two fixed has the following effects.

1. Raising n leaves r^* and $k^\perp$ unchanged and lowers $r_{\text{crit}}(\mathsf{D}_s)$. The condition $r_{\text{crit}}(\mathsf{D}_s) < r^* - s$ then admits a larger s, so the distinguishing degree $r = r^* - s$ falls and both costs fall with it.
2. Raising m leaves r^* unchanged, since r^* depends on t alone, and raises $k^\perp$. A larger $k^\perp$ raises $r_{\text{crit}}(\mathsf{D}_s)$, which forces a smaller s and a larger r, so both costs rise.
3. Raising t raises r^* and $k^\perp$ together. Here $r_{\text{crit}}(\mathsf{D}_s)$ grows with t^2, being a ratio of $k_s^\perp(k_s^\perp + 1)$ to n_s, while $r^* = (a+2)t - 2^a$ grows with $t \log_2 t$; the faster growth of r_{crit} raises r, so both costs rise.

These three effects account for the comparisons at equal security level. At the 128-bit level, mceliece348864 uses $m = 12$ against PALOMA-128's $m = 13$, while t and hence r^* are equal, so the second effect gives Classic McEliece the lower cost. At the 256-bit level, mceliece6688128 and PALOMA-256 share $(m, t) =$

$(13, 128)$ and differ only in n, 6688 against 6592; the first effect then gives the larger n the lower cost. In both comparisons the mechanism is the extent to which the code can be shortened, and greater shortening leads to a lower syzygy distinguisher cost.

4.3 Costs under the Proven Lower Bounds

The value r^* used above is the tight heuristic bound, for which no complete proof is available. To show that the evaluation does not rest on it, we recompute the costs with the two proven lower bounds f and $\widehat{f}$. All three depend only on t, and $f < \widehat{f} < r^*$ holds for every parameter set considered.

Table 4. Koszul–Wiedemann costs under the three lower bounds on r^*

Parameter set	lower bound			$\log_2 \kappa_{\mathrm{KW}}$		
	f	$\widehat{f}$	r^*	f	$\widehat{f}$	r^*
mceliece348864	321	427	448	718.2	452.3	401.5
PALOMA-128	321	427	448	824.9	549.5	505.4
mceliece460896	545	683	736	1346.1	915.8	772.4
PALOMA-192	769	939	1024	1834.5	1261.6	1012.5
mceliece6688128	769	939	1024	1503.6	1037.7	834.2
mceliece6960119	706	867	943	1268.0	873.9	716.5
mceliece8192128	769	939	1024	1226.0	847.2	682.6
PALOMA-256	769	939	1024	1523.4	1054.3	846.0

A smaller bound admits a smaller shortening order s and hence a larger distinguishing degree $r = r^* - s$, so the cost rises. The cost obtained with r^* is accordingly the smallest of the three. For the Classic McEliece sets, the $\widehat{f}$ and r^* columns reproduce the costs of [14]; the f column and the PALOMA rows are not reported in [14].

4.4 Comparing ISD Cost and Syzygy Distinguisher Cost

The two computational problems in this comparison do not have the same attack goal. ISD measures the cost of a generic decoding attack, which targets message recovery. The syzygy distinguisher, the decisional variant of a key-recovery attack, asks whether a structured code can be distinguished from a random code. The difference between the two costs presented below is therefore not a claim that one attack algorithm runs faster than the other, nor a security level for either attack; it should be read as the cost margin that the parameters provide between the two attacks.

The ISD cost is taken as the BJMM estimate produced by the Esser–Bellini syndrome decoding estimator [3]. For each parameter set the estimator is given

the syndrome decoding problem of length n, dimension $k = n - mt$ and error weight $w = t$. We use the constant memory-access model, in which one memory access costs one unit of time regardless of the memory size. This is the most favorable model for the attacker, so the cost it reports is a lower bound on the true ISD cost. We write this cost κ_{BJMM} and list it in Table 5 next to the distinguishing costs.

Table 5 shows the ISD cost and syzygy distinguisher cost computed for the Classic McEliece and PALOMA parameters. Both κ_{iter} and κ_{KW} are the values obtained with r^*. As in Table 3, the $\log_2 \kappa_{\mathrm{iter}}$ entries enclosed in parentheses are those of the two sets for which condition 2 fails.

Table 5. ISD vs. syzygy distinguisher costs

Parameter set	$\log_2 \kappa_{\mathrm{BJMM}}$	$\log_2 \kappa_{\mathrm{iter}}$	$\log_2 \kappa_{\mathrm{KW}}$
mceliece348864	141.9	469.0	401.5
PALOMA-128	145.3	591.3	505.4
mceliece460896	180.8	(909.4)	772.4
PALOMA-192	219.6	(1195.3)	1012.5
mceliece6688128	246.0	982.4	834.2
mceliece6960119	245.7	841.1	716.5
mceliece8192128	276.8	801.5	682.6
PALOMA-256	244.5	996.7	846.0

In Table 5, the ISD cost is lower than the syzygy distinguisher cost for all parameter sets. The margin is read off κ_{KW} because that is the smaller of the two distinguishing costs and because κ_{KW} does not rely on condition 2.

5 Conclusion

We evaluated the syzygy distinguisher cost at the parameters of Classic McEliece and PALOMA and set it against the ISD cost. The distinguishing cost exceeds the ISD cost for every parameter set, under both the iterative and Koszul–Wiedemann approaches; since the two problems pursue different attack goals, this gap measures the margin the parameters provide, not a direct threat.

A key-recovery attack of heuristic sub-exponential complexity has since been reported [1], although it does not reach parameters at the Classic McEliece level. That attack likewise exploits the quadratic relations attached to the Goppa code, the structure from which the bound used here is derived; evaluating its cost at these parameter sets is left for future work.

Acknowledgments. This work was supported by the ICT R&D program of MSIP/IITP. (No. RS-2024-00397105, Development of core technology to produce KCMVP security high level 3 cryptographic module)

References

1. Briaud, P., Lemoine, A., Randriambololona, H., Tillich, J.P.: A heuristic subexponential attack on the McEliece cryptosystem. Cryptology ePrint Archive, Paper 2026/1232 (2026), https://eprint.iacr.org/2026/1232, version dated 12 June 2026
2. Classic McEliece Team: Classic McEliece: Conservative code-based cryptography: Cryptosystem specification. Tech. rep., Classic McEliece Project (Oct 2022), https://classic.mceliece.org/mceliece-spec-20221023.pdf, version dated 23 October 2022
3. Esser, A., Bellini, E.: Syndrome decoding estimator. Cryptology ePrint Archive, Paper 2021/1243 (2021), https://eprint.iacr.org/2021/1243
4. Faugère, J.C., Gauthier-Umaña, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high-rate McEliece cryptosystems. IEEE Transactions on Information Theory **59**(10), 6830–6844 (Oct 2013). https://doi.org/10.1109/TIT.2013.2272036
5. Kim, D.C., Jeon, C.Y., Kim, Y., Kim, M.: PALOMA: Binary separable Goppa-based KEM. In: Esser, A., Santini, P. (eds.) Code-Based Cryptography. LNCS, vol. 14311, pp. 144–173. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-46495-9_8
6. Kim, D.C., Kim, M., Park, D.H., Kim, D.H.: PALOMA: Binary separable Goppa-based KEM specification. Specification, Future Cryptography Design Lab., Kookmin University (Oct 2024), https://www.kpqc.or.kr/, version 1.2, updated October 16, 2024. Submitted to the Korean Post-Quantum Cryptography Competition
7. KpqC: Korean post-quantum cryptography (KpqC) competition. https://www.kpqc.or.kr/ (2026), accessed: August 8, 2026
8. Li, Y.X., Deng, R.H., Wang, X.M.: On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. IEEE Transactions on Information Theory **40**(1), 271–273 (Jan 1994). https://doi.org/10.1109/18.272496
9. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report **42-44**, 114–116 (1978), https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF, Jet Propulsion Laboratory, Pasadena, CA
10. Mora, R., Tillich, J.P.: On the dimension and structure of the square of the dual of a Goppa code. Designs, Codes and Cryptography **91**(4), 1351–1372 (Apr 2023). https://doi.org/10.1007/s10623-022-01153-w
11. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory **15**(2), 159–166 (1986)
12. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Designs, Codes and Cryptography **49**(1-3), 289–305 (Dec 2008). https://doi.org/10.1007/s10623-008-9175-9
13. Pellikaan, R., Márquez-Corbella, I.: Error-correcting pairs for a public-key cryptosystem. Journal of Physics: Conference Series **855**(1), 012032 (Jun 2017). https://doi.org/10.1088/1742-6596/855/1/012032
14. Randriambololona, H.: The syzygy distinguisher. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025. LNCS, vol. 15606, pp. 324–354. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-91095-1_12

Authorship Attribution for Dark-Web Actor Analysis: A Unified, Scale-Aware Comparison with Collective Attribution

Hiroshi Matsubara[1], Shingo Matsugaya[1,2,3], Taichi Aoki[3], and Masaki Hashimoto[1]

[1] Graduate School of Science for Creative Emergence, Kagawa University
hashimoto.masaki@kagawa-u.ac.jp
[2] Trend Micro, Inc.
[3] Japan Cybercrime Control Center

Abstract. On dark-web forums, pseudonymity makes it hard to determine who wrote a post, and prior closed-set authorship identification on such data has mostly remained near ten candidate authors. We compare four methods—character n-gram TF-IDF with a linear classifier, frozen and fine-tuned BERT embeddings, and metric learning with k-nearest neighbors—on two English dark-web corpora, Silk Road 1 and Agora, under a unified protocol with up to 600 authors. The lightweight n-gram method matches or exceeds the neural methods, and its advantage widens with scale (0.367 versus 0.251, 0.238, and 0.109 at 600 authors), echoing our earlier finding on clear-web reviews. Aggregating about twenty posts per author at test time recovers accuracy to about 0.94 even at 600 authors, and de-duplication proves essential, since duplicates inflate accuracy in a method-dependent way.

Keywords: Authorship attribution · Dark web · Threat intelligence · Character n-grams · Collective attribution

1 Introduction

Threat-intelligence (TI) actor analysis seeks to understand the actors behind cyber-criminal activity—their tooling, dealings, and tradecraft. Dark-web forums and marketplaces are a valuable source for this purpose, hosting discussions of vulnerabilities, stolen data, and illicit services. A central obstacle is anonymity: actors operate under pseudonyms, maintain multiple accounts, and change handles over time, so that content- or keyword-based analysis alone often cannot determine *who* authored a given post.

Authorship attribution (AA), which infers a writer from stylistic regularities in text, provides a signal that is partly independent of topic and therefore complements content-based methods. Character-level features are known to be robust for short and noisy web text; pre-trained language models (PLMs) such as BERT have advanced the task; and profile-based approaches that aggregate multiple posts of a known author can strengthen identification. The last point is

especially relevant on the dark web, where individual posts are short and noisy, yet a suspected actor may have authored many of them.

Despite this, work on dark-web authorship has largely centered on authorship *verification* (deciding whether two texts share an author) or on linking authors across the surface and dark web. Closed-set *identification* on dark-web market forums has typically been confined to small candidate sets of around ten authors [15, 4, 25], whereas identification over hundreds to thousands of authors has been studied mainly on non-dark-web corpora [11]; only a few studies examine how accuracy scales with the candidate-set size on dark-web data [19]. Evaluating such large-scale identification with operationally relevant measures, such as top-k screening and test-time aggregation, at candidate-set sizes approaching those of real forums thus remains comparatively underexplored. Two further gaps stand out. First, although topic leakage in cross-topic verification has received attention [21], the effect of duplicated posts on closed-set identification accuracy—and how that effect varies across methods—has, to our knowledge, rarely been quantified, even though dark-web crawls are prone to copy-pasted content. Second, while we have found lightweight stylometric methods effective on clear-web text [16], whether those findings reproduce on real dark-web posts—where the language, medium, and register all differ—has not been established.

We address these gaps through an empirical study on two English dark-web forum corpora drawn from a public data release: Silk Road 1 (SR1), our primary corpus, and Agora, used as a secondary corpus. We compare four representative methods—character n-gram TF-IDF with logistic regression, BERT embeddings with logistic regression, fine-tuned BERT, and metric learning with k-nearest neighbors—and study how identification accuracy behaves as the number of candidate authors grows, how test-time aggregation (collective attribution) affects it, and how duplicated posts inflate it. Our contributions are as follows.

1. **Unified comparison at dark-web scale.** On real English dark-web data, the lightweight TF-IDF character n-gram method with a linear classifier remains competitive with, or superior to, BERT-based and metric-learning methods, and its advantage widens as the candidate set grows. We compare all four methods uniformly across candidate sets of up to 600 authors—an order of magnitude beyond the roughly ten-author identification settings common to prior dark-web work—and find a ranking (e.g., at 600 candidate authors, 0.367 vs. 0.251, 0.238, and 0.109) that reproduces, in a different language and medium, one we previously reported for Japanese clear-web reviews [16].
2. **Accuracy recovery and scale robustness through collective attribution.** Although single-post accuracy declines as the candidate set grows, aggregating a small number of posts per author at test time restores accuracy to a practical level (about 0.94 or higher), and does so even at the largest candidate set we evaluate (600 authors).
3. **Method-dependent inflation from corpus duplication.** Duplicated posts inflate attribution accuracy by an amount that depends on the method—largest for representation- and neighbor-based methods and smallest for the

sparse character n-gram method (on the high-duplication corpus, e.g., +0.31 for metric learning vs. +0.14 for TF-IDF)—whereas a low-duplication corpus is essentially unaffected (within 0.014). Whereas topic leakage in verification has been studied, the method-dependent inflation from duplicated posts in closed-set identification has, to our knowledge, rarely been quantified; our analysis both cautions against evaluating dark-web AA without de-duplication and reinforces the robustness of lightweight features.

The remainder of this paper is organized as follows. Section 2 reviews related work; Section 3 describes the corpora and preprocessing; Section 4 presents the methods; Section 5 reports the experiments and results; Section 6 discusses the implications; Section 7 addresses ethics, limitations, and future work; and Section 8 concludes.

2 Related Work

Authorship attribution has a long history in stylometry, from early studies of disputed documents [17] to modern machine-learning surveys [26, 10]. Character n-gram features are a recurring choice because they capture orthographic and morphological regularities and remain effective on short, noisy text [27, 20]. More recently, pre-trained language models have been applied to the task, either by fine-tuning [3] or by learning author embeddings through contrastive or metric objectives [1, 5], and convolutional character models have been used for short texts [24].

A complementary line of work aggregates multiple texts of an author rather than classifying each text in isolation. Profile-based methods build a single representation per author from concatenated samples [8, 6], and profile-based verification and hybrid instance/profile schemes have been shown to improve robustness [18, 2]. The test-time aggregation (collective attribution) we study follows this lineage, but applies the aggregation at inference over a fixed classifier, which makes it directly usable as an operational knob.

On the dark web specifically, authorship analysis has been pursued mainly for forensic de-anonymization. Spitters et al. [25] combine character n-grams, stylometric, and timestamp features with an SVM for alias and authorship tasks on a Tor marketplace forum, and Ho and Ng [4] link authors across dark-web forums using stylometric fingerprints; both operate at the scale of roughly ten authors. VeriDark [15] provides large verification datasets from dark-web markets (Agora and Silk Road 1), but frames its closed-set identification task as a ten-way problem on a clear-web subreddit. Other studies vote over multiple classifiers [22] or analyze radical forums while varying the number of users [19], and recent work links similar users across the surface and dark web using topic modeling and attribution [23]; domain-specific language models such as DarkBERT [7] have also appeared. Across these studies, closed-set identification on the dark-web market forums themselves has remained small in scale, and operational measures such as top-k screening and test-time aggregation are rarely reported.

Outside the dark web, the difficulty of identification under many candidate authors is well documented: accuracy degrades as the candidate set grows into the hundreds or thousands [11, 14], and the PAN shared tasks have repeatedly examined cross-domain and open-set generalization [9]. A separate concern is evaluation validity: topic leakage between training and test data can inflate verification performance and destabilize rankings [21]. Duplicated posts are a related but distinct source of inflation that, to our knowledge, has not been quantified across methods for closed-set dark-web identification.

In summary, few prior studies combine real dark-web data, closed-set identification at a non-trivial scale, test-time aggregation, top-k evaluation, and lightweight methods in a single unified comparison. This work occupies that gap and asks, in particular, whether a ranking we reported for Japanese clear-web reviews [16] also holds on English dark-web posts—reproducibility across settings rather than transfer.

3 Dataset and Preprocessing

We conduct our study on two English dark-web forum corpora from a public release of cryptomarket forum data with obfuscated usernames [12], itself derived from the corpus of Ladegaard [13]: the discussion forums of Silk Road 1 (SR1) and Agora, two large, now-defunct Tor marketplaces. We use dark-web marketplace forums because they are the actual target of threat-intelligence actor analysis rather than a clear-web proxy, and we use two corpora that differ in scale and quality so that our findings do not rest on a single source. We confirmed with an automatic language detector (lingua) that the posts are overwhelmingly English. Because both corpora were collected before the widespread availability of generative language tools, the posts can reasonably be treated as reflecting each author's natural writing style, consistent with the non-adversarial assumption stated in Section 7. Table 1 summarizes the corpus statistics; SR1 serves as our primary corpus, while Agora is used mainly to study the effect of duplication (Section 5.5).

