Performance Analysis of Signing Algorithms and Integrity Enhancement Techniques for MAVLink in PX4

KyeongMin Lee
School of Computer Science and Engineering
Pusan National University
Busan, South Korea
min99819@pusan.ac.kr

YeonJeong Hwang
School of Computer Science and Engineering
Pusan National University
Busan, South Korea
yeonjeong@islab.re.kr

Thi-Thu-Huong Le
Blockchain Platform Research Center
Pusan National University
Busan, South Korea
lehuong7885@gmail.com

JunYoung Son
School of Computer Science and Engineering
Pusan National University
Busan, South Korea
jysonpaperinfo@gmail.com

***Abstract*—The MAVLink protocol is widely adopted for communication between drones and ground control stations (GCS) due to its efficiency in resource-constrained environments. This paper presents a comprehensive performance analysis of various cryptographic algorithms applied to the MAVLink signature field in PX4, with the goal of identifying the most suitable algorithm for ensuring secure drone communication. The evaluation considers both hash-based message authentication codes (MACs), specifically SHA256, Blake2b, and Poly1305, and digital signature algorithms (DSA), including Ed25519, Ed448, and ECDSA. The analysis highlights the trade-offs between security and computational efficiency, particularly in the context of real-time drone operations. Based on the findings, the paper proposes several techniques to enhance message integrity and security in MAVLink, offering solutions that are practical for implementation in embedded systems with limited resources. This study provides valuable insights into optimizing the security and performance of drone communication systems, guiding the effective adoption of cryptographic methods in unmanned aerial vehicle (UAV) networks.**

***Index Terms*—UAV, MAVLink, PX4, Signing, Data Integrity, Verification.**

I. Introduction

An Unmanned Aerial Vehicle (UAV) is an aircraft that operates without a human pilot on board, as defined by the Office of the Secretary of Defense. c, UAVs can be equipped with various instruments such as sensors and cameras, making them applicable across a wide range of fields [1]. In particular, UAVs have become critical assets in modern warfare [2], as they enable precision.

On targeting over long distances. Given their dual-use potential for both surveillance and offensive operations, securing drone communication protocols has become an increasingly urgent research priority.

UAVs communicate with Ground Control Stations (GCS) through various communication protocols [3]. Among these, MAVLink [4] is a widely used open-source protocol, developed and maintained by the PX4 project. MAVLink has proven to be suitable for resource-constrained environments, such as those encountered in drones. However, this efficiency comes with a security trade-off, as the protocol lacks robust protections against eavesdropping and impersonation attacks [5]. Although MAVLink 2.0 introduced support for message signing to provide authentication, the significant computational overhead associated with this security feature has led to its omission in many practical implementations.

MAVLink currently relies on the CRC-16 algorithm for integrity checking. However, because CRC-16 does not use a secret key to calculate checksums and follows a linear structure, it is vulnerable to forgeries, enabling unauthorized attackers to bypass the integrity verification process [6].

In this paper, we implement and evaluate various cryptographic hash functions for the MAVLink 2.0 signature field, comparing their performance characteristics. Our primary contributions are as follows:

- Implement and compare multiple hashing algorithms to identify those with the lowest and highest computational overhead in the context of MAVLink 2.0.

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core Talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(RS-2020-II201797) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). And this research was also supported by Korea Planning & Evaluation Institute of Industrial Technology(KEIT) grant funded by the Korea government (MOTIE) (No.RS-2024-00508087, Development of Virtual Environment-based Cyber Security Verification Technology).
Corresponding author: JunYoung Son (jysonpaperinfo.com)

- Evaluate the performance characteristics of MAVLink 2.0, both with and without signature verification enabled.
- Experimentally demonstrate how the MAVLink 2.0 signature field mitigates hijacking attacks.
- Analyze potential attack vectors in MAVLink communication protocols, comparing scenarios with and without the signature field implementation.

This paper aims to contribute to the secure communication between UAVs and GCS by identifying hashing algorithms with the lowest computational overhead, thereby facilitating the practical implementation of the signature field in MAVLink 2.0 messages.

II. Previous Work

Prior to this study, several related works were conducted on securing MAVLink communication. [7], [8]. One notable approach involved applying the ECDH (Elliptic Curve Diffie-Hellman) algorithm over the MAVLink communication channel to establish a shared symmetric secret key [9]. This key was then used to generate the signature field in MAVLink 2.0. The overheads associated with the signature were evaluated, and since only minimal overhead was observed, it was concluded that enabling the signature is feasible in terms of performance.

