Enhancing MAVLink Security: Implementation and Performance Evaluation of Encryption on a Drone Testbed

Adheeba Thahsin, Ananthapadmanabhan A., Saketh Pathak, Arnab Maity, and Gaurav S. Kasbekar

***Abstract*—Unmanned Aerial Vehicles (UAVs) or drones are being extensively deployed in various military as well as civilian applications. The Micro Air Vehicle Link (MAVLink) protocol is widely used for communication between a UAV and a Ground Control Station (GCS). However, under this protocol, the UAV and the GCS communicate through an unencrypted channel, due to which malicious actors can eavesdrop on the channel with relative ease. Numerous research efforts have aimed to integrate encryption into the MAVLink protocol, but these works have been confined to theoretical analyses or simulation-based demonstrations. This paper demonstrates the integration of Advanced Encryption Standard (AES) encryption with counter (CTR) mode into the MAVLink protocol on a drone testbed. We conducted a number of flight tests to assess the drone functionality with and without encryption. Our results show that the integration of AES-CTR encryption into MAVLink causes only minimal overheads in the system performance. In particular, the encryption and decryption times are small and there is only a slight increase in the memory consumption and CPU load due to encryption. Our validation of the AES-CTR encryption scheme on a drone testbed demonstrates that it is a secure and practical solution for real-world applications.**

***Index Terms*—Unmanned Aerial Vehicle (UAV), MAVLink, Encryption, Testbed, Advanced Encryption Standard (AES)**

I. Introduction

Unmanned Aerial Vehicles (UAVs) or drones are being extensively deployed in various military as well as civilian applications, e.g., search and destroy operations, border surveillance, managing wildfire, as relays for ad hoc networks, in disaster monitoring, remote sensing, traffic monitoring, etc. [1]. The typical components in a UAV system include the UAV itself, a Ground Control Station (GCS), sensors, and a communication platform that facilitates seamless interactions among them [2]. A UAV can operate autonomously based on its pre-programmed software or be supervised and managed remotely from the ground by the GCS. Since there is no human pilot onboard, it is crucial to establish an effective communication link between the GCS and the UAV to accomplish critical missions.

The communication between the UAV and the GCS occurs through various means, such as a radio frequency (RF) link, satellite link, or cellular network, depending on the range and requirements of the mission [1]. The information exchanged between the UAV and the GCS includes command and control information, telemetry data, payload data, and status alerts. These pieces of information are exchanged and interpreted utilizing standardized communication protocols to ensure reliable and efficient data exchange during missions [3].

Due to the significant amount and sensitive nature of UAV data, it has become a prime target for cyber-attacks. Ensuring the security of the communication between the UAV and the GCS is crucial for the success of UAV missions, as it frequently contains valuable information sought after by adversaries [4]. Malicious actors could exploit vulnerabilities in communication protocols or software systems to gain unauthorized access, manipulate commands, or intercept sensitive data transmitted between the GCS and the UAV [3], [5]. Moreover, the reliance on wireless communication channels exposes the system to risks such as interception, spoofing, and compromise of the integrity and reliability of the communication link [6], [7]. Therefore, safeguarding the control and data exchange between the GCS and the UAV against security threats is paramount to ensuring the safety, security, and integrity of sensitive information exchanged between them.

The Micro Air Vehicle Link (MAVLink) [8] protocol stands out as one of the most widely embraced protocols for communication between the UAV and the GCS. Despite its robustness and widespread adoption, the MAVLink communication protocol lacks a complete security mechanism [6], [9], rendering it susceptible to attacks. In its earlier version, MAVLink 1.0, the protocol lacked native support for authentication and authorization and was reported in the National Vulnerability Database [10]. However, the current version, MAVLink 2.0, has made significant improvements by incorporating an authentication mechanism based on message signing and a message authentication code (HMAC) for integrity purposes [9]. Despite these advancements, encryption support to provide confidentiality remains absent due to concerns about its impact on performance [11]. Consequently, the GCS still communicates with the UAV through an unencrypted channel. Without encryption, malicious actors equipped with appropriate transceivers can eavesdrop on the communication channel and launch attacks on UAV systems with relative ease.