From each corpus we construct a balanced closed-set identification problem. We first remove very short posts (those of at most 10 characters), which carry little stylistic signal; then, for a target number of authors U and a number of posts per author n, we select U authors who each have at least n posts and sample exactly n posts per author, yielding a balanced U-class dataset. This uniform sampling lets us vary U and n independently. The number of authors meeting a given n decreases as n grows; on SR1, $n = 186$ is the largest per-author count for which a broad range of U—up to roughly 600 authors—can be formed, which sets the upper end of our scaling analysis. Agora yields substantially fewer authors at the same threshold, which, together with its higher duplication rate, is why we adopt SR1 as the primary corpus.

The two corpora differ markedly in the prevalence of duplicated posts: about 2% of SR1 posts are exact duplicates, against about 28% for Agora. Such duplicates—often quoted text or boilerplate—can let a classifier match a test post to an

Table 1. Corpus statistics after our preprocessing (short-post removal and uniform sampling); figures may differ from the original data release. Avg. length is the mean number of words per post; Dup. is the duplicate rate; Max U is the largest candidate-author count for which n=186 posts per author are available.

Corpus	Web	#authors	Avg. length	Dup.	Max U
SR1	Dark	39,954	78	2.1%	∼600†
Agora	Dark	15,197	47	28.3%	176

†The largest such candidate set contains 624 authors.

identical training post, inflating accuracy in a way that does not reflect genuine stylistic discrimination. To make this explicit, we evaluate every configuration both on the raw data and after removing duplicate posts—a duplicate being the second or later occurrence of an identical post body by the same author—and we report both (Section 5.5). Treating raw and de-duplicated results as distinct is, to our knowledge, not yet standard practice in dark-web authorship studies, yet it proves essential to interpreting accuracy correctly.

4 Methods

We compare four authorship-attribution methods that span a spectrum from lightweight sparse features to fine-tuned and metric-learned neural representations, all evaluated under the same closed-set protocol. Given a post, each method outputs a score (or probability) over the U candidate authors and predicts the highest-scoring one.

- **TF-IDF + LR.** Character n-grams (lengths 2–3, vocabulary capped at 10,000) are weighted by sublinear TF-IDF and classified with logistic regression (C=10). This sparse, lightweight model is our primary method and requires no GPU.
- **BERT-Emb + LR.** Fixed sentence representations are extracted from a frozen `bert-base-uncased` model (mean-pooled token embeddings, maximum length 128) and classified with logistic regression (C=1). This isolates the value of pre-trained representations without task-specific fine-tuning.
- **BERT-FT.** The same `bert-base-uncased` model is fine-tuned end-to-end with a U-way classification head (learning rate 2×10^{-5}, batch size 16, up to 10 epochs with early stopping, weight decay 0.01, warmup ratio 0.1, label smoothing 0.1).
- **Metric + kNN.** An author embedding space is learned with a batch-hard triplet objective (margin 0.2, cosine distance), and a query post is attributed by k-nearest-neighbor voting (k=3, cosine distance).

Collective attribution aggregates several posts of the same author at test time while leaving the trained model unchanged; the four methods above, by contrast, attribute each post in isolation. Given b posts written by the same

(unknown) account, we average the per-class predicted probabilities over the b posts and assign the author with the highest averaged probability. The b posts are drawn from a held-out set disjoint from the training data, and we vary the bundle size b to characterize the trade-off between the number of posts required and the resulting accuracy. Because the aggregation operates only at inference, it is a lightweight operational knob applicable in principle to any base method; we report it for TF-IDF (Section 5.4).

We treat identification as a closed-set U-class problem and report Accuracy, macro-averaged F1, and top-k Accuracy, where top-k Accuracy is the fraction of queries for which the correct author appears among the k highest-scoring candidates. We include top-k because, in an actor-screening setting, narrowing a large candidate set to a short ranked list is often more actionable than a single top-1 decision. The data splitting, repeated runs, and random seeds are described in Section 5.1.

5 Experiments and Results

5.1 Experimental Setup

We evaluate closed-set authorship identification on the two dark-web corpora described in Section 3. For a given number of candidate authors U, we sample U authors and, for each, n posts, forming a balanced U-class problem; unless otherwise noted, results are reported on Silk Road 1 (SR1). We study three axes: the number of candidate authors U, the number of posts per author n, and the bundle size b used in collective attribution. For the method comparison we fix $U = 100$ and vary $n \in \{100, 200, 300, 400\}$; for the scaling analysis we fix $n = 186$, the largest n available across the broadest range of U, and vary U from 2 to 600. We report Accuracy, macro-averaged F1, and top-k Accuracy, following Section 4; reported values are averaged over the 5 cross-validation folds, with standard deviations computed across folds (small, e.g., ± 0.007 at $U = 100$). We use 5-fold cross-validation (an 80%/20% train/test split, random seed 42); TF-IDF with logistic regression is implemented in scikit-learn, while the BERT-based and metric-learning methods use PyTorch with HuggingFace Transformers, trained on a single RTX 5070 Ti GPU.

5.2 Method Comparison

Table 2 reports single-post ($b = 1$) Accuracy on SR1 at $U = 100$ for the four methods. The lightweight TF-IDF method with a linear classifier achieves the highest Accuracy at every n, and the remaining order is consistent across n: TF-IDF > BERT-FT > BERT-Emb > Metric+kNN for $n \geq 200$, with fine-tuned BERT overtaking BERT embeddings only once $n \geq 200$ (at $n = 100$ the two are comparable). The margin of TF-IDF over BERT embeddings widens with n, from $+0.074$ at $n = 100$ to $+0.114$ at $n = 400$. Macro-F1 tracks Accuracy closely, and top-5 Accuracy is substantially higher than top-1—for TF-IDF it reaches 0.753

Table 2. Single-post (b=1) Accuracy on SR1 (U=100) across posts-per-author n. The rightmost column reports top-5 Accuracy at n=400, illustrating the screening benefit of returning a short ranked list rather than a single top-1 guess. Best top-1 per column in bold.

	Top-1 Accuracy				Top-5
Method	n=100	n=200	n=300	n=400	(n=400)
TF-IDF (n-gram) + LR	**0.414**	**0.488**	**0.525**	**0.546**	0.753
BERT-FT	0.326	0.414	0.458	0.487	0.721
BERT-Emb + LR	0.340	0.386	0.414	0.432	0.680
Metric + kNN	0.226	0.255	0.267	0.287	0.520

at n=400 (Table 2), versus 0.546 for top-1—which is relevant to the screening use case discussed below. The lightweight method is also the cheapest, with a full $U = 100$ run completing in about 140 seconds, well below the PLM-based methods.

5.3 Scaling with the Number of Authors

Figure 1(a) shows how single-post Accuracy ($n = 186$, $b = 1$) changes as the candidate set grows from $U = 2$ to $U = 600$. All methods degrade with U, as expected. At very small U the PLM-based methods are competitive with or better than TF-IDF (e.g., at $U = 2$, fine-tuned BERT reaches 0.957 versus 0.890 for TF-IDF), but from moderate U upward TF-IDF becomes the strongest method and its relative advantage grows: at $U = 600$ it attains 0.367, against 0.251 (BERT-Emb), 0.238 (BERT-FT), and 0.109 (Metric+kNN).

5.4 Recovery via Collective Attribution

Single-post Accuracy is modest at large U, but it can be recovered by aggregating several posts of the same author at test time. We apply collective attribution to TF-IDF, the strongest and lightest method: we partition each author's held-out posts (disjoint from the training set) into non-overlapping bundles of b, average the classifier's predicted probabilities over the b posts in a bundle, and take the arg-max as the bundle-level prediction; with $n = 186$ and 5-fold splits this yields about $37/b$ bundles per author per fold, over which we report the mean Accuracy. Figure 1(b) shows the result on SR1 across U for several bundle sizes b. Bundling restores Accuracy to a practical level even as U grows: at $U = 100$, Accuracy rises from 0.479 (b=1) to 0.949 (b=10) and 0.982 (b=20); at $U = 600$, from 0.367 (b=1) to 0.854 (b=10) and 0.943 (b=20). For very small candidate sets ($U \leq 7$), Accuracy is already near 1.0 at b=1, so the gain from bundling matters most at large U. Together with the higher top-k Accuracy noted above, this indicates that aggregating on the order of ten to twenty posts per author suffices to screen a several-hundred-author candidate set down to a short, high-precision list.

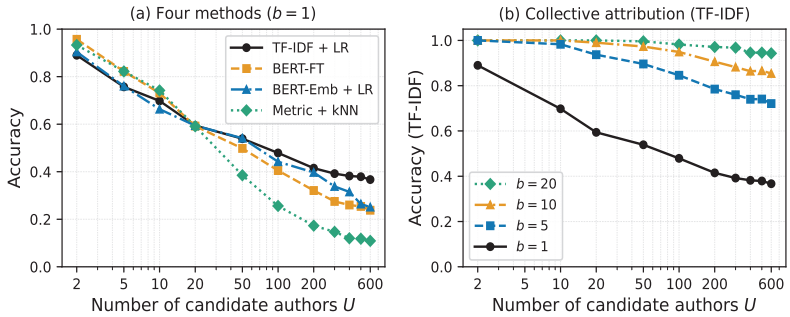


Fig. 1. Identification accuracy on SR1 (n=186) versus the number of candidate authors U. (a) Single-post (b=1) accuracy for the four methods. (b) Recovery by collective attribution (TF-IDF): even at U=600, b=20 reaches 0.943.

Table 3. Effect of de-duplication on SR1 (low duplication, 2%): $\Delta = \mathrm{Acc}_{\mathrm{dedup}} - \mathrm{Acc}_{\mathrm{raw}}$ (U=100, b=1). All within 0.014.

Method	$\Delta_{n=100}$	$\Delta_{n=200}$	$\Delta_{n=300}$	$\Delta_{n=400}$	$\max\lvert\Delta\rvert$
TF-IDF	+0.005	+0.002	−0.003	+0.001	0.005
BERT-Emb	−0.010	−0.006	−0.006	−0.006	0.010
BERT-FT	+0.003	0.000	−0.003	+0.003	0.003
Metric+kNN	0.000	−0.009	−0.014	−0.003	0.014

5.5 Effect of Corpus Duplication

Finally, we quantify how duplicated posts affect Accuracy, contrasting the low-duplication SR1 (2%) with the high-duplication Agora (28%). Table 3 reports the change on SR1 when exact duplicates are removed: across all four methods and all n, the change stays within 0.014, so the SR1 results above are essentially unaffected by duplication and can be read at face value. On Agora, by contrast, de-duplication lowers Accuracy substantially, and the size of the drop depends strongly on the method (Table 4): at $U = 100$, $n = 100$, removing duplicates reduces Accuracy by 0.309 for Metric+kNN, 0.205 for BERT-FT, and 0.192 for BERT embeddings, but only 0.142 for TF-IDF, with the same ordering at $n = 200$. Methods that rely on learned representations or nearest-neighbor matching are thus more susceptible to duplication-driven inflation than the sparse character n-gram method. This carries two implications: dark-web attribution accuracy should be reported after de-duplication to avoid method-dependent inflation, and the relative robustness of the lightweight method to this data-quality issue reinforces the comparison of Section 5.2.

Table 4. Accuracy inflation from duplication on Agora (high duplication, 28%; U=100, b=1). Inflation ↑= raw − dedup; largest for representation/neighbor-based methods.

	n=100			n=200		
Method	raw	dedup	↑	raw	dedup	↑
TF-IDF	0.601	0.459	+0.142	0.643	0.519	+0.123
BERT-Emb	0.545	0.353	+0.192	0.578	0.404	+0.174
BERT-FT	0.547	0.342	+0.205	0.606	0.429	+0.177
Metric+kNN	0.559	0.250	+0.309	0.576	0.279	+0.297

6 Discussion

The central practical takeaway of our study is that authorship attribution can be turned into a workable actor-screening pipeline for dark-web forums, despite the very conditions that make such forums difficult: large candidate sets, heavily duplicated text, and pseudonymous accounts. The pipeline we arrive at is deliberately simple—lightweight character n-gram features, de-duplication as a mandatory preprocessing step, and test-time collective attribution followed by top-k ranking. Its operational behavior is concrete: although single-post accuracy is modest once the candidate set reaches several hundred authors, aggregating about twenty posts per account restores accuracy to roughly 0.94 even at $U = 600$, while around ten posts suffice at smaller candidate sets (Section 5.4); top-k scoring further narrows a large candidate pool to a short, high-precision list. In threat-intelligence terms, this answers a question an analyst actually faces—how much text per suspected account is needed to screen a several-hundred-author pool—with a usable figure: on the order of ten to twenty posts, at negligible computational cost.

Two design choices in this pipeline are justified directly by the data rather than by convenience. First, de-duplication is not optional: Section 5.5 shows that duplicated posts inflate accuracy by a method-dependent amount, so a pipeline that skipped this step would report optimistic numbers that do not reflect genuine stylistic discrimination. Second, the choice of lightweight sparse features is reinforced by the same analysis: the character n-gram method is the least inflated by duplication and degrades the most gracefully as the candidate set grows (Sections 5.2 and 5.3), making it the most dependable component for noisy, large-scale dark-web data while running without a GPU. We note, tentatively, that this robustness may stem from sparse character features being less sensitive than subword-based neural models to the slang, deliberate misspellings, and tokenization irregularities common on these forums; isolating this effect is left to future work (Section 7).

Beyond its operational value, our comparison yields a secondary observation: the ranking obtained here on English dark-web posts is consistent with one we reported for Japanese clear-web reviews [16], despite the change in language, medium, and register. We offer this as corroborating evidence that the ranking

is not an artifact of a single setting—not as the principal contribution—and leave a controlled account of the markedly lower absolute accuracy on dark-web data to future work.

7 Ethics, Limitations, and Future Work

We use only publicly released research data derived from now-defunct dark-web marketplaces (Silk Road 1 and Agora); we neither crawl live forums nor attempt to de-anonymize real individuals, and we report only aggregate accuracy. The intended use of this work is defensive: to help analysts screen and prioritize candidate accounts in threat-intelligence settings, not to produce unilateral attributions. Authorship attribution is dual-use—the same techniques could be misused for surveillance or to unmask whistleblowers and dissidents—so such methods should be deployed only under appropriate legal and ethical oversight.

Our study has five main limitations. It assumes a non-adversarial setting in which posts reflect natural writing style; deliberate obfuscation, including stylistic transformation by large language models, is out of scope, and our corpora predate such tools. We treat only closed-set identification, leaving open-set and unknown-author detection untreated. We do not model actor-level phenomena such as handle changes or multiple accounts per actor. The scale is bounded by data: SR1 supports up to about 600 authors at n=186 (Section 3), so we characterize hundreds, not thousands, of authors. Finally, collective attribution requires several posts per account—which may not hold for low-activity actors—and we report it for TF-IDF only (Section 5.4).

Natural extensions include open-set and unknown-author detection, cross-platform identity linkage across the surface and dark web, a controlled decomposition of where the gains from collective attribution originate (text length, number of posts, and topical overlap), and robustness to adversarial style obfuscation such as paraphrasing by large language models.

8 Conclusion

We studied lightweight authorship attribution on real English dark-web forums, comparing four representative methods under a unified closed-set protocol on the Silk Road 1 and Agora corpora. Three findings stand out. First, a lightweight character n-gram model with a linear classifier matches or exceeds BERT-based and metric-learning methods at all but the smallest candidate sets, and its advantage widens as the candidate set grows toward several hundred authors—making it a dependable, GPU-free default for large-scale screening. Second, although single-post accuracy declines with scale, collective attribution at test time restores it to a practical level (about 0.94 with roughly twenty posts) even at 600 candidate authors, turning the number of aggregated posts into a simple operational knob. Third, duplicated posts inflate accuracy by a method-dependent amount—largest for representation- and neighbor-based methods and smallest

for the sparse model—so that de-duplication is a prerequisite for credible evaluation. As a secondary observation, the method ranking we obtain is consistent with one we previously reported for clear-web reviews, suggesting it is not an artifact of a single language or medium.

Taken together, these results suggest that a lightweight, low-cost pipeline that controls for corpus duplication, combined with test-time aggregation and top-k screening, offers a practical basis for authorship-based actor screening on the dark web. We hope that the unified, scale-aware evaluation and the explicit treatment of duplication presented here will support more reproducible comparisons in future work.

Acknowledgements. We acknowledge the use of Claude AI assistant (Anthropic) for the English translation of this paper. This work was supported in part by the JSPS/MEXT KAKENHI under Grant 24K14956.