A. Enabling Signature

In MAVLink 2.0, the introduction of the signature field provides data integrity. However, this feature is not natively utilized in the PX4 software [10]. To address this, the process of generating and verifying the signature field was implemented for bidirectional communication between PX4 and the Ground Control Station (GCS) application, QGroundControl (QGC) [11]. The symmetric key used for this process was generated via the ECDH algorithm, and the signature field was implemented following the MAVLink 2.0 standard.

To briefly explain the overall process of key exchange and signature generation/verification: upon establishing a connection, a random value is used to generate a private key, and elliptic curve operations are performed to create a corresponding public key. The public keys are then exchanged over a public channel as intended by the user. Each party uses its own private key and the received public key to generate a shared secret key [12]. This shared key, combined with the SHA256 hash function, is used to generate a Message Authentication Code (MAC) for the message, which is appended to the message before transmission. Upon receiving the message, the recipient performs a verification process to ensure integrity.

B. Simulated Attack

To test the integrity verification and authentication process, an attack scenario was simulated in which both a malicious QGC and a legitimate QGC were connected to the same drone, as shown in Figure 1. Before the signature field was enabled, both QGCs were able to transmit control commands to the drone successfully [13]. However, once the signature field was enabled, all control commands from the malicious QGC were ignored [7]. This demonstrates the effectiveness of the signature in preventing unauthorized control.

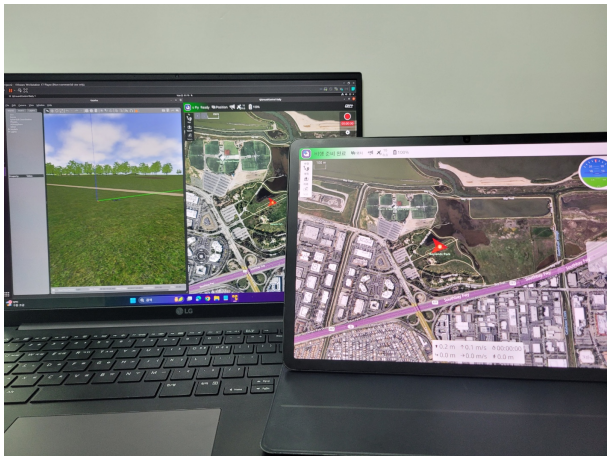

Fig. 1. Simulation of Attack Scenario: Command transmission with and without signature field enabled

Furthermore, if the key used for signature generation is exposed, the attacker could also generate a valid MAC, compromising the integrity of the communication. To mitigate this, a mechanism was implemented that allows the user to update the key, preventing potential key exposure scenarios manually.

C. Evaluation of Signature Suitability

To assess the suitability of the signature field in MAVLink communication for practical use, we analyzed its impact on system performance. Flight logs were used to measure CPU and RAM usage, with results showing negligible differences compared to when the signature field was inactive [14]. Additionally, the processing time for control and connection-related messages was evaluated to quantify the overhead introduced by signature verification, as shown in Figure 2. The overhead was observed to be approximately 1.2 to 1.5 microseconds, which is considered insignificant for practical drone operations [15].

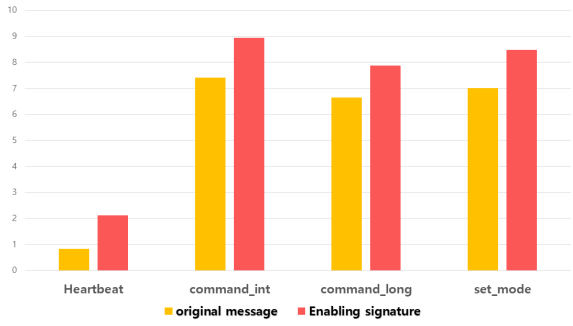

Fig. 2. Processing Time Comparison: MAVLink communication with and without signature verification

III. Background

Signing provides a mechanism to verify message authenticity between communication peers by generating a hash of concatenated elements such as timestamp, sequence number, and other relevant data. The overhead introduced to the original communication depends on the chosen hashing algorithm. There are two primary approaches to implementing MAVLink signatures: Standard Hash Functions and Digital Signature Algorithms.