Numerous research efforts have aimed to tackle these challenges by integrating encryption into the MAVLink protocol [11]–[16] (see Section II). However, the majority of these works have been confined to theoretical analyses or simulation-based demonstrations. To the best of our knowledge, there has been a notable absence of real-world demonstrations showcasing the incorporation of encryption in the MAVLink protocol used in actual drones. In this paper, we endeavor to bridge this gap by implementing encryption in MAVLink as well as evaluating its performance via extensive practical experiments conducted on a drone testbed.

This paper demonstrates the integration of Advanced Encryption Standard (AES) encryption with counter (CTR) mode

A. Thahsin, S. Pathak and A. Maity are with the Department of Aerospace Engineering, Indian Institute of Technology (IIT) Bombay, Mumbai 400076, Maharashtra, India. Ananthapadmanabhan A. and G. Kasbekar are with the Department of Electrical Engineering, IIT Bombay. Their email addresses are adeebathahsin@gmail.com, saketpathak321@gmail.com, arnab.maity@iitb.ac.in, anantha9102002@gmail.com, and gskasbekar@ee.iitb.ac.in, respectively. The contributions of A. Thahsin, S. Pathak, A. Maity, and G. Kasbekar have been supported in part by the project with code RGSTC01-001.

into the MAVLink protocol on a drone testbed. We selected AES encryption due to the strong security it provides and its compatibility with the hardware of drone flight controller boards such as the Pixhawk Cube Orange Plus [17], which comes with built-in support for AES accelerator modules. We conducted a number of flight tests to assess the drone functionality with and without encryption and evaluated its performance under various conditions. Our results show that the integration of AES with CTR mode encryption into MAVLink causes only minimal overheads in the system performance. In particular, the encryption (respectively, decryption) time was found to be only 124.6 (respectively, 121.6) microseconds on average, and the increase in memory consumption (respectively, CPU usage) was found to be only $0.73\,\%$ (respectively, $4.33\,\%$) on average. Our validation of the AES with CTR mode encryption scheme on a drone testbed demonstrates that it is a secure and practical solution for real-world applications.

The rest of this paper is organized as follows. Section II provides a review of related prior literature. Section III provides background on the MAVLink protocol and AES encryption. Section IV describes the system model and problem formulation. Subsequently, Section V provides details of the implementation, and Section VI describes our experimental setup. Section VII presents our experimental results and Section VIII provides conclusions and directions for future research.

II. RELATED WORK

In [14], MAVSec was introduced to enhance MAVLink communication security for Ardupilot [18] and PX4 [19] autopilots. The authors implemented various encryption algorithms, including AES-CBC, AES-CTR, RC4, and ChaCha20, within the Ardupilot Simulation in the Loop (SITL) setup, comparing their performance. Similarly, [15] evaluated the performance of various algorithms, including ChaCha20, encryption by Navid, and DMAV, proposed for MAVLink security. Utilizing the Gazebo simulation environment, the authors conducted a case study to assess the algorithms' performance in terms of packet transfer speed, memory utilization, and CPU consumption.

In [11], the potential for unauthorized access to MAVLink packets was illustrated. Building on this case study, the authors proposed an encryption algorithm to safeguard the MAVLink protocol, which included mapping to ASCII characters and implementing Caesar cipher encryption. Their encryption method effectively thwarted unauthorized access.

A key exchange procedure for MAVLink encryption was proposed in [16] to improve the security measures proposed in MAVSec [14]. The authors also proposed fourteen lightweight cryptography algorithms for use with MAVLink and compared their performance with that of ChaCha20 proposed in MAVSec. Then they analyzed the performance of the Speck128/192 encryption algorithm along with the key exchange procedure and found that the key exchange phase does not add much overhead to the system.

In [12], an algorithm called DMAV was proposed based on Dynamic DNA coding to secure the MAVLink protocol. The authors also conducted a performance analysis of the protocol based on the number of packets sent and the memory consumption and found that their encryption algorithm does not cause much overhead.

To enhance MAVLink security, [20] introduced a novel mechanism named MAV-DTLS. This algorithm was implemented on Ardupilot, demonstrating its resilience against attacks in a simulation environment. The authors also verified that MAV-DTLS has minimal impact on the energy consumption and latency.

However, all of the above studies [11], [12], [14]–[16], [20] are limited to proposing security protocols to encrypt MAVLink and evaluating their performance using a simulation setup. None of the above works has implemented the security mechanisms into a real drone and evaluated its performance. The feasibility and safety of deploying their designs in real-world scenarios remain uncertain since their assessments were limited to simulation setups. In contrast, in this paper, we implement the AES-CTR encryption algorithm on a real drone testbed with the Ardupilot Pixhawk Cube Orange Plus flight controller and evaluate its performance.