References

1. Ai, Q., et al.: Whodunit? learning to contrast for authorship attribution. In: Proceedings of AACL-IJCNLP (2022)
2. Bacciu, A., et al.: Cross-domain authorship attribution combining instance-based and profile-based features. In: CLEF Working Notes (2019)
3. Fabien, M., Villatoro-Tello, E., Motlicek, P., Chakraborty, S.: BertAA: BERT fine-tuning for authorship attribution. In: Proceedings of the 17th International Conference on Natural Language Processing. pp. 127–137 (2020)
4. Ho, T.N., Ng, W.K.: Application of stylometry to darkweb forum user identification. In: Lam, K.Y., Chi, C.H., Qing, S. (eds.) Information and Communications Security. pp. 173–183. Springer International Publishing, Cham (2016)
5. Huertas-Tato, S., Martín, A., Camacho, D.: PART: Pre-trained authorship representation transformer. arXiv preprint arXiv:2209.15373 (2022), https://arxiv.org/abs/2209.15373
6. Jankowska, M., Karkaletsis, V., Stamatatos, E., et al.: Common n-gram profiles for authorship attribution. In: Proceedings of COLING (2014)
7. Jin, Y., Jang, E., Cui, J., Chung, J.W., Lee, Y., Shin, S.: DarkBERT: A language model for the dark side of the Internet. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 7515–7533. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.415, https://aclanthology.org/2023.acl-long.415/
8. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of PACLING. pp. 255–264 (2003)
9. Kestemont, M., Manjavacas, E., Markov, I., Potthast, M., Stamatatos, E., Stein, B.: Overview of the author identification task at PAN-2018: Cross-domain authorship attribution and style change detection. In: CLEF Working Notes (2018)
10. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology **60**(1), 9–26 (2009)
11. Koppel, M., Schler, J., Argamon, S., Messeri, E.: Authorship attribution with thousands of candidate authors. In: Proceedings of SIGIR. pp. 659–660 (2006)

12. Kulatilleke, G.K.: Darkweb & clearweb-darktopics: Darkweb market forum datasets with obfuscated usernames. https://github.com/gayanku/darkweb_clearweb_darktopics (2021)
13. Ladegaard, I.: Open secrecy: How police crackdowns and creative problem-solving brought illegal markets out of the shadows. Social Forces **99**(2), 532–559 (2020). https://doi.org/10.1093/sf/soz140
14. Luyckx, K., Daelemans, W.: The effect of author set size and data size in authorship attribution. Literary and Linguistic Computing **26**(1), 35–55 (2011)
15. Manolache, A., Brad, F., Barbalau, A., Ionescu, R.T., Popescu, M.: Veridark: a large-scale benchmark for authorship verification on the dark web. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
16. Matsubara, H., Matsugaya, S., Aoki, T., Hashimoto, M.: Lightweight Authorship Attribution for Japanese Web Reviews: Character n-grams with Collective Attribution toward Actor Analysis. In: Proceedings of the 21st Asia Joint Conference on Information Security (AsiaJCIS 2026). EPiC Series in Computing, vol. 111, pp. 111–122. EasyChair (2026). https://doi.org/10.29007/1llv
17. Mosteller, F., Wallace, D.L.: Inference and Disputed Authorship: The Federalist. Addison-Wesley (1964)
18. Potha, N., Stamatatos, E.: A profile-based method for authorship verification. In: Artificial Intelligence: Methods and Applications, Lecture Notes in Computer Science, vol. 8445, pp. 313–326. Springer (2014). https://doi.org/10.1007/978-3-319-07064-3_25
19. Ranaldi, L., Ranaldi, F., Fallucchi, F., Zanzotto, F.M.: Shedding light on the dark web: Authorship attribution in radical forums. Information **13**(9) (2022). https://doi.org/10.3390/info13090435, https://www.mdpi.com/2078-2489/13/9/435
20. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. In: Proceedings of NAACL-HLT. pp. 93–102 (2015)
21. Sawatphol, J., Udomcharoenchaikit, C., Nutanong, S.: Addressing topic leakage in cross-topic evaluation for authorship verification (2024), https://arxiv.org/abs/2407.19164
22. Sennewald, B., Herpers, R., Hülsmann, M., Kent, K.B.: Voting for authorship attribution applied to dark web data. In: Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering. p. 217–226. CASCON '20, IBM Corp., USA (2020)
23. Shin, G.Y., Kim, D.W., Park, S., Park, A.r., Kim, Y., Han, M.M.: Identifying similar users between dark web and surface web using bertopic and authorship attribution. Electronics **14**(1) (2025). https://doi.org/10.3390/electronics14010148, https://www.mdpi.com/2079-9292/14/1/148
24. Shrestha, P., Sierra, S., González, F., Montes, M., Rosso, P., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: Proceedings of EACL. pp. 669–674 (2017)
25. Spitters, M., Klaver, F., Koot, G., van Staalduinen, M.: Authorship analysis on dark marketplace forums. In: 2015 European Intelligence and Security Informatics Conference. pp. 1–8 (2015). https://doi.org/10.1109/EISIC.2015.47
26. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology **60**(3), 538–556 (2009)
27. Stamatatos, E.: On the robustness of authorship attribution based on character n-gram features. Journal of Law and Policy **21**(2), 421–439 (2013)

Usage-adaptive cold-start pricing for copyrighted gaming-NFT rental mechanism

Islam Md Shariful[1[0009−0008−4200−2457]], Prof. Hye-Young Kim[2[0000−0001−7013−7510]⋆], Jung-Min Park[1[0009−0001−6751−5020]], Hyeong-Gyun Yun[1[0009−0005−2487−9827]], Won-Yong Lee[1[0009−0002−2516−2199]], Geon Park[1[0009−0003−0417−1704]], Myeong-Jae Lee[1[0009−0000−2419−8590]], Si-Young Nam[1[0009−0008−0616−6323]], Jae-Yeong Woo[1[0009−0002−4668−0008]], and Nguyen Huu Nhat Minh[3[0000−0002−3035−0816]]

[1] Department of Games, Graduate School, Hongik University, South Korea
sharif@mail.hongik.ac.kr, kalsbold@mail.hongik.ac.kr, xprise@naver.com, night98200@gmail.com, qkrrjs0773@gmail.com, audwolego@naver.com, namsy8471@gmail.com , dnwodud2579@gmail.com
[2] Department of Game Software, School of Games, Hongik University, South Korea
hykim@hongik.ac.kr
[3] Department of Science, Technology and International Cooperation, Vietnam-Korea University of Information and Communication Technology, The University of Danang, Da Nang, Vietnam nhnminh@vku.udn.vn

Abstract. Copyrighted gaming NFTs have drawn increasing attention in blockchain-based games, yet setting an initial rental-oriented price for newly introduced assets remains challenging when platform usage history is unavailable. In many cases, creators arbitrarily set sky high prices or otherwise limit accessibility and dampen user interest. Our study presents a gaussian process (GP) based dynamic pricing framework for copyright markets. First, our framework uses prior market knowledge to establish an initial price baseline which then learns from real time platform-session activity and usage driven signals. This helps address the cold start pricing problem linking copyrighted NFT with automated pricing mechanisms which will enable creators to better manage the value of their protected NFT. This opens a practical avenue for both budding NFT copyright platforms and mature marketplaces to move from a direct sale model to a sustainable rental-based ecosystem without compromising asset value.

Keywords: copyrighted NFT · dynamic pricing · cold-start problem. · rental system

1 Introduction

Digital blockchain asset pricing still remains challenging due to the high volatility and sensitivity to changes in demand, user activity, and market sentiment

⋆ Corresponding author: hykim@hongik.ac.kr

of these markets, which makes pricing reliable for newly emerging or sparsely observed assets [Ahmed et al.2024]. The challenge is even greater when it comes to NFT assets associated with copyright ownership or licensing rights, whose value depends not only on market behavior but also on the economic importance of protected digital content. As such assets arrive on a platform without sufficient transaction/usage history, the challenge is one of cold-start, where the system does not have enough information about how to estimate the asset's value [Ye et al.2023]. In such settings, using prior knowledge about related assets is more helpful than treating each asset individually, as history from similar assets provides useful priors for initialization [Zeng and Xiao2024]. Finally, Gaussian process regression is particularly suited for uncertainty-aware modeling as it presents a versatile approach to incorporate and model the predictive uncertainties naturally [Wang and Jing2022].

2 Methodology

We propose that our pricing framework has an initial offline phase followed by an online adaptive phase depicted in Fig 1.

In the offline phase, we estimate type-wise priors from the history of past marketplace data which has asset id, categories and sale price. These prices are altered to make a rental-biased target which retains relative prices. We use a Gaussian Process (GP) model to learn a base price distribution (specified with prior mean μ^t and variance σ^t for making a standard prior over types, which would be of service to solve the cold-start problem.

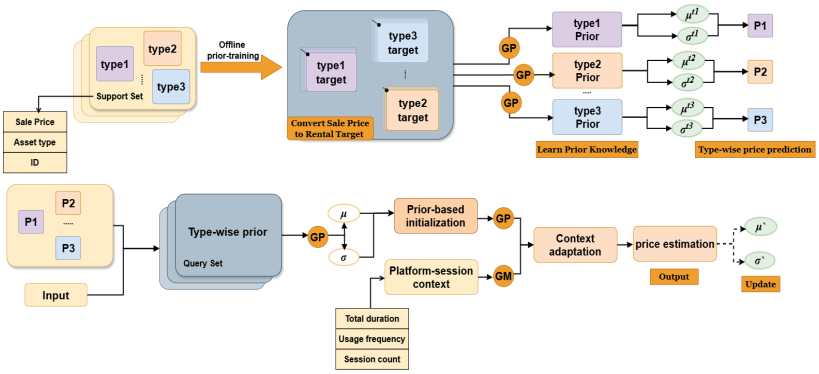


Fig. 1: Model architecture

In the online inference phase, when we receive a new asset query, the framework retrieves the corresponding type-wise prior to initialize an asset-specific statistical baseline (μ, σ). It then combines this prior-based initialization with real-time platform-session context, using session-level behavioural variables (such as

total duration, usage freq., and session count) extracted from same-type peer assets, to finesse the initial estimate. To make this adaptation more robust, we use geometric-mean-based aggregation alongside our GP prior, helping the model reduce the impact of extreme values while retaining important relative pricing information. By adapting this context, we generate a query-specific price estimate reflecting historical priors and in-platform conditions.

We return the inferred price as output and cache the updated statistical state (μ', σ'). Our two-phase design helps address cold-start pricing while remaining adaptive to evolving session-level behaviour in NFT rentals.

2.1 Datasets and Preprocessing

External market data : To build the historical prior, we use the OpenSea NFT Sales dataset, which contains successful OpenSea transactions from January 2019 to December 2021.[4] We transform sale prices into rental-oriented target metrics for the specified asset categories from the dataset. For efficiency and to testing result on limited-data setting, we sample up to 100 unique assets per subset for training. This preprocessing preserves category-level pricing patterns while allowing us to test whether the model can learn an effective prior and make reliable pricing estimates with small training sets.

Target metrics : The OpenSea dataset includes historical sale prices but lacks information on rental transactions, rental durations, or gameplay records for specific assets. The transformed value is not considered an actual market rental price; it serves instead as a rental-oriented target to assess a proposed cold-start pricing mechanism in a theoretical gaming-NFT scenario. To illustrate daily usage, aggregate weekly playtime intervals and cryptocurrency ownership rates from a blockchain gaming survey are utilized [Faridi2022,Efendioğlu2022]. However, these ownership rates are not direct measures of individual asset usage time or rental demand. The study uses group midpoints of 2, 7, 12, and 17 hours per week (Table 1), resulting in a weighted average of $\frac{20.78}{1.94} = 10.71$ hours per week, or about 1.53 hours per day, to define a simulation activity scale.

Table 1: Playtime calculation for blockchain gamers [Faridi2022,Efendioğlu2022]

Play time group	h_i **(h/week)**	w_i	$w_i h_i$
< 4 h/week	2	0.32	0.64
5–9 h/week	7	0.46	3.22
10–14 h/week	12	0.56	6.72
15+ h/week	17	0.60	10.20
Total	–	1.94	20.78

[4] Dataset available at: https://www.kaggle.com/datasets/bryanw26/opensea-nft-sales-2019-2021

If we let α be the estimated average usage time of a blockchain gaming asset during a billing period. Since only total NFT sale prices are recorded, α converts these sales into rental values. The reciprocal coefficient $\gamma = 1/\alpha$ is a time normalization factor for asset i defined as:

$$y_i = \frac{P_i^{\text{sale}}}{\alpha} = P_i^{\text{sale}}\gamma, \qquad \gamma = \frac{1}{\alpha}, \tag{1}$$

where P_i^{sale} is the observed total sale price. This transformation rescales total sale prices to the rental-oriented target scale while preserving the relative price ratio between assets.

Assumptions :

- **A1 - Uniform daily distribution:** The gaming session time on a weekly basis is considered to be uniformly distributed throughout the 7 days in the week (though in reality weekend usage tends to be higher). This allows for the target metric to be interpreted as a steady daily average [Gómez-Gonzalvo et al.2020,Azmi2022].
- **A2 - Full asset utilization:** A rented out NFT is assumed to be used throughout the entirety of the gaming session.
- **A3 - Single asset per session:** A primary NFT asset is assumed to be associated with each player in each session, without an explicit consideration of multi-asset utilization or switching in a single session.
- **A4 - Gaming-scenario applicability:** The target construction is limited to the gaming-NFT scenario studied here, in which asset value is assumed to be related to gameplay activity. It is not directly applicable to digital art, collectible NFTs, or other copyrighted NFTs whose demand and rental value are not determined by playtime.

2.2 Kernel and hyperparameter

For each asset type, a Gaussian Process Regressor (GPR) is trained using normalized usage inputs and transformed rental-oriented targets. We use a Matérn-5/2 kernel with an additive white-noise term:

$$k(x, x') = \sigma_f^2 k_{\text{Matérn}-5/2}(x, x'; \ell, \nu) + k_{\text{White}}(x, x'; \sigma_n^2). \tag{2}$$

Here, σ_f^2 is the signal variance, ℓ is the length scale, $\nu = 2.5$ controls smoothness, and σ_n^2 represents observation noise. The signal variance and length scale are initialized at 1.0 and optimized within $[10^{-5}, 10^5]$ and $[10^{-3}, 10]$, respectively. The noise level is initialized at 10^{-6} and optimized within $[10^{-10}, 10^{-1}]$. Hyperparameters are fitted on normalized targets using ten random restarts. This kernel configuration strikes a balance between sensitivity to price changes and robustness against market noise which is consistent with established GP kernel design practice [scikit learn].

2.3 Training and adaptation mechanisms

Usage score for GP adaptation : The usage of an asset in the session level is represented by a scalar usage score, calculated by combining three behavioral variables, including the overall duration of a session E, frequency of active usage U, and number of sessions R. All of these are likely to be highly skewed, thus are transformed with a $\log(1+x)$ before being combined to limit the impact of outliers and improve learning stability. The formula we use is,

$$S = 0.4\log(1+E) + 0.4\log(1+U) + 0.2\log(1+R). \tag{3}$$

We give more weights to E and U as they are considered as a stronger signal of active usage than R, which acts as a weaker passive signal. This is in line with typical digital engagement modeling, where active behaviors are assumed to provide stronger signals than passive existence [European Commission2026]. More generally, the construction can be understood as a weighted composite score:

$$M = w_1\log(1+v_1) + w_2\log(1+v_2) + ... + w_n\log(1+v_n) \tag{4}$$

where $v_1, v_2, \ldots, v_n$ represents behavioral variable i and $w_1, w_2, \ldots, w_n$ represents weight of variable i. In this model the weights are heuristically set to emphasize the usage rather than passive session presence. This score S is used as a scalar input to the asset-category specific GP in on-line adaptation: The system computes S from current session behavior and adds it to historical behavior sequence of the asset and update the GP with corresponding rental prices to obtain posterior mean μ and variance σ as illustrated in Fig.1. Through this process, the behavioral usage score informs the rental prices of an asset through on-line updates.

Initial pricing via. Geometric mean : In order to get a robust initial price for newly queried asset, the model is using the geometric mean of most recent prices from same-type peer assets. GM has less tendency to deviate in presence of outlier values in skewed distributions [Clark-Carter2010]. Such value selection provides us a more stable asset's price level and less influence on extreme price spikes, which are commonly encountered in NFT secondary market. Initial pricing is formulated as follows:

$$P_{\text{initial}} = \exp\left(\frac{1}{N_{\text{session}}}\sum_{i=1}^{N_{\text{session}}} \ln\left(P_{\text{latest},i}\right)\right) \tag{5}$$

Here, In eq 5, $P_{latest,i}$ denotes the most recent observed price of the i-th peer asset in the same category, and $N_{session}$ is the number of available peer observation. It dampens the effect of unusually high prices and yields a more reliable starting point for cold-start pricing.

Without any peer session data to utilize, the model degenerates to historical training beforehand and utilizes the identical geometric mean formulation as that in eq 6 over training data.

$$P_{\text{fallback}} = \exp\left(\frac{1}{N_{\text{external}}} \sum_{i=1}^{N_{\text{external}}} \ln\left(y_{\text{train},i}\right)\right) \tag{6}$$

When the asset is of the same type to other items but has fewer than 3 in-platform peer-session observations (shown in Equation 6) it is assigned an initial price estimated with external data. We use $y_{\text{train},i}$ as the target value for the i-th asset, rental price. N_{external} is the number of assets with comparable external prices to provide some validation in price or provide backup prices for when insufficient internal data exist in addition to also estimating initial prices where no internal peer-session observation exist.