A. Standard Hash Functions

Standard Hash Functions offer basic message authentication using algorithms like SHA-256 or HMAC. The following are common standard hash functions:

- **HMAC-SHA256** [16] utilizes a two-stage hashing process, combining a secret key with the message data. This process involves two XOR operations: one between the key and inner padding (0x36) and another between the key and outer padding (0x5c). The inner hash is generated by applying SHA-256 to the inner key concatenated with the message data. The final HMAC value is produced by applying a second SHA-256 operation on the outer key concatenated with the inner hash.
- **HMAC-Blake2b** [17] follows a similar structure to HMAC-SHA256 but uses Blake2b as the underlying hash function. Blake2b processes the input through a state vector v and a compression function G, integrating message blocks with the state vector to produce intermediate hash values. Blake2b offers security guarantees comparable to SHA3 while outperforming SHA-256 in terms of efficiency.
- **Poly1305** [18] is recognized for its efficient performance among Message Authentication Code (MAC) algorithms. It operates with a 256-bit key split into two 128-bit components, r and s, where r undergoes clamping with a specific mask. Messages are processed in 128-bit blocks, and the hash computation follows the formula:

$$h = (h + mi) \times r \mod p \quad \text{where} \quad p = 2^{130} - 5 \quad (1)$$

  The final authentication tag is generated by adding s to the accumulated hash.

B. Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) provides enhanced security through public key cryptography. While DSA offers superior security features such as non-repudiation (ensuring a sender cannot deny sending a message), its computational overhead makes it challenging to implement in resource-constrained environments. Among the DSA algorithms, ECDSA (Elliptic Curve Digital Signature Algorithm) is particularly known for its adaptability to lightweight environments.

- **ECDSA** [19] (Elliptic Curve Digital Signature Algorithm) offers security levels equivalent to RSA but with significantly shorter key lengths. Its security relies on the elliptic curve discrete logarithm problem (ECDLP). During signature generation, ECDSA requires a random nonce k, where the unpredictability and uniqueness of k are crucial for security. Poor randomness or reuse of k can lead to private key compromise, which has caused several notable security incidents. The need for high-quality randomness in k generation can impact the reliability and security of implementations.
- **EdDSA** [20] (Edwards-curve Digital Signature Algorithm) is a variant of Schnorr signatures using twisted Edwards curves, which is not directly derived from ECDSA. EdDSA improves upon ECDSA by using deterministic k generation via a hash function, mitigating risks associated with random number generation while maintaining security. It uses Edwards curves optimized for efficient and secure implementation. The most widely used variants are Ed25519, which provides 128-bit security using Curve25519, and Ed448-Goldilocks, which offers 224-bit security using Curve448. Ed25519 is particularly noted for its combination of high performance, strong security, and resistance to timing attacks. Ed448, while providing a higher security margin, results in reduced performance, making it suitable for long-term security applications.

In this paper, we compare the performance of the aforementioned cryptographic algorithms to determine the most suitable one for resource-constrained environments, such as UAVs. The experimental results and analysis are presented in Section IV.

IV. Experiment

A. Environment Setup

The implementation environment was based on the Ubuntu 20.04 operating system. The tools and their respective versions are detailed in Table I.

TABLE I
Setup Implementation

Tool	Version
Ubuntu	20.04
PX4-Autopilot	1.14.3
ROS	Noetic
QGroundControl (QGC)	4.4.1
Gazebo	11.0

B. Implementation

To evaluate optimization methods for addressing the overhead introduced by signature verification in MAVLink communication, the cryptographic algorithms discussed in the previous section were implemented, and their processing times were measured.

For hash-based MAC algorithms, the Blake2b, Poly1305, and Ed25519 implementations utilized the Monocypher library [21], while the ECDSA and Ed448 implementations were based on OpenSSL. These libraries were chosen for their performance and compatibility with resource-constrained systems.