In [13], the communication between the GCS and the UAV was secured by implementing the AES scheme using a drone prototype built upon the Next-Generation Universal Aerial Video Platform (NG-UAVP). This implementation involved integrating the UAV prototype with a Field-Programmable Gate Array (FPGA), which was used to implement AES. The primary focus of the authors was on securing payload data transmitted by the drone, such as video and image data. However, they observed that the additional weight due to the extra hardware adversely impacted the performance of the drone. In contrast, our implementation of AES-CTR does not require any extra hardware and hence does not have any adverse impact on the performance of the drone.

III. BACKGROUND

A. MAVLink Protocol

MAVLink is a lightweight communication protocol specifically designed for the exchange of information between a UAV and a GCS [6]. Developed as an open-source protocol, MAVLink enables bi-directional command and control and/ or data exchange between a UAV and a GCS in real-time [21]. Its design emphasizes efficiency, making it suitable for communication in resource-constrained environments commonly found in small UAV and micro air vehicle systems [22]. MAVLink has two versions: MAVLink 1.0, launched in 2009, and the current MAVLink 2.0, introduced in 2017, which is backward compatible [8]. MAVLink messages consist of command and control messages from the GCS to the UAV, and status information messages (e.g., location and system status messages) from the UAV to the GCS. Fig. 1 shows the message format of MAVLink. Each message includes a header with auxiliary message information and a payload with data [8]. Message types are identified by Message IDs, with payloads containing relevant data. The payload size varies depending on what parameters are communicated, with a maximum length of 255 bytes [20]. The checksum ensures message integrity during transmission and the signature ensures authenticity [6]. Notably, the heartbeat message, sent periodically from the drone to the GCS, is crucial for indicating the drone's status and the connection status [23].

STX	LEN	INC FLAGS	CMP FLAGS	SEQ	SYS ID	COMP ID	MSG ID	PAYLOAD	CHECKSUM	SIGNATURE

Fig. 1. The figure shows the MAVLink V2.0 packet format [8].

Despite the widespread use of the protocol, MAVLink lacks an inherent security mechanism for encryption [6]. Adding encryption to the MAVLink protocol is required to achieve complete security while using the MAVLink protocol for communication.

B. Security Issues of MAVLink

The MAVLink protocol was developed with a primary focus on optimizing performance and addressing resource constraints, which inadvertently led to the presence of several security loopholes. Table I depicts the current state of MAVLink security. It can be seen from the table that encryption is not supported by MAVLink. Numerous papers discussing security and privacy issues in UAVs have highlighted various concerns [3], [24]. MAVLink 2.0 is identified as susceptible to several of these attacks, including flooding, packet injection [5], Denial-of-Service (DoS), and eavesdropping [9].

TABLE I
THE TABLE PROVIDES AN OVERVIEW OF MAVLINK'S SECURITY SUPPORT.

Security Objective	Security Mechanism	MAVLink Support
Confidentiality	Encryption	Not supported
Integrity	Hashing	Supported
Authenticity	Signature	Supported

Furthermore, researchers have categorized security attacks on MAVLink into four classes: interception (attacks compromising data confidentiality), modification (attacks compromising data integrity), interruption (attacks compromising data availability), and fabrication (attacks on authenticity) [6].

An attacker can perform an eavesdropping attack on the communication link between the GCS and the UAV by intercepting data such as live video feeds, sensor readings, GPS data, telemetry feeds, and commands communicated between the GCS and the UAV. Since the MAVLink protocol does not encrypt this data, the attacker can eavesdrop on the exchanged information.

The MAVLink protocol can be secured against the eavesdropping attack by adding encryption to it. In this paper, we integrate AES-CTR encryption into the MAVLink protocol.

C. Advanced Encryption Standard (AES)

AES provides a secure and efficient method for symmetric key encryption. It was established by the U.S. National Institute of Standards and Technology (NIST) in 2001 and has since become a globally accepted encryption standard [25]. AES employs a symmetric key algorithm, utilizing the same secret key for both encryption and decryption processes [26]. AES supports key sizes of 128, 192, and 256 bits. The flexibility in key lengths allows for different levels of security. Operating on fixed-size blocks of 128 bits, AES utilizes a Substitution-Permutation Network (SPN) structure. This involves a series of mathematical operations, including substitution, permutation, mixing, and key addition, to transform the input data into the encrypted output [25]. The number of rounds in the encryption process varies with the key size: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. This iterative approach enhances security by increasing the complexity of the encryption process [27].