Time-aware adaption with exponential decay : To capture the present demand more accurately, the model is assigned to give higher weights to recently performed sessions and exponentially decay to older ones. In platforms like NFT or games, the players' behavior and the market environment are prone to change more frequently and if we treat all the history data equally, the latest performance may be washed out, which cause delays for market to reach its equilibrium and reflect to the price. Thus, we selected a half-life interpretation for defining decay, which represents that for a specified time period, the weight of a past session is diminished to a half of its original weight [Luxenberg and Boyd2024]. The usage of half-life make the time decay factor have a more intuitive meaning and on the other hand provides an optimal combination between responsiveness and steadiness. Hence, the raw weight w_i of the i-th session is given by

$$w_{\text{raw},i} = \lambda^{(n-1-i)} \tag{7}$$

In eq 7, n is the total number of sessions, i is the session index in chronological order, and λ is the decay factor which we concludes as 0.5 for the half-life principle. However, The raw decay weights don't sum to one as a function of s. Equation 8 then normalizes the weights over the historical session length for each of the assets for the session period. The effect of this normalization is that, instead of growing in response to increasing record counts of sessions, the produced time aware signal is now a weighted sum of past signals. The weight applied to the most recent session is the highest for $\lambda = 0.5$ while each immediately subsequent prior session has half of that weight as the next session over as defined below:

$$w_i = \frac{w_i^{\text{raw}}}{\sum_{j=1}^{n} w_j^{\text{raw}}}, \qquad \sum_{i=1}^{n} w_i = 1. \tag{8}$$

Finally the time-aware usage-signal for specific peer-asset is computed as in eq 9.

$$U_{\text{signal}} = \mathbf{W}^\top \mathbf{E} = \sum_{i=1}^{n} w_i e_i. \tag{9}$$

Equation 9 computes the exponentially weighted average epoch usage of a peer asset. Let $\mathbf{E} = [e_1, e_2, \ldots, e_n]^\top$ be the epoch-usage vector ordered chronologically, and let $\mathbf{W} = [w_1, w_2, \ldots, w_n]^\top$ be the normalized decay-weight vector. then $U_{\text{signal}} = \mathbf{W}^\top \mathbf{E}$ places increasing weight on the most recent sessions while attenuating the overall contribution of older sessions. This signal is then used in the peer-relative usage ratio calculation to determine pricing.

Peer-relative usage signal for internal market balancing : In platform-level gameplay settings, asset usage should not be measured only against fixed external baselines. Short-term drops in overall activity, such as during server updates or holidays, can make all assets appear underused. To handle this, the model evaluates each asset relative to other active assets in the same category. The total peer-group usage signal is then computed as the sum of the decay-weighted usage signals of all active peer assets, as defined in Eq. 10.

$$U_{\text{total}} = \sum_{j=1}^{N_{\text{peers}}} U_{\text{signal},j} \tag{10}$$

Next, the target asset's peer-relative usage ratio is defined as:

$$R_{usage} = (\frac{U_{signal,target}}{U_{total}}) * N_{peers} \tag{11}$$

In Eq. (11), $U_{\text{signal,target}}$ is the decay-weighted usage signal of the target asset, U_{total} is the total usage of the peer group, and N_{peers} is the number of active assets in the same category. This self-normalizing formulation gives $R_{\text{usage}} = 1$ when the target asset is at the peer average, while values above or below 1 indicate stronger or weaker demand. The ratio is then used as a pricing signal for real-time price updates after cold-start initialization.

GP predictive uncertainty and price update : For the current usage score $S_{i,t}$, the online GP produces a posterior mean $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$:

$$P_{i,t} \mid S_{i,t}, D_{i,t} \sim \mathcal{N}(\mu_{i,t}, \sigma_{i,t}^2). \tag{12}$$

Here, $\mu_{i,t}$ is the GP prediction for the current asset and $\sigma_{i,t}$ quantifies its predictive uncertainty. A larger $\sigma_{i,t}$ indicates limited or more variable evidence for the current usage condition. A Thompson-sampling candidate is drawn from this posterior and bounded below by a feasible price floor:

$$P_{i,t}^{\text{GP}} = \max(\tilde{P}_{i,t}, P_{\text{floor}}), \qquad \tilde{P}_{i,t} \sim \mathcal{N}(\mu_{i,t}, \sigma_{i,t}^2). \tag{13}$$

The sampled value $\tilde{P}_{i,t}$ enables exploration of plausible prices under uncertainty, while P_{floor} prevents an infeasible zero or negative rental price. Therefore, The peer-relative usage target is:

$$P_{i,t}^{\text{usage}} = \mu_c^{\text{prior}} \left(R_{i,t}^{\text{usage}}\right)^{\beta}, \qquad \beta = 0.3, \tag{14}$$

Here, μ_c^{prior} is the GP prior mean of the matched asset category and $R_{i,t}^{\text{usage}}$ is the asset's usage relative to active same-category peers. When $R_{i,t}^{\text{usage}} = 1$, the target equals the category prior; higher or lower values increase or decrease the target, respectively. The sub-linear elasticity $\beta = 0.3$ reduces sensitivity to short-term usage variation. The final target blends the GP candidate with the usage-based target:

$$P_{i,t}^{\text{target}} = \omega_{i,t} P_{i,t}^{\text{GP}} + (1 - \omega_{i,t}) P_{i,t}^{\text{usage}}, \qquad \omega_{i,t} = \frac{N_{\text{session}}}{N_{\text{external}} + N_{\text{session}}}. \tag{15}$$

The weight $\omega_{i,t}$ increases with the amount of in-platform session evidence. Thus, early decisions rely more on the external category prior, whereas observed in-platform activity receives greater weight as sessions accumulate. Finally, the update is limited by posterior uncertainty and take decison on the new price:

$$P_{i,t+1} = P_{i,t} \left[1 + \text{clip}\left(\frac{P_{i,t}^{\text{target}} - P_{i,t}}{P_{i,t}}, -\delta_{i,t}, \delta_{i,t} \right) \right], \qquad \delta_{i,t} = \min\left(0.10, \frac{\sigma_{i,t}}{\mu_{i,t}}\right). \tag{16}$$

Thus, uncertainty governs both exploration through Thompson sampling and the maximum permitted price movement, while the blend shifts from external prior information toward in-platform evidence as session observations grow.

3 Performance analysis

Target metrics preserve cross-category economic value : A key property of the target metric is that it rescales large sale prices to rental-level prices while preserving the relative economic differences across asset categories. Although the absolute price level becomes smaller after the transformation, the category ratios remain unchanged. As shown in Fig. 2, Category2 remains approximately 1.92 times the value of Category1, and Category3 remains approximately 16.67 times that of Category1 both before and after the transformation.

For example, the geometric-mean prices in the original space are approximately 0.03 ETH for Category1, 0.0575 ETH for Category2, and 0.50 ETH for Category3, while after transformation they become approximately 0.000012 ETH, 0.000025 ETH, and 0.000195 ETH, respectively. This indicates that the transformation rescales prices without altering the underlying value structure. Such a property is important in rental systems, where prices must be operationally practical while preserving the economic information learned from the

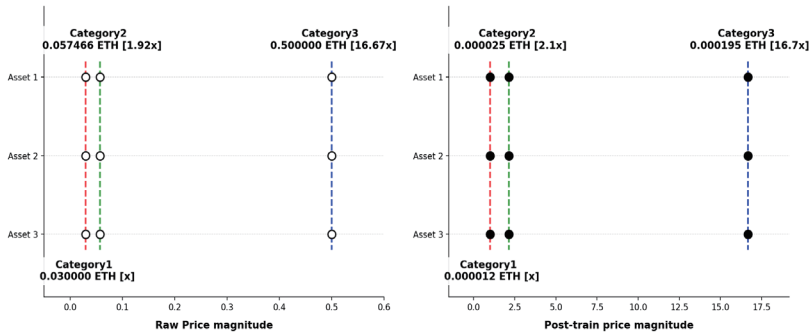


Fig. 2: Preserved ratio

original market data. By maintaining cross-category value ratios, the target metric preserves the category-level relationships required for reliable cold-start pricing. Thus, the transformed prices remain suitable for rental use while reflecting the same value hierarchy as the source marketplace. Thus, the transformation preserves the relative category-level value structure of the external sale-price data on the smaller proxy scale used for the simulated gaming-NFT rental scenario.

In-platform cold-start pricing via. GM : When a new asset enters the platform, its initial price is set by the geometric mean of the currently relevant same-type peer prices, as defined in Eq. 5. This provides a cold-start price that is robust to extreme observations and consistent with the internal market state at entry. The first three assets share the same initial geometric mean because they enter before an updated in-platform benchmark is formed under Eq. 6. In contrast, later assets are initialized using a newly computed staged geometric mean based on the most recent prices of active peer assets in the same category. Therefore, the initial price is stage-dependent and reflects the current internal pricing environment rather than a fixed benchmark.

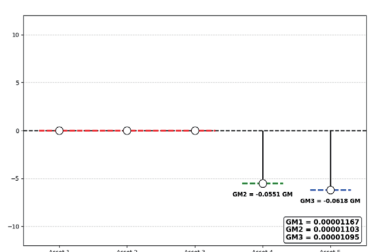
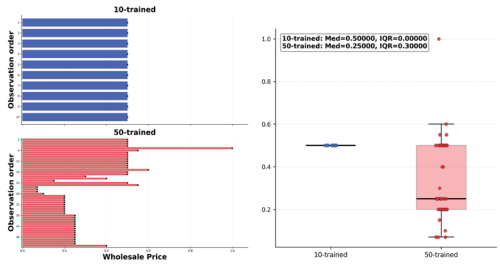


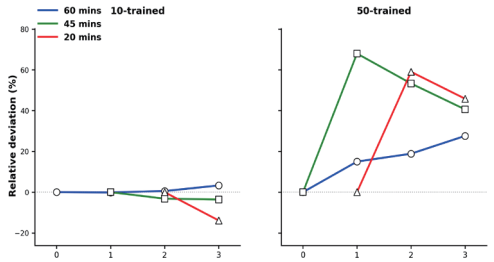
Fig. 3: Cold start pricing

Fig. 3 illustrates the staged initialization process. Assets 1–3 start from the same geometric-mean level because they belong to the earliest entry stage and therefore share the same initial benchmark. By contrast, Assets 4 and 5 begin at lower levels because the updated peer-based benchmark at their entry time has moved below the original one. This shows that the model recalibrates entry prices according to the latest in-platform conditions rather than assigning a universal initial price. As a result, each new asset is initialized according to the evolving internal market environment.

Training data influence on price update decision (concentrated vs. broad) : We compare two same-type prior settings: a concentrated 10-trained case and a broader 50-trained case. As shown in Fig. 4a, the 10-trained observations are concentrated around 0.50, whereas the 50-trained observations span approximately 0.07–1.00 and contain multiple clusters.



(a) Datapoints (b) Median and IQR



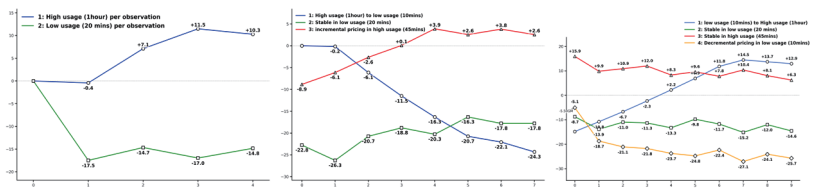
(c) Market behavior based on training data influence

Fig. 4: Training datapoints and platform behavior

Figure 4b (median/IQR) graphically displays the distinction: For the case of 10 trained it has an IQR 0 and median of 0.50 for value that whereas the 50-case it has median 0.25 and IQR of 0.30 so it gives a narrower price guideline range in starting. This difference directly influences price updating within the platform. As shown in Fig. 4c, the concentrated prior yields smoother price

updates that follow the usage ordering of peer assets. The broader prior produces larger early deviations, including temporary reversals in which a lower-usage asset exceeds a higher-usage asset. These results suggest that, in peer-based pricing, a concentrated prior provides a clearer initial benchmark and enables faster alignment with actual in-platform behavior.

Dynamic pricing behavior : To show how the model's decision mechanism influences pricing, several same-category assets were evaluated together and their percentage deviations were tracked relative to the initial benchmark price. In Figure 5a contrasts high usage with low usage. Asset 1 peaks around 11.5% above initial geometric mean after high usage, while Asset 2, with 20 minutes usage, is about 14–17% below benchmark.



(a) High vs.low (b) High, low and medium (c) 2nd-staged GM asset

Fig. 5: Dynamic pricing behavior based on the usages

Figure 5b demonstrates how prices readjust after Asset 3 joins. As Asset 1 shifts from using the asset for an hour to only 10 minutes, the price drops from a premium to about 21%–24% below the initial benchmark. With 45-minute usage, Asset 3 joins at a small premium and Asset 2 continues to trade at a discount. Fig. 5c shows the staged-entry benchmark effect. Assets 1-3 start from the first GM, while Asset 4 enters later, 5.5% below the benchmark. With low usage (10 minutes), Asset 4 quickly becomes the lowest-valued asset, falling to 18.7% below the benchmark and then to about 22%-25% below it. Asset 1 rises from a 24.3% discount to a 12.9% premium as usage increases to 1 hour. Asset 2 stays at a 14%-15% discount and Asset 3 maintains a 6%-10% premium. The results show that pricing depends on the stage and peer comparison, influencing asset prices within the same category.

4 Closing remarks

This study proposes a dynamic pricing model for NFT rental platforms that creates a win-win situation for both owners and users through initial pricing prediction and usage-based price updates. A minimal training set can work well when prices are sufficiently consistent. The rental conversion metric is flexible, allowing platforms to choose daily, hourly, or minute-based pricing by surveying

thier own market research. As usage and peer-activity data grow, the model relies more on internal platform signals, helping prices better reflect relative asset value within each category. In this way, the framework reduces the burden for artists/users of manual market research while gradually improving pricing through observed behavior.

Acknowledgments. This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024(Project Name: Global Talent Training Program for Copyright Management Technology in Game Contents, Project Number: RS-2024-00396709, Contribution Rate: 100%)

References

[Ahmed et al.2024] Ahmed, M.S., El-Masry, A.A., Al-Maghyereh, A.I., Kumar, S.: Cryptocurrency volatility: A review, synthesis, and research agenda. Research in International Business and Finance **71**, 102472 (2024)

[Ye et al.2023] Ye, Z., Zhang, D.J., Zhang, H., Zhang, R., Chen, X., Xu, Z.: Cold start to improve market thickness on online advertising platforms: Data-driven algorithms and field experiments. Management Science **69**(7), 3838–3860 (2023)

[Zeng and Xiao2024] Zeng, W., Xiao, Z.-y.: Few-shot learning based on deep learning: A survey. Mathematical Biosciences and Engineering **21**(1), 679–711 (2024)

[Wang and Jing2022] Wang, W., Jing, B.-Y.: Gaussian process regression: Optimality, robustness, and relationship with kernel ridge regression. Journal of Machine Learning Research **23**(193), 1–67 (2022)

[Faridi2022] Faridi, O.: Blockchain Gaming Survey Reveals Gamers Are "Positive" About Benefits Of Play-and-Earn Games. https://www.crowdfundinsider.com/2022/12/199830-blockchain-gaming-survey-reveals-gamers-are-positive-about-benefits-of-play-and-earn-games/ (2022)

[Efendioğlu2022] Efendioğlu, A.: New survey reveals most gamers look positively upon blockchain gaming. https://mobidictum.com/survey-reveals-most-gamers-positive-on-blockchain-gaming/ (2022)

[Gómez-Gonzalvo et al.2020] Gómez-Gonzalvo, F., Devís, J., Molina, P.: Video Game Usage Time in Adolescents' Academic Performance. https://doi.org/10.3916/C65-2020-08

[Azmi2022] Azmi, E.: CoinGecko Axie Infinity Survey 2021. https://www.coingecko.com/research/publications/coingecko-axie-infinity-survey-2021 (2022)

[scikit learn] scikit-learn: scikit-learn: matern

[European Commission2026] European Commission: Step 6. Weighting. European Commission Knowledge for Policy (2026)

[Clark-Carter2010] Clark-Carter, D.: Measures of central tendency. In: International Encyclopedia of Education, Third edition edn., pp. 264–266. Elsevier, Oxford, UK (2010). https://doi.org/10.1016/B978-0-08-044894-7.01343-9

[Luxenberg and Boyd2024] Luxenberg, E., Boyd, S.: Exponentially weighted moving models. (2024)

Uncovering Illegal IPTV Service Infrastructure Through Static Analysis of Android Applications

Jeong Yun Ryu[1][0009-0000-5079-031X], Eunjin Kim[1][0000-0002-2840-4918] and Gibum Kim[1] [0000-0001-7298-4254]

[1] Department of Forensic Sciences, Sungkyunkwan University, Seoul, Republic of Korea
{stellano1, eunjin810}@g.skku.edu, freekgb02@gmail.com

Abstract. As illegal IPTV services have evolved into dedicated Android-based applications, identifying service operators through application analysis has become increasingly important. However, illegal streaming applications often employ access control mechanisms such as region-based restrictions and device authentication, making execution impossible without an authorized device or subscription and limiting the applicability of dynamic analysis. In this study, static analysis was performed on two illegal IPTV applications (ip.apk and my.apk) using JADX and Bytecode Viewer. Both applications were found to use the 'tvbus.android' and ExoPlayer libraries for video playback and contain hard-coded operator-tracing indicators, including authentication server addresses, API endpoints, and external service integration URLs. Comparative analysis further suggests that my.apk is a later version derived from ip.apk with extended functionality. This study demonstrates that static analysis alone can reveal inter-application relationships and infer aspects of the service operation structure even when execution is not possible, providing a foundation for more comprehensive investigations when combined with dynamic analysis.

Keywords: Illegal IPTV, Static Analysis of APK, Copyright Infringement.

1 Introduction

With the rapid growth of Internet-based broadcasting services, illegal services that retransmit broadcast content without copyright holders' authorization have become a serious global issue. In November 2024, Europol dismantled a large-scale illegal IPTV service through "Operation Taken Down," targeting a network serving approximately 22 million users worldwide with monthly revenues of €250 million [1]. In February 2026, a follow-up operation involving nine countries further dismantled illegal IPTV infrastructure [2]. Copyright infringement is also escalating into an international issue in Southeast Asia, with operators of illegal IPTV services targeting Malaysia and Singapore indicted through Interpol's I-SOP program [3].