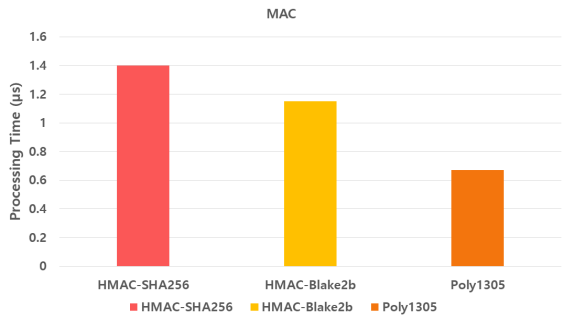

Fig. 3. Processing Time for Hash-based MAC Algorithms

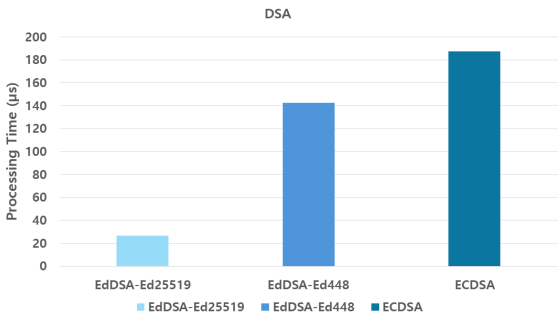

Fig. 4. Processing Time for DSA Algorithms

C. Measurement of Message Processing Time

The first set of measurements focused on the processing time for algorithms using MAC for verification. The results are as follows:

- **HMAC-SHA256**: Average verification time of approximately 1.4 microseconds.
- **HMAC-Blake2b**: Average verification time of approximately 1.15 microseconds.
- **Poly1305**: Demonstrated the shortest verification time at 0.67 microseconds.

Figure 3 illustrates the comparison of processing times for these MAC algorithms. The results indicate that Poly1305, with its minimal verification time, is the most suitable for resource-constrained environments, such as drones.

Next, the processing times for Digital Signature Algorithms (DSAs) were measured. The results are as follows:

- **Ed25519**: Average verification time of 26.7 microseconds.
- **Ed448**: Average verification time of 142.8 microseconds.
- **ECDSA**: Average verification time of 187.5 microseconds.

Figure 4 presents a comparison of the processing times for these DSA algorithms. While DSA algorithms provide enhanced security, their significantly higher overhead poses challenges for real-time implementation in drone systems. As a result, it is recommended to apply DSA algorithms only to security-sensitive messages while utilizing hash-based MAC algorithms for general message integrity to balance security and performance.

V. Discussion

The experimental results presented in Section 5 highlight the trade-offs between performance and security when using hash-based MAC algorithms and digital signature algorithms (DSA) for UAV communication.

Hash-based MAC algorithms, such as HMAC-SHA256, HMAC-Blake2b, and Poly1305, demonstrated consistent performance within the range of 0 to 1.5 microseconds. Among these, Poly1305 exhibited the shortest latency at approximately 0.6 microseconds, attributed to its efficient single-pass design. Unlike HMAC variants, which require additional operations such as inner and outer padding (ipad and opad), Poly1305 achieves minimal computational overhead by eliminating redundant hash function computations. This makes it an ideal choice for resource-constrained UAV systems where low latency is critical.

In contrast, the DSA algorithms, including ECDSA, Ed25519, and Ed448, exhibited significantly higher computational overhead, with latencies ranging from 26.7 microseconds (Ed25519) to 187.5 microseconds (ECDSA). Despite these higher latencies, DSAs provide essential security properties, such as non-repudiation and strong authentication. Notably, Ed25519 outperformed other DSAs by achieving nearly one-ninth the latency of ECDSA, making it the most efficient option within this category.

The results emphasize the importance of balancing security requirements with performance limitations. While DSAs are well-suited for security-critical messages, their higher processing times make them less practical for routine message authentication in UAV systems. Conversely, hash-based MAC algorithms provide a lightweight solution for ensuring message integrity, albeit without the advanced security features of DSAs.

VI. Conclusion

In this paper, we conducted a comparative performance analysis of hash-based MAC algorithms and digital signature algorithms (DSA) in the context of UAV communication. The experiments revealed a substantial performance disparity between the two categories, with DSA algorithms exhibiting latencies of up to 200 microseconds, while hash-based MAC algorithms consistently maintained processing times under 1.5 microseconds. This highlights the trade-offs between the high-security guarantees provided by DSAs and the low-latency advantages of hash-based MAC algorithms.

Poly1305 emerged as the most efficient hash-based MAC algorithm, achieving a latency of approximately 0.6 microseconds. This performance advantage can be attributed to its streamlined, single-pass design, which minimizes computational overhead compared to HMAC variants. For digital signature algorithms, Ed25519 demonstrated superior efficiency,

achieving significantly lower latencies compared to ECDSA and Ed448 while maintaining robust security properties. Its deterministic signature generation offers enhanced stability and strong authentication, making it particularly suitable for UAV environments where both security and efficiency are critical.