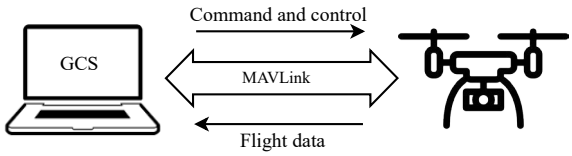


Fig. 2. The figure shows communication between a UAV and a GCS using MAVLink.

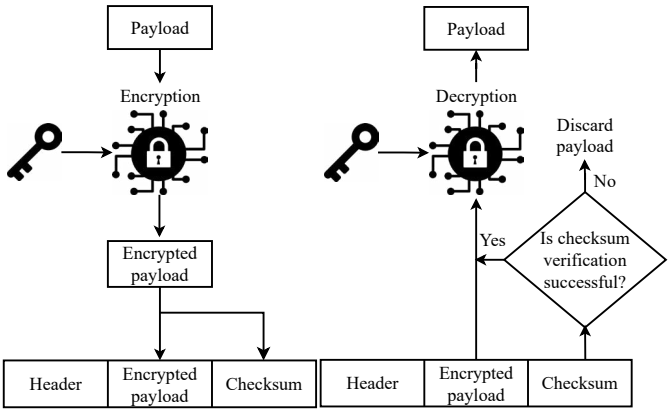
AES encryption offers various modes of operation, including Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR), each with unique characteristics and suitability for different applications [28]. The CTR mode in AES encryption transforms it into a stream cipher, utilizing a counter value for encryption. This mode allows for parallel encryption and decryption, making it ideal for high-performance applications [28]. Unlike block cipher modes, which require ciphertext lengths in multiples of the block size (16 bytes for AES-128), CTR mode ensures that the length of the ciphertext aligns with the length of the plaintext, making it suitable for MAVLink encryption within its payload length limit of 255 bytes.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

This study focuses on a communication link employing the MAVLink protocol between a UAV and a GCS (see Fig. 2). Key components of the system include the UAV, featuring a MAVLink-enabled flight controller, and the GCS software responsible for UAV monitoring and control. The flight controller also known as autopilot uses different sensors such as an inertial measurement unit (IMU), a barometer, and a GPS to control the UAV. Communication takes place over a wireless link, which commonly utilizes a radio frequency (RF) link or the Wi-Fi protocol. MAVLink provides bidirectional communication between a UAV and a GCS. Commands from the GCS are sent to the UAV and the UAV sends back flight details to the GCS. Both the UAV and the GCS exchange heartbeat messages within specific intervals to keep alive the communication. This information is encapsulated in a MAVLink frame, whose format is shown in Fig. 1. Notably, the MAVLink protocol utilized in this communication lacks encryption, and transmits even mission-critical information without protecting it via encryption. Our objective is to design and implement a secure communication framework for UAV-GCS interactions, by integrating AES-CTR encryption within the MAVLink protocol, while minimizing the impact on performance. We seek to evaluate the feasibility of the system by testing the solution on a drone testbed.

V. IMPLEMENTATION

In this section, we outline our implementation of encryption mechanisms in the MAVLink protocol. When integrating encryption into MAVLink, it is crucial to consider its impact on



(a) MAVLink payload encryption (b) MAVLink payload decryption

Fig. 3. The figure shows the integration of encryption and decryption into MAVLink.

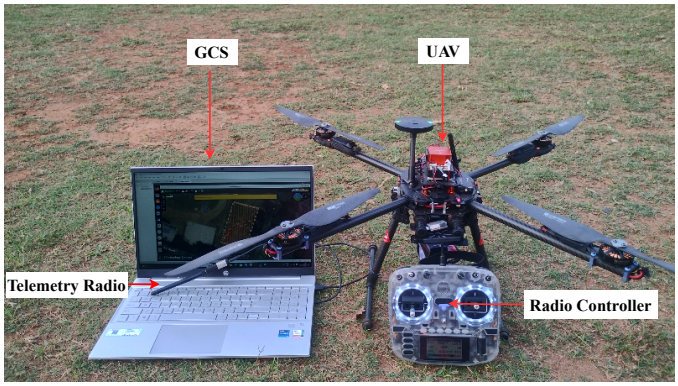


Fig. 4. The figure shows our drone testbed experimental setup.