While illegal IPTV services initially operated via websites or Kodi-based add-ons, they are evolving into harder-to-trace Android applications, as earlier formats were easily blocked through domain and server IP tracking. Illegal IPTV services targeting Korean expatriates are proliferating across Southeast Asia, increasingly delivered via

dedicated apps. However, these apps employ protective measures such as source code obfuscation, encrypted communication, and geoblocking, often making execution impossible without the actual device or a valid subscription. Consequently, traditional dynamic-analysis-centered methodologies are difficult to apply, creating a need for techniques that can track operators without requiring app execution.

This paper analyzes the code structure and service infrastructure of two illegal Android IPTV applications through static APK analysis to identify operator-tracing indicators. It demonstrates that inter-application relationships and aspects of the service's operational structure can be identified through static analysis alone—even when the apps cannot be executed—establishing a foundation for deeper tracking when combined with dynamic analysis.

The paper is organized as follows: Chapter 2 reviews prior research, Chapter 3 presents the analysis subjects and methodology, Chapter 4 details the analysis results, and Chapter 5 offers conclusions and future research directions.

2 Background and Related Works

2.1 Operational Structure of Illegal IPTV Services

Illegal IPTV refers to copyright infringement involving the unauthorized retransmission, reproduction, or distribution of broadcast content by exploiting Internet Protocol-based broadcasting technology. While legitimate IPTV services are restricted to domestic use, requiring dedicated lines provided by an ISP, illegal IPTV bypasses these limitations, enabling real-time reception of domestic broadcasts from overseas [4].

The typical operational structure involves extracting broadcast signals via legal set-top boxes and encoders installed domestically, transmitting them to overseas servers, and distributing the content to users. This setup—content extraction equipment located domestically, transmission infrastructure based overseas—allows stable service delivery even in regions with unstable internet connectivity, since servers operate locally within the target areas [5]. Because the primary audience consists of nationals residing abroad and the core criminal activity occurs overseas, rapid law enforcement response is difficult; even after detecting illegal transmission, authorities face complex hurdles such as securing cooperation from local ISPs to identify server operators, resolving jurisdictional issues, and following international legal assistance procedures.

2.2 Related Work

Research on illegal IPTV initially focused on collecting Kodi-based forensic artifacts and tracing distributors; it subsequently evolved to include APK reverse engineering, library analysis, and dynamic protocol analysis.

Sheppard (2018) [6] investigated the Kodi illegal streaming distribution network by analyzing addon.xml files and visualizing developer-repository relationships using the GitHub GraphQL API, proposing an investigative methodology for tracing cloud-based distributors. Bromley et al. (2020) [7] identified digital evidence from SQLite databases, log files, and add-on directories across multiple operating systems and proposed

a forensic investigation framework for Kodi-based illegal IPTV services. Warrior et al. (2020) [8] developed de-Kodi, collecting 9,146 unique add-ons and revealing through mitmproxy-based traffic analysis that illegal content add-ons deliberately utilize obscure domains to evade detection. Huang et al. (2022) [9] decompiled 21 illegal streaming applications, identifying commonly used libraries including TVBus and Gemini, and revealed that Binstream.io was a key technology provider supplying P2P libraries while also directly operating illegal streaming services. Choi et al. (2023) [10] employed Frida-based dynamic analysis to decrypt P2P network traffic, tracing broadcast source servers and peer IP addresses across 38 Korean television channels. Kim et al. (2025) [4] proposed an investigation system integrating Wireshark, Frida, and the Criminal IP API to automate the process from packet collection to international law enforcement cooperation. Ahn et al. (2025) [11] reverse-engineered libtvcore.so, identifying 131,175 users across 116 countries and 78 operational servers, and proposed countermeasures including blocking hardcoded DNS domains and conducting Sybil attacks. As discussed above, existing studies primarily rely on dynamic analysis in executable environments. However, many illegal IPTV applications employ access control mechanisms that prevent execution. Therefore, this study proposes a methodology for identifying structural characteristics and operator-tracing indicators solely through static analysis without requiring application execution.

2.3 Challenges

Illegal IPTV investigations typically involve acquiring an illegal streaming device or application, identifying streaming servers through network analysis, and apprehending operators via international cooperation. However, illegal streaming applications employ various protection techniques that make analyzing their internal structures difficult, and authentication and content distribution servers are often distributed across different countries.

These limitations can be summarized as follows. First, code obfuscation and packing make it difficult to understand the internal structures and operational mechanisms of illegal streaming applications through static analysis alone. Second, access control mechanisms restricting service access to specific regional IPs or pre-registered devices often prevent application execution, making conventional dynamic analysis difficult to apply. Third, limited research has systematically identified structural characteristics common across Android-based illegal streaming applications for operator tracing. Therefore, this study proposes a methodology for identifying structural characteristics and operator-tracing artifacts through APK static analysis, without requiring application execution.

3 Analysis Methodology

3.1 Analysis Objectives and Environment

Illegal IPTV services often store critical operational information, such as authentication server addresses, API endpoints, and encryption keys, as hardcoded values within the

application to support core functions including device authentication, channel list retrieval, and content streaming. Such information can be extracted through static analysis without executing the application, providing valuable insights into service operational mechanisms and operator identification.

In this study, static analysis was performed on ip.apk and my.apk to examine their execution flows, core functionalities, and hardcoded URLs, while also comparatively analyzing their code structures to identify the relationship between the two applications. The applications were obtained from their official distribution website prior to its shutdown. Detailed information on the analysis tools are presented in Table 1.

Table 1. Analysis Environment

Div	Name	Version	Usage
Analysis Tools	Jadx[12]	1.5.5	APK Disassembling
	Java Eclipse	4.32.0	Text Decoding and Parsing
	Bytecode Viewer[13]	2.13.2	Text Extraction from APK Files
	Nox Player	Nox version: 7.0.6.2 Device model: SM-S9180 Android version: 9	Install APK files
	Criminal IP	-	Check Domian Info
	Who.is	-	Check Domian Info
	VMware Workstation Pro	VMware version: 17.5.2 Ubuntu version: 26.04	so File Analysis
Analysis Targets	ip.apk	5.0.6	Illegal IPTV APK
	my.apk	2.01	Illegal IPTV APK

3.2 Analysis Procedure and Method

The static analysis was conducted in three main stages: application execution, code analysis, and string extraction.

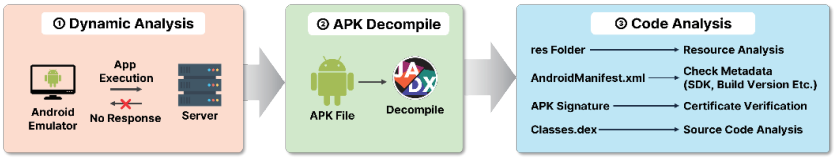


Fig. 1. APK File Analysis Procedure

Application Execution: Prior to static analysis, ip.apk and my.apk were installed on the Nox Player emulator and execution was attempted. However, both applications failed to operate properly due to server errors during startup, presumably because authentication and content servers were shut down following the closure of the service's distribution website. Consequently, this study was limited to static analysis.

Code Analysis: The APK files were decompiled using JADX to reconstruct the Java source code, from which the overall execution flow, class and function invocation

relationships, and functional organization were analyzed. The AndroidManifest.xml file was examined to identify requested permissions and component configurations.

String Extraction: Bytecode Viewer was used alongside JADX to extract hardcoded strings from the APKs, facilitating identification of operator-tracing indicators such as URLs, domain names, and APIs. Extracted strings were decoded and organized using Java Eclipse, after which WHOIS lookups were performed to obtain domain registration and registrar details.

4 Analysis Results

4.1 ip.apk

Metadata & Authenticaion. The package is named com.myip4k.tv, with versionCode 506 (versionName 5.0.6). It uses compileSdkVersion 34 (Android 14), minSdkVersion 16 (Android 4.1), and targetSdkVersion 28 (Android 9)—below current Google Play target API requirements. Among its 14 permissions, potentially misused ones such as SYSTEM_ALERT_WINDOW, deprecated permissions, and READ/WRITE_EXTERNAL_STORAGE were identified. The usesCleartextTraffic="true" setting permits unencrypted HTTP traffic, while the org.sopcast.android class path suggests repackaging of the open-source SopCast streaming framework.

The signing certificate is self-signed, with identical issuer and subject fields containing O=IPTV4K, OU=dev team, and CN=dev. Its country code (C=60) does not conform to the ISO 3166-1 alpha-2 standard. Valid for roughly 25 years (2020–2045) and using a 2048-bit RSA key with SHA256, the certificate indicates issuance in a small-scale development environment without a trusted certificate authority.

Execution Flow. When the ip.apk is launched, the ‘onCreate()’ method of ‘SopCast’—which serves as the ‘mainActivity’—is launched. During the ‘onCreate()’ phase, ‘com.tvbus.engine.TVCore’, a P2P streaming engine used for proprietary streaming, and ‘com.google.android.exoplayer2’, which is responsible for standard media playback, are initialized. The application is then presumed to communicate with the authentication server and subsequently retrieve the channel list. During the authentication process, the ‘AuthInfo’ data model manages authentication-related information, including ‘Keys’, ‘Service’, and ‘User’, while the channel list is received. Finally, the user interface is presumed to be displayed during the ‘onResume()’ phase.

Server address resolution employs DNS-over-HTTPS, making it difficult to trace servers through conventional DNS lookups. Automatic execution at system boot is handled by ‘org.sopcast.android.broadcast.ActionReceiver’, which utilizes the ‘RECEIVE_BOOT_COMPLETED’ permission to automatically start the service after the device restarts. The overall application execution flow is illustrated in Fig. 2.

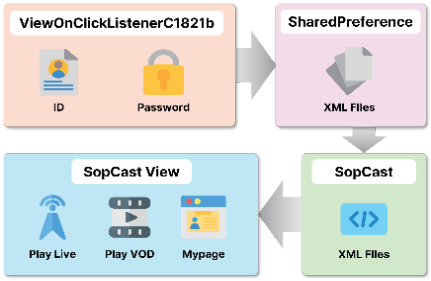


Fig. 2. ip.apk execution flow

The major data models used by the application are summarized in Table 2.

Table 2. SopCast Major Data Model

Class	Role
AuthInfo(Keys, Service, User)	Authentication information
LoginInfo, RegisterInfo	Login and Registration Information
ChannelBean	Channel Name, Logo, Source URL, Tags, and Subtitle Information
CustomUserInfo, ResellerBean	Account Information
EpgBeans, ProgramInfo	EPG(Electronic Program Guide) Information
GroupL1, Group	Channel Grouping Information
HistoryBean	Viewing History
UpdateInfo	Application Update and Server Configuration Information
DoH, DoHMsg	DNS-over-HTTPS Configuration and Message Information

'SopCast' is implemented using a single-screen view switching architecture rather than an activity transition-based structure. Specifically, it employs a combination of 'MyViewPager' and 'AutoLayoutRadioGroup', where selecting a 'RadioButton' triggers page transitions within the 'ViewPager'. 'View switching' and execution support both remote-control navigation keys and touch interactions, indicating that the application was designed for use in both TV and devices equipped with touchscreens.

In the video playback view, both 'android.widget.VideoView' and 'com.google.android.exoplayer2.ui.PlayerView' (ExoPlayer) are declared as dual media players and are used interchangeably depending on the playback context. The actual streaming is handled by 'com.tvbus.engine.TVCore' (libtvcore.so), which operates by converting channel URLs encoded with the custom 'tvbus://' protocol into actual stream addresses through a P2P broker and then passing them to the media player for playback. However, since 'libtvcore.so' is protected with UPX packing, detailed internal behavior cannot be fully examined through static analysis.

Major URL. The ip.apk applies string obfuscation, where a 'deobfuscation' function 'D.a' is invoked at the point of string usage to decode and utilize the obfuscated values. The execution flow of 'D.a' is illustrated in Fig. 3. In this study, we reverse-engineered this process and developed a Java-based tool that batch-decrypts obfuscated strings

extracted using Bytecode Viewer. The decrypted strings were subsequently parsed, organized, and structured into an Excel file for further analysis.

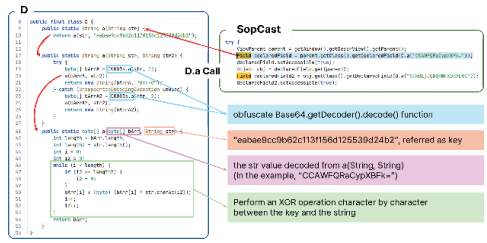


Fig. 3. String Obfuscation Removal Process

Obfuscation has also been applied to function names. However, to 'deobfuscate' function names, runtime information must be obtained by performing a memory dump in a dynamic analysis environment. Since this study was conducted primarily based on static analysis, 'deobfuscation' of function names was excluded from the scope of analysis. The main URLs identified from the decryption results are shown in Table 3.

Table 3. List of URLs in ip.apk

#	URL	Registrant Organization	Register	AS Name
1	http://2.4989.***/Launcher-worldiptv.apk	-	-	-
2	https://auth1.world***/api/v2/register	Shield Whois	Name SRS AB	-
3	https://auth1.world***/api/v2/auth			
4	https://aomedia.org/emsg/ID3	Chrome Open Media Team @ Google	Squarespace Domains II LLC	FASTLY
5	https://developer.apple.com/streaming/emsg-id3	Apple Inc.	Nom-iq Ltd. dba COM LAUDE	Apple Inc.
6	https://auth1.myip***/v2/register	-	GoDaddy.com, LLC	Trunk Networks LTD
7	https://auth1.myip***/v2/auth	-		
8	psp.myip***.***	-		

There are three several URLs correspond to the Android XML namespace, and the Android UI layout namespace. Two URLS correspond to Tencent's mobile crash reporting and analytics service, through which application crash data is transmitted to the respective servers.

http://2.4989.***/Launcher-worldiptv.apk provides an APK file for the WorldIPTV service, a known illegal IPTV application. This link corresponds to the "WorldIPTV installation navigation" function identified in the UI structure analysis (Lc/c/a/b/m view). The service has previously been associated with enforcement actions, and part of its operating group was reportedly arrested in 2019.

https://developer.apple.com/streaming/emsg-id3 is a reference to Apple's time-based metadata specification, serving as a standard identifier used in ExoPlayer's EMSG (Event Message) processing module.

https://auth1.myip***/v2/register and https://auth1.myip***/v2/auth are presumed to be used for device registration and user authentication, respectively. However, during the static analysis process, no logic explicitly invoking these URLs was identified; therefore, their actual usage and implementation could not be verified.

Login and Authentication. The login process is performed in a separate view, where the client first validates whether the input values for the user ID and password satisfy predefined conditions such as string length. If the conditions are met, the credentials are stored in a local XML file on the device using 'SharedPreferences' and then passed to the 'SopCast' authentication function. During this process, no evidence of hashing or encryption applied to the password was identified, suggesting that the credentials are stored in plaintext on the local device. This implies that if access to the local file system is obtained during forensic analysis, user credentials can be retrieved in plaintext form.

The actual authentication request is not performed directly within the application's Java code but is handled internally by the 'TVCore' component of the 'tvbus.android' library. The 'tvbus.android' library is an Android live streaming SDK based on P2P communication, released as open source by Binstream.io, a company located in Shanghai, China. Authentication is performed by passing the authentication server address (auth_url_sdk) to 'TVCor'e via the 'TVCore.setAuthUrl' method. However, during the static analysis process, the actual authentication server URL passed to 'TVCore.setAuthUrl' could not be identified.

Native Library. The shared library files 'libBugly.so', 'libgojni.so', and 'libtvcore.so' were identified in ip.apk. 'libBugly.so' is a native library associated with Tencent's crash analysis and monitoring service. 'libgojni.so' is a shared library enabling Go-language code to run within the Android environment. 'libtvcore.so' was identified as a native library providing P2P-based live streaming functionality. A resource class associated with 'libtvcore' was found for libgojni.so, but it was empty, and no other code invoking or utilizing 'libgojni' could be identified.

The symbolic links for 'libffmpegJNI.so' and 'libgojni.so' were removed, while the symbolic link for 'libtvcore.so' remained intact. However, since the application could not be executed normally during analysis, library calls and actual execution behavior could not be verified through dynamic techniques such as hooking. A study by Ahn et al. [11] found that the 'libtvcore.so' in a similar illegal IPTV application appeared to handle URL encryption and decryption. Based on this finding, 'libtvcore.so' in ip.apk is presumed to perform a similar function.

4.2 my.apk

Metadata & Authentication. The package name is com.iptv.myiptv, with a versionCode of 201 (versionName 2.01). Both the compileSdkVersion and

targetSdkVersion are set to 34 (Android 14), while the minSdkVersion is set to 23 (Android 6), indicating that the application was built according to a more recent Android specification than ip.apk. Multiple advertising- and tracking-related permissions, including AD_ID and ACCESS_ADSERVICES_ATTRIBUTION, are included, suggesting that an advertising-based revenue model is incorporated into the application.

The application is configured with android:debuggable="true", and its self-signed certificate (CN=Android Debug, O=Android, C=US) matches the standard Android debug key template. The use of a 1024-bit RSA key and SHA1 signature indicates that the APK was distributed as a debug build without a formal release signing certificate.