Based on these findings, we recommend the adoption of Ed25519 for security-critical messages such as key distribution and authentication protocols. Its ability to provide essential security properties, including non-repudiation and access control, makes it ideal for safeguarding against communication hijacking attacks in UAV systems. Conversely, for non-critical routine messages, Poly1305 is recommended due to its minimal latency and efficient message authentication, which ensure an optimal balance between performance and security without the additional overhead of digital signature algorithms.

For future work, the scope of analysis should be extended to evaluate algorithm performance across diverse network conditions, such as varying latency and throughput. Additionally, a deeper investigation into potential attack vectors in unauthenticated communication channels is crucial for identifying and mitigating vulnerabilities. These efforts will contribute to the development of more robust and secure communication protocols tailored to the unique demands of UAV systems, ensuring both high performance and strong security.

REFERENCES

[1] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE communications surveys & tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[2] C. Daggett, "Drone disorientations: How "unmanned" weapons queer the experience of killing in war," *International Feminist Journal of Politics*, vol. 17, no. 3, pp. 361–379, 2015.

[3] S. Atoev, K.-R. Kwon, S.-H. Lee, and K.-S. Moon, "Data analysis of the mavlink communication protocol," in *2017 International Conference on Information Science and Communications Technologies (ICISCT)*. IEEE, 2017, pp. 1–3.

[4] "Mavlink." [Online]. Available: https://mavlink.io/en/

[5] H. Xu, H. Zhang, J. Sun, W. Xu, W. Wang, H. Li, and J. Zhang, "Experimental analysis of mavlink protocol vulnerability on uavs security experiment platform," in *2021 3rd International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2021, pp. 1–6.

[6] E. Dubrova, M. Näslund, G. Selander, and F. Lindqvist, "Cryptographically secure crc for lightweight message authentication," *Cryptology ePrint Archive*, 2015.

[7] I. Hughes, A. Pupo, J. Wynd, Z. Thurlow, C. Ivancik, and Y. Wang, "Securing the unprotected: Enhancing heartbeat messaging for mavlink uav communications," in *2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2024, pp. 1–6.

[8] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, "Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 43 203–43 212, 2018.

[9] P. Mykytyn, M. Brzozowski, Z. Dyka, and P. Langendörfer, "Towards secure and reliable heterogeneous real-time telemetry communication in autonomous uav swarms," *arXiv preprint arXiv:2404.07557*, 2024.

[10] N. Sabuwala and R. D. Daruwala, "Securing unmanned aerial vehicles by encrypting mavlink protocol," in *2022 IEEE Bombay Section Signature Conference (IBSSC)*. IEEE, 2022, pp. 1–6.

[11] E. Giorgi, "Implementazione di autopilota per sistemi uav/ugv basati su smartphone android= development of autopilot fro uav/ugv system on high-end android smartphone," Ph.D. dissertation, Politecnico di Torino, 2019.

[12] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.

[13] M. Bakirci, "A novel swarm unmanned aerial vehicle system: Incorporating autonomous flight, real-time object detection, and coordinated intelligence for enhanced performance." *Traitement du Signal*, vol. 40, no. 5, 2023.

[14] C. Verbowski, E. Kiciman, A. Kumar, B. Daniels, S. Lu, J. Lee, Y.-M. Wang, and R. Roussev, "Flight data recorder: Monitoring persistent-state interactions to improve systems management," in *Proceedings of the 7th symposium on Operating systems design and implementation*, 2006, pp. 117–130.

[15] G. Nathan, "Examination of drone localization performance with commercially available embedded gps sensors," Master's thesis, University of Washington, 2024.

[16] D. Ravilla and C. S. R. Putta, "Implementation of hmac-sha256 algorithm for hybrid routing protocols in manets," in *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)*. IEEE, 2015, pp. 154–159.

[17] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "Blake2: simpler, smaller, fast as md5," in *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings 11*. Springer, 2013, pp. 119–135.

[18] D. J. Bernstein, "The poly1305-aes message-authentication code," in *International workshop on fast software encryption*. Springer, 2005, pp. 32–49.

[19] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, pp. 36–63, 2001.

[20] S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (eddsa)," Tech. Rep., 2017.

[21] "Monocypher." [Online]. Available: https://monocypher.org/