both security and protocol functionality. Encrypting the header would hinder the recipient's ability to recognize the message type [6]. Therefore, our design only encrypts the payload while keeping the header unencrypted, ensuring seamless message recognition (see Fig. 3a). Next, consider MAVLink's checksum mechanism, which verifies message integrity. In our design, encryption is applied before checksum calculation, ensuring that the payload's confidentiality is safeguarded (see Fig. 3a). Subsequently, decryption takes place after checksum verification, guaranteeing the integrity of the message during its transmission and reception (see Fig. 3b). This approach ensures that our encryption-decryption process does not interfere with the integrity and functionality of MAVLink communication.

Our implementation involves modifying the MAVLink modules within both the GCS and UAV flight controller software. Our drone's flight controller system relies on Ardupilot [18] as the autopilot and QGroundController (QGC) [29] as the GCS. Ardupilot and QGC use the MAVLink libraries generated for the C language. Within the MAVLink library, the $mavlink\_helpers.h$ file handles the reception, decoding, transmission, and encoding of MAVLink messages. In our implementation, this file is enhanced to integrate encryption support for packets before transmission and decryption for received payloads during message parsing. The encryption key is hard-coded in this file.

Additionally, we made modifications to the $APMFirmwarePlugin.cc$ file in QGC to ensure proper firmware setup. This file acts as the interface between the QGC and Ardupilot firmware, ensuring seamless communication. It validates and re-encodes MAVLink messages required for the firmware setup before transmission and after reception to maintain compatibility across the MAVLink message versions exchanged between them. However, this reconstruction process internally encrypts the message, necessitating additional decryption to counteract this effect.

VI. EXPERIMENTAL SETUP

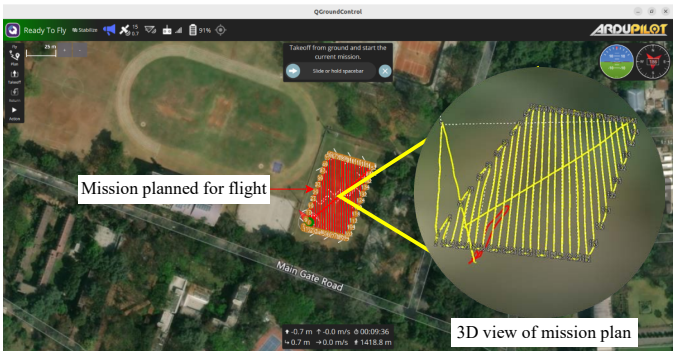
Our experimental drone setup for performance evaluation comprised a quadcopter with a Pixhawk Cube Orange Plus flight controller board [17], a Radiomaster Boxer Radio Controller [30], a Holybro-SiK telemetry radio module [31], and a laptop running QGroundController (QGC) [29] as the GCS. Additional components included a ublox GPS with antenna [32], radio modules [31], a LiPo 6s 10000 mAh battery [33], and four motors [34] with electronic speed control (ESC) [35]. To minimize the complexity, no external sensors were integrated into the drone. Fig. 4 shows the essential components of our testbed. The Cube Orange Plus flight controller utilized in the drone is an advanced open-source autopilot powered by 400MHz Cortex M7 and 200 MHz Cortex M4 CPUs [17].

Ardupilot served as the software controlling the Pixhawk Cube Orange Plus flight controller. To enhance security, modifications were made to the MAVLink code within the Ardupilot framework to support encryption, resulting in custom firmware tailored for the Ardupilot Cube Orange Plus flight controller. A custom QGC application image with encryption capabilities was developed for deployment on Linux 22.04 installed in the laptop. Communication between the laptop and the drone was established via a Holybro-SiK telemetry radio module.

The experimental setup also included a Wireshark network traffic analyzer [36] running on the same laptop as the GCS to capture and analyze MAVLink packets. Lua scripts [37] to parse the MAVLink messages were generated using the MAVGen tool [38]. This script was integrated as a plugin into Wireshark, enabling the parsing of MAVLink messages directly within the Wireshark interface.