Execution Flow. The application initiates execution by launching 'StartupActivity', which is subsequently transitioned to 'LoginActivity' to perform user authentication. Within 'LoginActivity', the methods 'checkMainRoute', 'checkVersion', 'checkMeNow', and 'checkLogin' are implemented. These methods sequentially perform server route validation, version verification, and session validation prior to authentication. Upon successful login, 'HomeActivity' is launched, and each functional module is provided through its associated activities. The overall execution flow is illustrated in Fig. 4. The composition of activities connected to 'HomeActivity' is presented in Table 4.

Table 4. Types of Activities in 'HomeActivity'

Name	Role
LiveGridActivity	Provides live streaming play function
MovieGridActivity	Provides a movie content menu and play function
SeriesGridActivity	Provides a series content menu and play function
DramaGridActivity	Provides a drama content menu and play function
SportGridActivity	Provides a sports content menu
FavoriteGridActivity	Provides a list of favorited (liked) videos
HitGridActivity	Provides a list of popular videos
VipGridActivity	Provides a list of VIP-only content
SearchActivity	Provides a content search functionality
UserSettingActivity	Provides content search function
SettingActivity	Provides user customization features
TopupActivity	Provides in-app recharge and payment function
AdvertiseActivity	Provides ad display function
LoginActivity	Perform user login

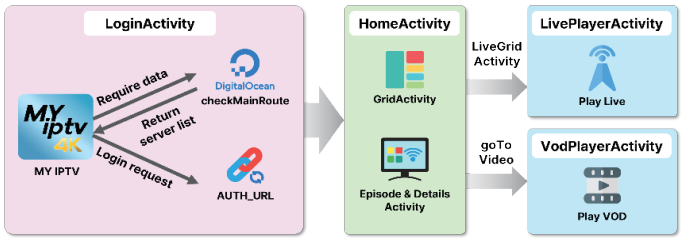


Fig. 4. my.apk execution flow

'SportGridActivity' provides country and year-based filtering for sports content. 'VipGridActivity' and 'VipPasswordActivity' include password authentication for VIP-exclusive content. 'TopupActivity' supports multiple recharge methods, including PIN code, wallet reference, and monetary value. 'AdvertiseActivity' indicates the presence of an ad-supported revenue model. The activities support remote control, keyboard, and touch interactions, suggesting compatibility with smart TVs, PCs.

Live streaming is handled by 'LivePlayerActivity', which performs P2P streaming via the 'tvbus://' protocol through 'TvbusService', switching to ExoPlayer when a standard protocol URL is provided. Session management, forced logout, and channel list updates are also processed within the playback screen.

VOD playback is divided into two activities: 'VodPlayerActivity' using ExoPlayer and 'VodPlayerTvCarActivity' using the libtvcar engine via libgojni.so. The appropriate activity is invoked from SeriesEpisodeActivity, MovieDetailsActivity, and DramaEpisodeActivity depending on the stream type.

Major URL. my.apk manages URL information used within the application by defining it in the 'ApiUtils' utility class. Each component within the application references these URLs through variables declared in this class. However, it was also observed that some URLs are dynamically generated based on the 'BASE_URL' value, as illustrated in Fig. 5.

```
static {
    StringBuilder sb = new StringBuilder();
    sb.append(BASE_URL);
    sb.append("/me");
    CHECK_TOKEN = updateBaseUrl(sb.toString());
    AUTH_URL = updateBaseUrl(BASE_URL + "/auth");
    AUTH_LOGOUT_URL = updateBaseUrl(BASE_URL + "/auth/logout");
    CHECK_VERSION = updateBaseUrl(BASE_URL + "/apk");
    MOVIE_URL = updateBaseUrl(BASE_URL + "/movies");
    SERIES_URL = updateBaseUrl(BASE_URL + "/series");
    DRAMA_URL = updateBaseUrl(BASE_URL + "/drama");
```

Fig. 5. Example of Dynamically Declared URLs in ApiUtils

The 'BASE_URL' is stored in the 'android.support.AppLaunchChecker.xml' file and managed via 'SharedPreferences'. However, since this file is generated at runtime, it could not be identified during the static analysis process. The URLs identified in my.apk are summarized in Table 5.

Table 5. List of URLs in my.apk

#	Variable name	URL	Registrant Organization	Register	AS Name
1	AUTH_TVBUS	https://auth.myip***/v2/auth	-	GoDaddy.com, LLC	Trunk Networks LTD
2	AUTH_TVCAR	https://thai_auth.myip***/v1/auth			
3	CHECK_IP_ADDRESS_FROM_ISP	http://checkip.amazonaws.com/	Amazon.com, Inc.	MarkMonitor Inc.	AMAZON-02

4	CHECK_ROUTE_URL	https://plankton-***.ondigital-ocean.app/host	-	1API GmbH	CLOUDFLARENET
5	CHINA_VOD_AUTH	http://tvbus.myip***:3000/api/v1/vod-auth	Domain Protection Services, Inc.	Name.com, Inc.	-
6	IMDB_BACKDROP_FIRST_URL	https://api.the-moviedb.org/***/	-	MarkMonitor Inc.	Amazon.com, Inc.
7	IMDB_BACKDROP_PATH_URL	https://image.tmdb.org/***	c/o whoisproxy.com	Amazon Registrar, Inc.	Datacamp Limited

Most URLs integrate external services and functionalities. ‘CHECK_IP_ADDRESS_FROM_ISP’ is an Amazon-provided public IP lookup service used to retrieve and display the user's IP address [14]. ‘CHECK_ROUTE_URL’ is a DigitalOcean-assigned URL implementing a traffic distribution routine that randomly selects among returned responses, allowing flexible adaptation to server changes or blocking attempts without hardcoding server addresses.

‘IMDB_BACKDROP_PATH_URL’ and ‘IMDB_BACKDROP_FIRST_URL’ are presumed to be based on the TMDB API, used to verify a video item's existence and retrieve its background image.

‘AUTH_TVBUS’ serves as the authentication server address in ‘TvBusService’, sending POST requests via EventBus when a URL contains the ‘tvbus://’ scheme. ‘TVCarService’ uses ‘AUTH_TVCAR’ as its authentication server, with ‘VodPlayerTvCarActivity’ invoking ‘Libtvcar.start’ rather than ‘startPlayback’, indicating a distinct playback mechanism.

Authentication is implemented through the ‘signIn’ method of ‘LoginActivity’, which sends an HTTP POST request to ‘AUTH_URL’, dynamically constructed from ‘BASE_URL’ in the ‘ApiUtils’ class. As a result, the final authentication URL could not be determined through static analysis. Unlike ip.apk, which stores credentials in plaintext within ‘SharedPreferences’, my.apk performs server-side authentication, and local credential storage could not be confirmed through static analysis alone.

Native Library. Analysis of the ‘lib’ directory identified three shared library files: ‘libffmpegJNI.so’, ‘libgojni.so’, and ‘libtvcore.so’. ‘libffmpegJNI.so’ is a shared library file of FFmpeg, an open-source multimedia framework that supports the encoding and decoding of video, audio, and images. It may be included when ExoPlayer uses the FFmpeg extension. ‘libgojni.so’ and ‘libtvcore.so’ were also identified in the previously analyzed ip.apk. Considering their names and structural characteristics, these libraries are considered likely to perform functions similar to those identified in ip.apk. Unlike ip.apk, the symbolic links for all three files had been removed. However, it was difficult to determine whether the removal of the symbolic links was the result of intentional obfuscation.

4.3 Discussion and Limitation

This study conducted a static analysis of two illegal IPTV applications, ip.apk and my.apk, examining their code structure, functional composition, and network

communication. Both share common traits: providing VOD and live streaming, using ExoPlayer for playback, employing the same P2P streaming engine from Binstream.io (‘com.tvbus.engine.TVCore’, ‘libtvcore.so’), and using the same authentication server domain, “myip***.com”. Despite these similarities, the two applications differ substantially. ip.apk supports API levels 17–29 and manages functionality via view switching within a single interface, whereas my.apk targets API levels 19–34 and uses a conventional Android architecture with separate activities. my.apk also includes additional components (‘SportGridActivity’, ‘VipGridActivity’, ‘TopupActivity’, ‘AdvertiseActivity’), reflecting broader functionality and monetization. Regarding security, ip.apk stores credentials in plaintext within ‘SharedPreferences’, while my.apk performs server-side authentication; ip.apk's ‘libtvcore.so’ is also UPX-compressed and heavily obfuscated, indicating deliberate anti-analysis measures. These differences suggest my.apk is a more mature version of ip.apk, though a definitive developmental relationship cannot be confirmed from static analysis alone.

This study also faces limitations inherent to static analysis. Obfuscated strings and function names in ip.apk limited code comprehension and could not be resolved without runtime memory dumping. my.apk's ‘BASE_URL’ is retrieved dynamically at runtime, preventing determination of final server URLs—an apparently intentional evasion mechanism. Static analysis also cannot verify runtime behavior, including the purpose of unresolved URLs like ‘CHINA_VOD_AUTH’ and the P2P mechanism in 'libtvcore.so', which require further dynamic analysis.

5 Conclusion

This study proposes a methodology for identifying operator attribution indicators through static analysis of two Android-based illegal IPTV applications (ip.apk and my.apk), without executing them. Both applications employ 'tvbus.android' ('libtvcore.so'), a P2P streaming library from Binstream.io, and share the authentication server domain "myip***.com," indicating common service infrastructure. A hardcoded APK download link for WorldIPTV and its set-top box authentication server address were also found within ip.apk, establishing a direct link between the two services. Comparative analysis further suggests my.apk is an evolved successor to ip.apk, adding features such as sports content, VIP-exclusive content, subscription payment, and advertising-based monetization. These results demonstrate that static analysis can effectively identify inter-application relationships and underlying service infrastructure of illegal streaming applications.

Limitations include obfuscated string literals and function names, dynamically resolved server addresses such as 'BASE_URL' that could not be identified, and UPX compression of 'libtvcore.so' restricting P2P communication analysis. Future work should incorporate dynamic analysis using VPN bypass and emulator-based authentication, and develop automated detection methods based on common indicators such as the 'tvbus.android' library and "myip***.com" domain. This study provides a foundation for developing investigative methodologies to combat illegal IPTV services.

Acknowledgments. This study was conducted with the support of the Korea Creative Content Agency and the government (Ministry of Culture, Sports and Tourism) in 2026 (No. RS-2024-004395553).

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

References

1. Europol, "European law enforcement stops illegal IPTV service providers", 2024/11/28. https://www.europol.europa.eu/media-press/newsroom/news/european-law-enforcement-stops-illegal-iptv-service-providers, last accessed 2026/07/01
2. Eurojust, "Successful operation against illegal streaming services with millions of users worldwide", 2026/02/03. https://www.eurojust.europa.eu/news/successful-operation-against-illegal-streaming-services-millions-users-worldwide, last accessed 2026/07/01
3. The Online Citizen, "Thai man and Malaysian couple charged over MyIPTV4K streaming piracy", 2026.05.28. https://theonlinecitizen.com/2026/05/28/thai-man-and-malaysian-couple-charged-over-my-iptv-4-k-streaming-piracy, last accessed 2026/07/01.
4. Kim EunJin, Park JaeHyun, Park YeonJi, Park HyunMin, Kim GiBum, Design of an Illegal IPTV Tracking and Investigation System Based on ISD Analysis, The Transactions of the Korea Information Processing Society, 14(7), pp. 497-505 (2025)
5. Moon ByoungHun, Kim EunJin, Lee SangEun, Kim GiBum, Crime Response through Crime Script Analysis of Illegal IPTV, Journal of Digital Forensics, 19(3), pp. 20-40 (2025)
6. J. Sheppard, Cloud Investigations of Illegal IPTV Networks, 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 2018, pp. 1942-1947 (2018)
7. Samuel Todd Bromley, John Sheppard, Mark Scanlon, Nhien-An Le-Khac, "Retracing the Flow of the Stream: Investigating Kodi Streaming Services", Multimedia; Cryptography and Security. (2020)
8. Marc Anthony Warrior, Yunming Xiao, Matteo Varvello, and Aleksandar Kuzmanovic, De-Kodi: Understanding the Kodi Ecosystem. In Proceedings of The Web, Conference 2020. Association for Computing Machinery, New York, NY, USA, pp. 1171–1181. (2020)
9. Kong Huang, Ke Zhang, Jiongyi Chen, Menghan Sun, Wei Sun, Di Tang, Kehuan Zhang, Understanding the Brains and Brawn of Illicit Streaming App, International Conference on Digital Forensics and Cyber Crime(ICDF2C 2021), pp. 194-214, 441 (2022)
10. Choi Jungwan, Choi Geunyeong, Lee Sangjin, Tracing Copyright Infringement Activities through Illegal Streaming Device Protocol Analysis, Journal of Digital Forensics, 17(2) (2023)
11. Jungun Ahn, Sueun Jung, Seungwan Yoo, Jungheum Park, and Sangjin Lee, Watch Out Your TV Box: Reversing and Blocking a P2P-based Illegal Streaming Ecosystem, In Proceedings of the 34th USENIX Security Symposium. August 13–15, Seattle, WA, USA (2025)
12. Github, "skylot / jadx", https://github.com/skylot/jadx, last accessed 2026/06/11
13. Github, "Konloch / bytecode-viewer", https://github.com/konloch/bytecode-viewer, last accessed 2026/06/11
14. AWS, https://docs.aws.amazon.com/ko_kr/batch/latest/userguide/create-a-base-security-group.html, last accessed 2026/06/24

The 27th World Conference on Information Security Applications

WISA 2026

Posters

Poster Session

Poster Session 1 09:30–10:50, August 28 (Friday), 2026

Chairs: Jun Lee (KISTI), Jaedeok Lim (ETRI)

Poster 1-1 Confused Deputies in the Cloud: Indirect Prompt Injection Attacks on LLM Multi-Agent Systems
Seo-Yi Kim, Seungha Jee, Seongmin Kim and Il-Gu Lee (Sungshin Women's University)

Poster 1-2 Large Language Model Agents for Security on a Budget: Reasoning over Guessing
Anak Agung Ngurah Dharmawangsa, Thi-Thu-Huong Le, Jaehan Cho, Ahmada Yusril Kadiptya, Hyunjin Jang, Hoekwang Jeong and Howon Kim (Pusan National University)

Poster 1-3 Prompt-Based Line-Level Vulnerability Detection for Smart Contracts with LLMs
Ikkaku Yasuda and Mamoru Mimura (National Defense Academy)

Poster 1-4 White-Box Backdoor Training for Vision-Language Models Using Optimized FGSM Patches as Harmful Visual Triggers
Andro Aprila Adiputra, Mulyasari Wulan Silvani, Thi-Thu-Huong Le, Shin Dawit and Howon Kim (Pusan National University)

Poster 1-5 A Security Analysis of Intel TDX Live Migration
Jeongsu Kim, Taehun Kim and Youngjoo Shin (Korea University)

Poster 1-6 Correlation Power Analysis Attack on SPHINCS+
Jinwook Jang, Minjeong Lee and Dong-Guk Han (Kookmin University)

Poster 1-7 A Device-Internal Cross Domain Solution for Android Enterprise: Content-Level Mediation of Work-Personal Profile Transfers
Kim Subin and Park Junsoo (Kangnam University)

Poster 1-8 A Comparative Study of URL-Based and QR Image-Based Machine Learning for Quishing Detection
Kyunga Kang, Jaekyeom Kim and Jiyoon Kim (Gyeongsang National University)

Poster 1-9 Design and Implementation of an Overlay Methodology for K-RMF Security Control
Sarah Cho, Hojoon Kim and Sungil Oh (Hyundai Autoever)

Poster 1-10 Automated SBOM Generation for Automotive Embedded C/C++ Software: A Multi-Stage Extraction Approach with Bundled Library Detection
Sungil Oh, Hojoon Kim and Sarah Cho (Hyundai Autoever)

Poster 1-11 Route-Based Predictive Handover Authentication Protocol for Dual-Mobility NTN Environments
Seungbin Lee, Chanwoo Jung and Jiyoon Kim (Gyeongsang National University)

Poster 1-12 Design and Deployment of a Visualization System for International Cyber Ranges: A Case Study of APEX
Soohyun Jo (National Security Research Institute)

Poster 1-13 Adaptive Successive Narrowing Cost-Efficient Probe Location Shortlisting for Electromagnetic Side-Channel Evaluation
Jae Hyo Kim and Dong-Guk Han (Kookmin University)

Poster 1-14 Know Your Enemy, Know Yourself: Rethinking SASV in Realistic Multi-Attack Scenarios
Yowon Lee, Seongkyu Han, Thien-Phuc Doan and Souhwan Jung (Soongsil University)

Poster 1-15 A Behavior-Based Linkability Audit of Pseudonymized Discord Event Logs
Eunhee Jeong and Joonho Kwon (Pusan National University)

Poster 1-16 Who Spreads Toxicity?: Role-Dependent Diffusion in Team-Based Games
Hyunah Jung (Korea University)

Poster 1-17 poster_Flexible Side-Channel and SAT Attack Resistance in Logic Locking via Threshold Implementation
Yuki Shimmura, Yusaku Harada, Daiki Miyahara, Yang Li and Kazuo Sakiyama (The University of Electro-Communications)

Poster 1-18 Input Guard: An Input Guardrail Model for Prompt Injection Detection
Yelim Bae and Jinho Yoo (Sangmyung University)

Poster 1-19 poster_When Task Success Hides Tool Misuse: Analyzing Silent Compromise in Browser-Use Agents
Soobin Yim, Chanhyeok Lim, Hung Dinh-Xuan, Hyesong Choi and Souhwan Jung (Soongsil University)

Poster 1-20 PBFT-Based Lightweight Consensus Scheme for Sensor Event Consistency in Industrial IoT
Soowang Lee, Dayun Oh and Jiyoon Kim (Gyeongsang National University)

Poster 1-21 Evaluating Target Field-Value Manipulation in Structured Outputs via Indirect Prompt Injection
Seoyoung Son, Tae Uk Ha and Jinsoo Jang (Chungnam National University)

Poster 1-22 Effect of Code Representation Layers on LLM-Based Security Weakness Detection
Seojin Jeong, Hyunji Cho and Jinsoo Jang (Chungnam National University)

Poster 1-23 OA Sketch: High-Quality Feature Extraction Using Multi-Order Sketch for ML-Based NIDS
Jiyeon Kim, Yunju Bae, Sian Kim and Daehun Nyang (Ewha Womans University)

Poster 1-24 Transforming 5G Security Requirements into Implementation-Level Verdict Algorithms
Jieon Oh, Taeho Won and Ilsun You (Kookmin University), Hyungrok Jo and Junji Shikata (Yokohama National University)

Poster 1-25 Do Selected Compound Post-Processing Attacks Generalize? Evidence from a Frozen Quality-Constrained Protocol
Upjun Ha, Muyoung Kim and Howon Kim (Pusan National University)

Poster 1-26 Towards Evidence-Aware Triage of Korean-Language Social Engineering: A Risk-Primitive Framework and Preliminary Ablation Study
Seongyeop Kim and Jinho Yoo (Sangmyung University)

Poster 1-27 (PoseA Verification Method for Game Image Assets Using C2PA Manifest Validation and Perceptual Hash-Based Similarity Analysis
Young-Sang Lee, Jung-Min Park, Won-Yong Lee, Geon Park, Myeong-Jae Lee, Yun-Ho Kim, Jung-Guen Lee and Hye-Young Kim (Hongik University), Đại Thọ Đặng (Vietnam-Korea University of ICT, The University of Danang)

Poster 1-28 RF Self-Consistency-Based One-Class Anomaly Detection for Wireless Devices
Kang Beomseok (A.I. Cyber Security)

Poster 1-29 Unlearned but Still Identifiable: A Candidate-Set Identification Attack on Machine Unlearning for Korean Public-Figure Information
So Yeon Kim, Soo In Kim, Seong Hyun Min, Sun Woo Kim and Won Gyum Kim (AIDeep Co. Ltd.)

Poster 1-30 Practical Hardware Trojan Detection using Deep SVDD
Yongjae Kim and Dong-Guk Han (Kookmin University)

Poster 1-31 Anonymity-based User-Agent Identity Binding with Accountability
Su Jin Shin and Sang Uk Shin (Pukyong National University)

Poster 1-32 Sign-Decoupled Max Sketch: Preserving Gradient Direction under Differential Privacy
Gayoung Kim (Ewha Womans University)

Poster 1-33 Adaptive Feature-Level Privacy Protection for Efficient Privacy-Preserving Machine Learning
Yu-Na Kim, Hye-Yeon Shim and Il-Gu Lee (Sungshin Women's University)

Poster 1-34 Identifying First-Exposure Signals of Indirect Prompt Injection in Staged Latent Reasoning
Wonbin Cho and Hyunmin Song (Dankook University)

Poster 1-35 Training-Free Concept Unlearning for Text-to-Image Diffusion Models using Orthogonal Projection
Sooin Kim, Soyeon Kim, Seonghyun Min, Sunwoo Kim and Wongyum Kim (AIDeep Co. Ltd.)

Poster 1-36 Theoretical Analysis and Empirical Validation of Code Polymorphism-based Side-Channel Countermeasures
Sung-Bin Oh and Dong-Guk Han (Kookmin University)

Poster 1-37 SVF: A Stackelberg Game-Theoretic Strategic Verification Framework for Distributed Trust Systems
Ga-Yeong Kim and Hyun-Woo Choi (Sungshin Women's University)

Poster 1-38 Lightweight Authentication for Private 5G IIoT Networks: Evaluating EAP-EDHOC over Non-3GPP Access
I Wayan Adi Juliawan Pawana (Udayana University), Ilsun You (Kookmin University)

Poster 1-39 Quantifying AIBOM Inconsistencies Across Generation and Conversion Paths
Wonhong Jeong, Hyunsoo Oh, Sieun Ju, Jeongwoo Lee and Heejo Lee (Korea University)

Poster 1-40 Guarded LLM Decisions for Crash-Consistent Root-Key Rotation
Youngho Choi, Hoyeong Kim, Soonwook Choi, Youngsoo Kim and Dooho Choi (Korea University)

Poster 1-41 Hybrid Post-Quantum Cryptography for Next-Generation IoT Remote SIM Provisioning Based on GSMA SGP.32
Adi Panca Saputra Iskandar (Udayana University), Eldridge Aaron Miole and Ilsun You (Kookmin University), Hyungrok Jo and Junji Shikata (Yokohama National University)

Poster 1-42 Masked Multimodal Knowledge Distillation for Unsupervised Encrypted Traffic Anomaly Detection
Ilhwan Ji, Juhyeon Lee and Jung Taek Seo (Gachon Univ.)

Poster 1-43 A Quantitative Evaluation Methodology for Assessing Cyber Incident Response Capabilities in Nuclear Facilities
Dongjun Choi and Jung Taek Seo (Gachon Univ.)

Poster 1-44 A Cybersecurity Maturity Model for Nuclear Facilities
Hyeonseo Choe, Dongjun Choi, Ilhwan Ji and Jung Taek Seo(Gachon Univ.)

Poster 1-45 Privacy-Evaluability Diagnosis of Visual-Text Biometric Prompts in Safe-LLaVA
Shin Dawit, Thi-Thu-Huong Le, Agung Anak Ngurah Dharmawangsa, Andro Aprila Adiputra, Jang Hyunjin and Howon Kim (Pusan National University)

Poster Session 2 11:10–12:30, August 28 (Friday), 2026

Chairs: Kyungho Joo (Soongsil Univ.), Seyoung Lee (Kangwon Univ.)

Poster 2-1 Vexplain: Explainable Semantic Validation of Standalone VEX Documents
Hyunsoo Oh, Jeongwoo Lee, Duyeong Kim, Wonhong Jeong and Heejo Lee (Korea University)

Poster 2-2 LLM-Assisted Hybrid Intrusion Detection for EVN-IVN Gateways
Chaewan Hong, So-Eun Jeon and Il-Gu Lee (Sungshin Women's University)

Poster 2-3 Drone Anomaly Detection Based on Rule-Based and LSTM-Autoencoder, Using Flight Logs
Wookjin Choi, Seunghyun Cho, Seonghyeon An, Wonseok Lee, Rino Jo and Mee Lan Han (Korea University Sejong Campus)

Poster 2-4 EDHOC-Guard: A Specification-Based Intrusion Detection System for EDHOC on Constrained IoT
Eldridge Aaron Miole, Adi Panca Saputra Iskandar and Ilsun You (Kookmin University), Hyungrok Jo and Junji Shikata (Yokohama National University)

Poster 2-5 FTQC-Oriented Optimization of HIGHT Quantum Circuits Using the Lisa-Babu Adder
Jaeman An, Jinseob Oh, Youngsoo Kim and Dooho Choi (Korea University Sejong Campus)

Poster 2-6 Performance Comparison of Deep Learning Models for Profiling Attacks on Random Delay Protected AES
Yunwoo Song and Dongguk Han (Kookmin University)

Poster 2-7 A Comparative Analysis of Remote Attestation Protocols for Trusted Execution Environments
Jimin An, Myeongsub Jung, Gunseung Jin, Woojin Im and Jinsoo Jang (Chungnam National University)

Poster 2-8 SCVT: Automated Binary Vulnerability Detection and Exploit Chain Construction for Automotive ECU Firmware
Hojoon Kim, Sarah Cho and Sungil Oh (Hyundai Autoever)

Poster 2-9 A2A-VEX: Agent-to-Agent-based VEX for Validating of CVE
Ka-Kyung Kim, Ju-Yeon Park and Ieck-Chae Euom (Chonnam National University)

Poster 2-10 AIR: Diffusion-Based Physical Layer Security for JSCC-Enabled Semantic Communications
Seungha Jee and Il-Gu Lee (Sungshin Women's University)

Poster 2-11 Layer-Wise Detection of Partial Speech Deepfakes under Neural Codec Resynthesis
Jaeyoon Lee, Aram Kim and Sanghoon Jeon (The University of Suwon)

Poster 2-12 Diagnose, Then Fix: A Stage-Based Framework for LLM Cyber Agents
Mulyasari Wulan Silvani, Thi-Thu-Huong Le, Jaehan Cho and Howon Kim (Pusan National University)

Poster 2-13 Performance and Cost Efficiency Analysis of a Role- Specialized Lightweight LLM-based Multi-Agent System
Minji Cho, So-Hyun Park and Il-Gu Lee (Sungshin Women's University)

Poster 2-14 An OCSF-Based De-identification Method for Heterogeneous Logs Ensuring Integrity and Re-identification Resistance
Jangwoo Nam, Deukhun Kim and Jin Kwak (Ajou University)

Poster 2-15 Evaluating Residual Memory in Multimodal Unlearning with Counterfactual Visual Contexts
Hong Seoyun, Daewoong Kim, Jisu Lee and Changhoon Kim (Soongsil University)

Poster 2-16 LightGAE: Graph-Based Detection of Indirect Prompt Injection in Multi-Agent LLM Systems
Jin Yu, Lelisa Adeba Jilcha and Jin Kwak (Ajou University)

Poster 2-17 Scanner-Free Security Assessment for Container Image Risk Prediction in Air-Gapped Environments
Seo-Yeon Kyong, Ye-Rim Jeong and Il-Gu Lee (Sungshin Women's University)

Poster 2-18 HEC-A2A: A Hybrid Erasure Coding-based Trustworthy Agent-to-Agent Framework for Efficient Data Recovery
Ye-Sin Kim, Chae-Yeon Park and Il-Gu Lee (Sungshin Women's University)

Poster 2-19 Evaluating GNSS Spoofing Detection Models under Incomplete Signal Reception Conditions
Yejin Jang, Seung-A Park, Ho Yeong Kim, Dooho Choi and Mee Lan Han (Korea University)

Poster 2-20 A Without-ancilla Quantum Circuit Generator for Cryptographic S-boxes
Changmin Lee and Jinseob Oh (Korea University Sejong Campus), Youngsoo Kim and Dooho Choi (Korea University)

Poster 2-21 Channel-Aware Secure Semantic Communication via Adaptive Shamir's Secret Sharing
Chaesong Hong, So-Eun Jeon and Il-Gu Lee (Sungshin Women's University)

Poster 2-22 Post-Quantum SUCI for 6G: An Experimental Performance Evaluation of TR 33.703
Hoseok Kwon, Yongho Ko and Ilsun You (Kookmin University)

Poster 2-23 A Proposal for a Zero Trust Maturity Model for Agentic AI Environments
Jung-Hyun Mok and Sokjoon Lee (Gachon University)

Poster 2-24 CpsRAG: Operational Context-Based Explainable Anomaly Detection in Industrial Control Systems
Ju-Yeon Park, Ka-Kyung Kim and Ieck-Chae Euom (Chonnam National University)

Poster 2-25 Preserving Reasoning Faithfulness in LLMs via Null-Space Constrained DPO
Jisu Lee, Seong Gyu Park, Sohee Park, Wonho Lee and Daeseon Choi (Soongsil University)

Poster 2-26 Cross-Feature Security in AI-Assisted Coding: A Document-Sharing Case Study
Youngsuk Kim, Subin Lee, Minseo Park and Jinsoo Jang (Chungnam National University)

Poster 2-27 RF Trigger-based Active Electromagnetic Analysis for Device Identification and User Activity Recognition
Ye-Rim Jeong, Seo-Yeon Kyong and Il-Gu Lee (Sungshin Women's University)

Poster 2-28 Performance Analysis of Wireless Selfish Attacks Considering Capture Effect
Juwon Lee, Yeon-Ji Lee and Il-Gu Lee (Sungshin Women's University)

Poster 2-29 Lightweight Open-Set Screening of Unknown Traffic Flows for Resource-Constrained IoT Networks
Jungmin Moon and Il-Gu Lee (Sungshin Women's University)

Poster 2-30 Lightweight Deepfake Defense via PCA-Guided Target Embedding and Joint Perturbation
Yein Kim, Jungmin Moon and Il-Gu Lee (Sungshin Women's University)

Poster 2-31 Design and Preliminary Validation of a Multi-Layer Cybersecurity Testbed for Nuclear I&C Systems
Yesol Seo, Jisoo Lee, Hyeonbin Im, Aram Kim and Sanghoon Jeon (The University of Suwon)

Poster 2-32 Secure Wi-Fi Sensing via Secret-Key-Based Midamble Randomization
Chae Yeon Park and Il-Gu Lee (Sungshin Women's University)

Poster 2-33 Compositional Memory Poisoning of Self-Managed LLM Agents
Kim Hyunseo, Jeon Sua and Lee Taejin (Gachon University)

Poster 2-34 Analysis of Face Restoration Attacks on Differentially Compressed Images
Ahn Yehyeon, Jeon A-Young and Il-Gu Lee (Sungshin Women's University)

Poster 2-35 A Study on Semi-Supervised Intrusion Detection for Private 5G Networks Using a Testbed-Generated Flow Dataset
Halin Jeon (University of Science and Technology (UST)), Jonghoon Lee and Daesung Moon (Electronics and Telecommunications Research Institute)

Poster 2-36 Buffer Status Report Manipulation Attack on Resource Unit Scheduling for Wireless Local Area Network
Gyuha Son, Seungha Jee and Il-Gu Lee (Sungshin Women's University)

Poster 2-37 De-identifying Colloquial Korean in an LLM-Based Patient Intake System
Wonjae Kim, Jimin Seo and Jaekwon Lee (Kangwon National University)

Poster 2-38 Performance–Cost Trade-off Between Single-Agent and Multi-Agent Systems for Malicious IP Classification in Security Operations Centers
Tae-Kyeong Oh, Seo-Yi Kim and Il-Gu Lee (Sungshin Women's University)

Poster 2-39 What Aggregate False-Positive Rates Hide — Workflow Concentration and Run-to-Run Instability in Provenance-Based Intrusion Detection
Seonwoo Lee, Juyoung Lee and Taejin Lee (Gachon University)

Poster 2-40 A Study on the Cryptography Bill of Materials (CBOM) for Post-Quantum Migration
Ye Jin Kim, Dong Hyeon Kim, Yae Chan Kim, Seung-Min Ahn, A Yeong Choi and Dong Chan Kim (Kookmin University)

Poster 2-41 A Seven-Layer Vulnerability-Coordinate Framework for Pre-Deployment Safety Testing of Physical Agents
Hyeran Jung and Taejin Lee (Gachon University)

Poster 2-42 Enhancing Information Recovery from LeftoverLocals Leakage via LLM-Based Post-processing
Su-Yeong Kim, Chae-Won Song and Il-Gu Lee (Sungshin Women's University)

Poster 2-43 Agreement and Reliability of Multiple LLM-as-a-Judge Models in Medical Hallucination Detection
Gaeun Park, Seo-Yi Kim and Il-Gu Lee (Sungshin Women's University)

Poster 2-44 From Support to Interoperability: A Layered Assessment of PQC Deployment Readiness in Open-Source Cryptographic Libraries
Donghyun Kim, Donghyuk Kim, Jaewon Choi, Jeong Woo Lee and Yongje Shin (Semyung University)

Poster 2-45 Present Does Not Mean Used: What the Image and the Dialogue Contribute in a Multi-Turn Multimodal Safety Benchmark
Min-Gyu Kim, Howon Kim, Muyoung Kim, Ahmada Yusri Kadiptya and Subin Jang (Pusan National University)

Poster 2-46 A Survey on Evaluation Methods for Voice Phishing Detection
Hyunwoo Lee, Jinha Kim, Minki Jeon, Jiyeon Yang and Jinsoo Jang (Chungnam National University)

Poster 2-47 Evidence-Grounded Chain Reconstruction for Attack Investigation
Jeong Woo Lee (Semyung University), Hyeonjin An and Yebin Kim (Seoul Women's University), Yeong Choi (Korea University Sejong Campus)

Poster 2-48 Security Control Architecture based on Zero Trust Principle for Agentic AI Tool Calls
Jimin Park, Jung-Hyun Mok and Sokjoon Lee (Gachon University)

Poster 2-49 Evidence-Scoped Multi-Agent Network Defense with Predicate Chains
Yebin Kim and Hyeonjin An (Seoul Women's University), Yeong Choi (Korea University Sejong Campus), Jeongwoo Lee (Semyung University)

Registration

International Participants Payment Method

Registration Deadline: Registration Deadline: August 19, 2026 (UTC+09:00).
Participants can register for the conference until Registration Deadline: August 19, 2026 (UTC+09:00).
Please read carefully the registration guidelines below.
We use online registration to complete the registration for participants.
Click on the registration page link: https://kiisc.or.kr/payment/pay/331

Important Notice for Authors

At least one author of each accepted paper or poster must complete offline registration and present the work in person at the conference venue.
Author Registration Deadline: August 12, 2026 (UTC+09:00)

Registration Fee (International)

Online Attendance	Student	350 USD
	General Participants	
Offline Attendance	Student	350 USD
	Offline Attendance General Participants	500 USD

Credit Card
- Use a credit card payment system (EXIMBAY)

Wire Transfer
- beneficiary's name : KIISC
- beneficiary's account number : 754-01-0008-146
- beneficiary's bank : Kookmin Bank
- the branch name : Yeoksamyeok Branch
- SWIFT code : CZNBKRSE
- beneficiary' address : Room 909, Seongji Heights 3-Cha Bldg., 507, Nonhyeon-ro, Gangnam-gu, Seoul, Korea 06132

* This payment method is provided by Eximbay and is billed as www.eximbay.com.
* Note: Please note that the billing descriptor will be listed as EXIMBAY.COM.