VII. PERFORMANCE EVALUATION

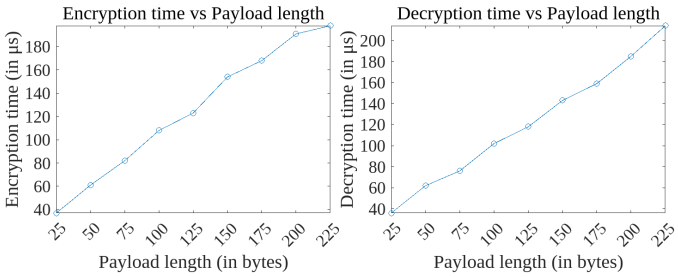
In this section, we evaluate the system's performance using metrics such as the encryption and decryption times, available memory, and CPU load. The Ardupilot firmware on the drone was modified twice for testing: first, the standard version was used, and then it was replaced with the version incorporating encryption. With both versions, the same mission plan was executed separately. The flight test used the mission plan shown in Fig. 5a: in particular, the drone traversed the trajectory shown in yellow in Fig. 5a in a playground within the IIT Bombay campus. Fig. 5b depicts the drone executing the mission. We conducted measurements across a range of payload sizes, from 25 bytes to 225 bytes, with increments of



(a) Mission plan used for the flight test

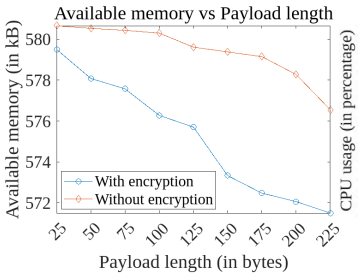
(b) Drone executing the mission

Fig. 5. The figure shows the flight test of the drone with and without encryption using a planned mission.

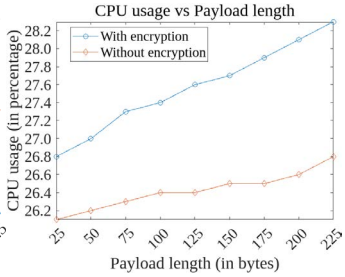


(a) Encryption time

(b) Decryption time



(c) Available memory in the system



(d) CPU load of the system

Fig. 6. Figs. (a) and (b) show the encryption time and decryption time, respectively, with varying payload sizes. Figs. (c) and (d) show the available memory and CPU load, respectively, with and without encryption for varying payload sizes.

25 bytes each. These adjustments were implemented by modifying the length of the $HEARTBEAT$ message as required. All measurements, viz., those of encrytion and decryption times, available memory, and CPU load, were conducted within Ardupilot (drone). The data was recorded using the dataflash log feature within Ardupilot. The experiments that were conducted and the corresponding results are described below.

Encryption and decryption times: Encryption and decryption times reflect the efficiency and speed of cryptographic operations within a system. The measurement process involved sending $HEARTBEAT$ messages with varying payload lengths, while the drone was in flight, from the QGC (GCS) to Ardupilot (drone) for decryption time evaluation and from Ardupilot to QGC for encryption time evaluation. A timer was initiated before invoking the AES-CTR encryption or decryption function, and the time difference upon function completion was logged into Ardupilot's dataflash log. Fig. 6a (respectively, Fig. 6b) shows the encryption time (respectively, decryption time) for varying MAVLink payload sizes. The average encryption time (respectively, decryption time) is around $124.6\ \mu$s (respectively, $121.6\ \mu$s). These measurements suggest that encryption and decryption are executed in short durations, indicating highly efficient encryption and decryption performance.

Memory consumption: The available memory reflects the remaining (free) RAM for executing tasks and storing data. In drone systems, where storage resources are often limited, monitoring the available memory helps to check whether there is enough free space when encryption and decryption processes run, so that there is no memory exhaustion or system instability. To understand the effect of encryption on memory consumption, the average available memory for each payload length with and without encryption were observed from the Ardupilot cube dataflash logs. Fig. 6c shows measurements of the available memory recorded during the drone's flight tests, comparing the cases with and without encryption. As expected, the available memory without encryption exceeds that with encryption. However, on average, the difference is only 4.27 kB, and there is a percentage decrease of only $0.73\,\%$, indicating that encryption does not significantly burden the memory.

CPU load: Encryption introduces a computational overhead, which can increase the CPU load. We quantified the impact of encryption on the drone's CPU load by measuring the average CPU load during flight tests with and without encryption. The CPU load for each payload length with and without encryption was observed from the Ardupilot cube dataflash logs. Fig. 6d shows the CPU utilization of the drone system, comparing the scenarios with and without encryption. It demonstrates that encryption only increases the CPU load by a small amount of around $4.33\,\%$ on average.

Performance comparison: Table II provides a comparison between the results from our drone testbed and those from various simulation-based studies. Due to the absence of data on encryption and decryption times in the simulation-based studies, the comparison focuses solely on memory consumption and CPU usage. In terms of memory consumption, the flight test shows an increase of $0.86\,\%$ when encryption is used, which is notably higher than for the simulation results for various algorithms. The simulation result for AES-CTR from [14] shows an increase in memory consumption that is as low as $0.054\,\%$. This indicates that real-world conditions may impose higher memory demands than those expected in simulations. In terms of CPU usage, the flight test shows a

$4.33\,\%$ increase when encryption is used, which is lower than for all the simulation results, except that in [11], which reports a $3.63\,\%$ increase. This difference is due to the fact that the real-world implementation benefits from hardware-accelerated AES encryption, resulting in lower CPU usage. Overall, while the increase in memory consumption due to encryption in real-world tests is higher than in simulations, the increase in CPU usage is lower, reflecting a discrepancy between simulated and actual operating environments. This shows that it is important to evaluate the performance of encryption algorithms using a practical drone testbed, as in our study; simulation-based studies do not suffice.

TABLE II
PERFORMANCE COMPARISON WITH SIMULATION BASED APPROACHES

Method Name	Percentage Increase in Memory Consumption	Percentage Increase in CPU Usage
Simulation AES-CTR [14]	0.054	9.09
Simulation ChaCha20 [15]	0.043	9.09
Simulation DMAV [15]	0.13	18.18
Simulation Navid et al. [11]	0.0625	3.63
Drone Testbed	0.86	4.33

VIII. CONCLUSIONS AND FUTURE WORK

We demonstrated the integration of AES encryption with CTR mode into the MAVLink protocol on a drone testbed. Our results show that the integration of AES-CTR encryption into MAVLink causes only minimal overheads in the system performance. In particular, the encryption and decryption times are small and there is only a slight increase in the memory consumption and CPU load due to encryption. Our validation of the AES with CTR mode encryption scheme on a drone testbed demonstrates that it is a secure and practical solution for real-world applications. A direction for future research is to integrate other encryption schemes, e.g., ChaCha20, into MAVLink using a drone testbed and compare their performance with that of our AES-CTR implementation.

REFERENCES

[1] G. Singhal, B. Bansod, and L. Mathew, "Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review," *Preprints*, November 2018. [Online]. Available: https://doi.org/10.20944/preprints201811.0601.v1

[2] F. Ahmed, J. C. Mohanta, A. Keshari, and P. S. Yadav, "Recent Advances in Unmanned Aerial Vehicles: A Review," *Arabian Journal for Science and Engineering*, vol. 47, no. 7, pp. 7963–7984, Jul 2022.

[3] Y. Mekdad, A. Aris, L. Babun, A. E. Fergougui, M. Conti, R. Lazzeretti, and A. S. Uluagac, "A Survey on Security and Privacy Issues of UAVs," *Computer Networks*, vol. 224, p. 109626, 2023.

[4] S. Dahiya and M. Garg, "Unmanned Aerial Vehicles: Vulnerability to Cyber Attacks," in *Proceedings of UASG 2019*. Cham: Springer International Publishing, 2020, pp. 201–211.

[5] 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.

[6] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, 2019.

[7] M. Ficco, R. Palmiero, M. Rak, and D. Granata, "MAVLink Protocol for Unmanned Aerial Vehicle: Vulnerabilities Analysis," in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2022, pp. 1–6.

[8] "MAVLink Micro Air Vehicle Protocol." [Online]. Available: https://github.com/mavlink

[9] 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 *IAI*, 2021, pp. 1–6.

[10] "National Vulnerability Database - CVE-2020-10282 Detail," March 2020. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2020-10282

[11] N. Khan, N. Jhanjhi, S. Brohi, and A. Almazroi, "A Secure Communication Protocol for Unmanned Aerial Vehicles," *Computers, Materials and Continua*, vol. 70, pp. 601–618, 01 2021.

[12] G. Emad Kassim and S. Hassan Hashem, "DMAV: Enhanced MAV Link Protocol Using Dynamic DNA Coding for Unmanned Aerial Vehicles," *iJOE*, vol. 18, no. 11, p. pp. 4–16, Aug. 2022.