Registration Includes

(Online) Registration includes (a) access to the online conference and (b) the access to the online proceedings.
(Offline) Registration includes: (a) participation in the offline conference (b) access to the online conference (c) access to the online proceedings (d) two lunches and one banquet (e) a souvenir.

Registration-Related Inquiries

If you have any questions regarding your registrations, please contact the Registration Company via email at kiisc@kiisc.or.kr

Registration Policy

At least one presenter per accepted paper, poster must register for the conference as an author.
We are unable to offer refunds, cancellations, or substitutions for any registrations for this event.

Registration

▶ 한국인 (국내거주인) 등록 안내

* 사전등록: 2026년 8월 19일(수)까지

온라인 참가	학생(전일제)/일반	250,000원
오프라인 참가	학생(전일제)	500,000원
	군·공무원	500,000원
	일반	700,000원
시니어(63세 이상) 종신회원	무료	0원

▶ 학회 홈페이지(www.kiisc.or.kr)접속 → 학회행사 → 사전등록바로가기 → WISA 2026 (Domestic) 클릭
(신용카드/계좌이체 결제 가능)

▶ 사전등록 송금처
- 예금주 : 한국정보보호학회
- 계좌번호 : (국민은행) 754-01-0008-146

※ 사전등록 시 등록비는 위의 계좌로 송금하시고 입금자명이 대리인명이나 단체명으로 송금되는 경우, 등록자 본인 확인이 어려울 수 있습니다. 이 경우 등록 시 메모란에 입금자명과 신청자명을 기재하거나, 이메일 또는 전화로 별도로 알려주시기 바랍니다.

* 계산서 신청 시, 익일 안으로 등록하신 이메일로 청구용 계산서가 발행됩니다. 영수용 계산서가 필요하신 경우, 사전에 학회로 연락 바랍니다.
* 등록확인서는 한국정보보호학회 홈페이지 상단 행사 등록 및 참가확인서 바로 가기를 클릭하신 후 등록 시 기재하신 성함과 이메일을 기재하시면 출력 가능합니다. (결제 완료 후 발행 가능)
* 참가확인서는 행사 종료 후 익일부터 한국정보보호학회 홈페이지 상단 행사 등록 및 참가확인서 바로 가기를 클릭하신 후 등록 시 기재하신 성함과 이메일을 기재하시면 출력 가능합니다. (결제 완료 후 발행 가능)
* 학생은 다른 소속이 없는 전일제(학부생/대학원생)에 한합니다.
* 학생의 경우 kiisc@kiisc.or.kr로 학생증 사본 송부 바랍니다.
* 군·공무원 등록(오프라인에 한함)은 주무관청에 소속 중인 공무원증 소지자에 한합니다. (국공립 교직원 제외) kiisc@kiisc.or.kr로 공무원증 사본 송부해 주시기 바랍니다.
* 시니어 무료등록은 학회 종신회원으로 1964년 12월 31일 이전 출생자에 한합니다.
* PAPER/POSTER 논문 발표자(저자 중 최소 1인) 는 오프라인 참석 비용으로 등록해주시기 바랍니다. (저자 사전등록 8월 12일까지)

▶ 등록 혜택
(오프라인) 오프라인 학술대회 참가, 온라인 학술대회 참석 링크, 온라인 프로시딩, 점심2회, 만찬 1회, 기념품 제공
(온라인) 온라인 학술대회 참석 링크, 온라인 프로시딩 제공

[행사 문의처] 한국정보보호학회 사무국 02-564-9333(내선2), kiisc@kiisc.or.kr
[계산서 문의처] 한국정보보호학회 사무국 02-564-9333(내선5), kiisc@kiisc.or.kr

사이버 위협정보 분석·공유 시스템

C-TAS System

Cyber Threat Analysis & Sharing

기업 보안성 강화를 위한 가장 쉬운 방법

회원가입 바로가기

ctas.krcert.or.kr

※ 간단한 회원가입으로 실시간 상황전파체계 (문자·이메일) 서비스를 제공 받아보세요.

실시간 상황전파

위협정보

보안공지

보안정보

과학기술정보통신부 KISA 한국인터넷진흥원

자율보안형 차세대 제어시스템 보안 솔루션

IMES

ICS Security for MAF by Eunicesoft

MITRE ATT&CK Framework 기반 자율보안형 차세대 제어시스템 보안 솔루션

제어시스템 사이버공격/이상징후에 대한 MAF와 인공지능 융합형 보안솔루션

IMES는 전력, 가스, 수도, 교통, 통신, 스마트공장 등 다양한 제어시스템의 보안위협을 지능적으로 탐지하고, 대응하는 자율보안형 차세대 제어시스템 보안 솔루션 입니다.

IMES는 IT/OT망의 다양한 프로토콜을 자동 수집·분석하여 보안이벤트로 처리합니다.

IMES는 XAI기반으로 이상징후를 탐지하여 사이버공격/고장/장애를 구분·탐지하고 신속한 대응을 지원함으로써 제어시스템의 가용성을 제고합니다.

IMES는 인공지능 기반 위협·위험헌팅을 통해 잠재 위협과 위험요소를 선제적으로 식별하고, 분석·판단·대응을 연계한 자율보안을 지원합니다.

특장점

- 표준 프로토콜 변환 및 데이터 분산처리
- IT프로토콜 및 OT프로토콜 수용
- IT-OT 이상(징후) 탐지 및 위협 예측
- 능동적 위협·위험헌팅 및 인공지능기반 제어보안위협분석
- 다양한 제어망 관리

주요기능

- 정상·이상·위험 상태 가시화
- 인공지능 기반 제어시스템 이상 (징후) 탐지 및 예측
- IT-OT 융합보안 위협 인텔리전스 분석
- 위협/위험헌팅 및 제어보안 위협분석
- 분석·판단·대응이 연계된 지능형 자율보안 수행

(주)유니스소프트

대전광역시 대덕구 대화로 106번길 66, 711호
(대화동, 펜타플렉스)

제품문의 : (주)유니스소프트 전략기획본부
T. 042-482-8367 E. eunice@eunicesoft.co.kr

BUILD *FEARLESSLY*, SCALE *INFINITELY*

Secure by default

Flexible by design

Limitless from the start

Build your **Agentic AI** on

aws

LG U+

Simply. U+

안심을 위한 일에 나중은 없기에.

U+를 믿고 사용해 주시는 고객님께.

안심을 위한 일에 나중은 없기에
LG유플러스 전 고객 대상으로
유심 업데이트·무료 교체를 진행합니다.

이는 더 안전한 통신 서비스를 위해
보안 수준을 강화하기 위한 예방적 조치입니다.

이번 유심 업데이트·교체로
번거로움을 드리게 된 점 양해 부탁드립니다.

LG유플러스는 이번 조치와 더불어
3년 내 사용 가능한 유심 교체 무료 쿠폰을 제공하고
원하실 때 언제든 유심을 업데이트하실 수 있도록 하겠습니다.

앞으로도 고객 여러분의 안전한 통신생활을
지키기 위해 최선을 다하겠습니다.

누구나 쉽게 진행하실 수 있도록 간편한 유심 업데이트 · 교체 방법을 준비했습니다.

온라인 간편 업데이트 방법

기간 : 4월 13일(월)부터 가능

U+one 앱 접속 및 로그인 > 우측 상단 메뉴 내 [유심 업데이트] 클릭 > [업데이트 시작하기] 버튼 클릭 > 종료 후, 휴대폰 껐다 켜고 문자 확인

U+one 앱 접속하기

* 온라인 간편 업데이트는 매일 오전 8시~오후 10시까지 가능합니다.

❋ 온라인 업데이트가 어렵거나 유심 교체가 필요한 경우 매장을 방문해주세요.

[기간 : 4월 8일(수)부터 방문 예약 가능, 4월 13일(월)부터 매장 방문 가능]

1. U+one 앱 / 홈페이지 접속 및 로그인
2. 우측 상단 메뉴 내 [유심 업데이트] 클릭
3. 본인 인증 후 [매장 방문 예약 버튼] 클릭
4. 방문을 원하시는 날짜, 시간대, 매장 선택
5. 방문 예약 완료 후 확정 문자 메시지 확인
6. 방문 예약일에 선택한 매장으로 방문

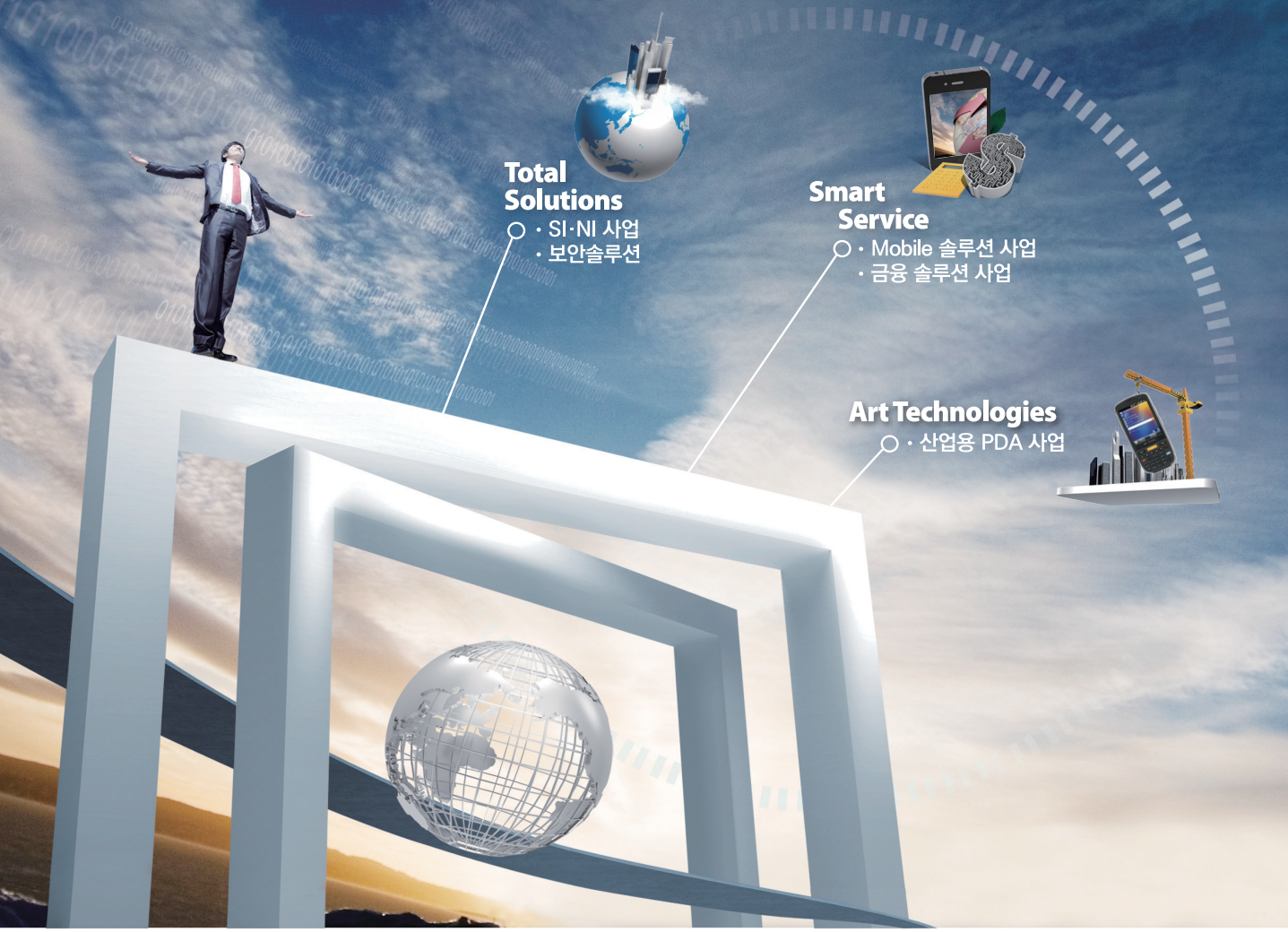
안심하고 진행하시도록 24시간 유심 상담센터(080-864-1010)를 운영 중입니다.

더 자세히 알아보기

Global IT Leader!

모든 비즈니스 영역을 통합하는 통찰력으로
고객의 니즈를 완벽히 분석한 최적의 서비스로
미래를 선도하는 최첨단 기술력으로

미래의 가치를 먼저 생각하는 기업



큰大믿을信
대신정보통신주식회사 Daishin Information & Communications Co., Ltd.

서울특별시 금천구 가산디지털1로 205-28 대신IT타워 14층 Tel 02-2107-5000 Fax 02-2107-0515 www.dsic.co.kr

ITCEN ENTEC

Inspire with Technology

세상을 변화시키는 기술

우리는 고객에 대한 이해를 넘어서,
변화의 흐름을 읽고, 멈추지 않는 도전을 통해
지식과 기술로 세상을 혁신합니다.

깊은 신뢰감으로 늘 고객과 함께하고,
기술, 사람, 기업간 새로운 연결 속에서
새로운 경험과 가치를 발굴하며
아이티센은 지속가능한 변화를 실현합니다.

더 나은 내일을 향한 기술로
고객의 삶에 필요한 가치를 창조하는 것.
아이티센이 생각하는
우리의 역할이자 우리의 존재 이유입니다.

신뢰받는 **AI의 시작**, 철저한 **신원 인증**부터 빈틈없는 **권한제어**까지

오키 통합인증플랫폼

Okey

AI 에이전트 인증

OTP

생체 인증

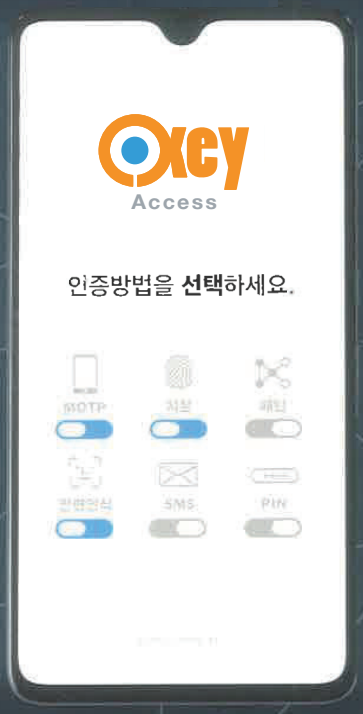
PKI 인증

기기 인증

PIN

QR 코드

위치기반 인증



AI 에이전트, 편리하지만 불안하시죠?

사람도 AI도, OKEY 인증플랫폼이 확실하게 인증합니다.

KSA
(주)한국보안인증

서울시 성동구 성수2가 280-6 생각공장 데시앙플렉스 1709-1710호

www.kcert.co.kr

AI·디지털 전환 시대, AX·DX 융합
보안 전문기업

COrporation ON by TEChnology

COONTEC

쿤텍은 AX·DX 융합보안 전문기업으로 디지털 전환에서 필요한 융합 산업의 보안 관련 기술력을 중심으로 고객의 디지털 연결 가속화를 돕고, 더 나아가 쿤텍의 사람과 기술, 기업문화를 통해 시대 전환을 이끌고 있습니다.

쿤텍은 보안 솔루션 핵심 기술에 대한 폭넓은 이해도를 보유한 전문 인력을 기반으로 진화하는 보안 위협에 대한 빠르고 전문적인 대응 방안을 제시하여 다양한 환경의 고객 자산을 보호합니다.

공급망 보안

- AEZIZ
- AEZIZ RMS
- mend.io
- sonatype

인프라 보안

- TeraGRID
- CLAROTY

AI 보안

- AEZIZ AICA
- zscaler
- ASH ModelGate

조선/해양

- TGM
- KR-CyberOne
- KR-Tester

공격표면 보안

- Cyber Hawk Eye
- penzzer

임베디드 보안

- SecureALPS
- SECURE-iC

임베디드 가상화

- FastVLabs
- RAPITA Systems
- PikeOS RTOS & Hypervisor

5G 특화망

- NXTCore

임베디드 개발검증

- TASKING
- LDRA

비공개 데이터 연결

- NEXA Next data EXperience for AI

COONTEC Co., Ltd. 쿤텍 주식회사
T. 031-751-9088 E. marketing@coontec.com H. www.coontec.com
A. 경기도 성남시 수정구 창업로 54, LH기업성장센터 가동 609호

COONTEC

ⓒ 2026. COONTEC Co., Ltd. All rights reserved.

Global Platform Based
Service Company

TOTAL *Network* SERVICES

트리즈엔은
기술과 사람, 세상을 연결하는
지능형 네트워크의 미래를 열어갑니다.

TRIZN

Global Platform
경계를 넘어
세상을 연결하는
세계 최대 서비스 플랫폼

Smart & Intelligent
지능형 AI 기술로
토탈 네트워크의
새로운 가치 창출

Future Innovation
데이터와 기술로
미래를 연결하는
지속가능한 디지털 혁신

㈜트리즈엔 경기도 수원시 영통구 대학3로 4번길 15 테라스힐 2F / 전화 (031) 247-5966 / Fax (031) 212-5903
www.trizn.kr