[13] A. Shoufan, H. AlNoon, and J. Baek, "Secure Communication in Civil Drones," in *Information Systems Security and Privacy*. Cham: Springer International Publishing, 2015, pp. 177–195.

[14] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui, and T. Abbes, "MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems," in *2019 15th IWCMC*, 2019, pp. 621–628.

[15] N. Sabuwala and R. D. Daruwala, "Securing Unmanned Aerial Vehicles by Encrypting MAVLink Protocol," in *2022 IEEE Bombay Section Signature Conference (IBSSC)*, 2022, pp. 1–6.

[16] R. Pizzolante, A. Castiglione, F. Palmieri, A. Passaro, R. Zaccagnino, and S. La Vecchia, "Improving Drone Security in Smart Cities via Lightweight Cryptography," in *Computational Science and Its Applications – ICCSA 2023 Workshops*. Cham: Springer Nature Switzerland, 2023, pp. 99–115.

[17] "The Cube Orange/+ With ADSB-In Overview." [Online]. Available: https://ardupilot.org/copter/docs/common-thecubeorange-overview.html#system-features

[18] "Ardupilot Documentation." [Online]. Available: https://ardupilot.org/ardupilot/

[19] "Open Source Autopilot For Drone Developers." [Online]. Available: https://px4.io/

[20] L. Chaari, S. Chabanu, and J. Rezugi, "MAV-DTLS toward Security Enhancement of the UAV-GCS Communication," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–5.

[21] "The Dronecode Foundation - We are Setting the Standards in the Drone Industry with Open-Source." [Online]. Available: https://dronecode.org/

[22] L. Reichstein, S. Schopferer, and F. Jünger, "A Comparison of Command and Control Communication Protocols for Unmanned Aircraft: STANAG 4586 vs. MAVLink," in *ICUAS*, 2022, pp. 1283–1292.

[23] S. Atoev, K.-R. Kwon, S.-H. Lee, and K.-S. Moon, "Data Analysis of the MAVLink Communication Protocol," in *ICISCT*, 2017, pp. 1–3.

[24] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security Analysis of Drones Systems: Attacks, Limitations, and Recommendations," *Internet of Things*, vol. 11, p. 100218, 2020.

[25] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced Encryption Standard (AES)," 2001-11-26 2001.

[26] A. M. Abdullah *et al.*, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," *Cryptography and Network Security*, vol. 16, no. 1, p. 11, 2017.

[27] M. Vaidehi and B. J. Rabi, "Design and Analysis of AES-CBC Mode for High Security Applications," in *2nd ICCTET*, 2014, pp. 499–502.

[28] S. Almuhammadi and I. Al Hejri, "A Comparative Analysis of AES Common Modes of Operation," in *CCECE*, 2017, pp. 1–4.

[29] "QGroundControl User Guide." [Online]. Available: https://docs.qgroundcontrol.com/master/en/qgc-user-guide/index.html

[30] "Radiomaster - Boxer Radio Transparent Version (ELRS / M2)." [Online]. Available: https://www.radiomasterrc.com/collections/boxer-1/products/boxer-radio-transparent-version

[31] "SiK Telemetry Radio V3." [Online]. Available: https://docs.holybro.com/radio/sik-telemetry-radio-v3

[32] "Cubepilot - Here 3 Manual." [Online]. Available: https://docs.cubepilot.org/user-guides/here-3/here-3-manual

[33] "Tattu 6S 10000mAh 30C 22.2V Lipo Battery Pack With EC5 Plug For UAV Drone." [Online]. Available: https://genstattu.com/ta-30c-10000-6s1p-ec5.html

[34] "TMOTOR Antigravity MN4006 KV380 Brushless Multirotor Motor." [Online]. Available: https://shop.tmotor.com/products/mn4006-brushless-multirotor-motor

[35] "TMOTOR AIR 40A 6S ESC for Multirotor Drones." [Online]. Available: https://shop.tmotor.com/products/air-40a-6s

[36] J. Bullock and J. T. Parker, *Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework*. John Wiley & Sons, 2017.

[37] "Debugging with Wireshark." [Online]. Available: https://mavlink.io/en/guide/wireshark.html

[38] "Generating MAVLink libraries." [Online]. Available: https://mavlink.io/en/getting_started/generate_libraries.html