An Approach Utilizing a Random Keystream Generator to Enhance the Security of Unmanned Aerial Vehicles

Noshin A Sabuwala
Electrical Department
Veermata Jijabai Technological Institute
Mumbai, India
nasabuwala_p21@el.vjti.ac.in

Rohin D Daruwala
Electrical Department
Veermata Jijabai Technological Institute
Mumbai, India
rddaruwala@el.vjti.ac.in

***Abstract*—The Unmanned Aerial Vehicles' (UAVs) capability to respond to people's needs accounts for their pervasiveness. UAVs with extended functions and capabilities when supplied with communication equipment can be deployed to appropriate places in the field to supplement the networks operate more efficiently and in vital missions such as infrastructure monitoring operations. To be effective, an UAV must interact securely with its network's entities, such as ground control stations, other UAVs, air traffic control systems, and navigation satellite systems. UAVs are exposed to a dangerous and costly world of cyber dangers as a result of cyber technology and connections. The UAV and the Ground Control Station (GCS) exchange information using communication lines, which are vulnerable to cyber attacks. The Micro Air Vehicle Link (MAVLink) protocol is a widely used lightweight communication protocol for enabling communication between UAVs and GCSs. It carries information about the UAV's condition as well as commands for control from the GCS. Although widely used, the MAVLink protocol lacks sufficient security measures and is susceptible to various types of attacks. In the current study, a new stream cipher method with low duty cycles is proposed for protecting data in UAVs and is compared with existing security algorithms on basis of various factors. The research findings' show that by including the suggested method into MAVLink, it is possible to maintain both message secrecy and battery life for a resource-constrained UAV.**

***Index Terms*—Unmanned Aerial Vehicles (UAVs), Communication Security, MAVLink, Stream Cipher, Internet of Things(IoT)**

I. Introduction

Unmanned vehicles are autonomous, easily programmable technologies that can complete tasks with or without human aid. UAVs often interact with a GCS wirelessly, which controls their actions and monitors their status. They however are susceptible to a variety of cyber threats including message manipulation, injection, GPS spoofing, and jamming. Security is of the utmost significance in such networks as critical information may be transferred between various network entities. MAVLink is a widely adopted message serialization protocol for UAVs that was introduced in 2009 by Lorenz Meier under the LGPL license. MAVLink enables two-way interaction between the GCS and UAV. Additionally, by double-checking the header's checksum, it guarantees message integrity and reliability. MAVLink protocol has weaknesses and is susceptible to numerous attacks, including denial of service, spoofing, and message forging attacks, despite its widespread use [1]. The primary reason for these problems is that the protocol doesn't use any encryption or security measures. In this work, we aim at enhancing secure MAVLink communication between UAVs and GCSs. As a result, hostile attacks can be reduced. UAVs have historically been employed primarily in defense operations, although they are increasingly widely used for scientific, commercial, and recreational purposes. Corporations like Amazon and Google intend to deploy UAVs for the delivery of products and services [2]. Potential dangers and security issues also start to emerge with the development in UAV utilisation. UAVs may be simpler to hack because of their rapid and simple setup requirements, frequent usage of unencrypted communication and data transfer, and many accessible ports. A lot of the security processes and technologies are now being designed without performing an adequate threat analysis. Utilizing insecure devices runs the risk of causing the unauthorised disclosure of sensitive data.

The aim of this work is to recommend a lightweight, low-power stream cipher scheme for protecting data transferred between UAVs and GCS. In order to produce keys, we develop a new keystream generator that takes into account the need for a higher security level and less difficult procedures. The generated keys are used for encrypting the information. The primary contributions of this study are as follows: We we provide a novel strategy for maintaining the privacy of information passed between GCS and UAV. In order to demonstrate the viability of the method and to apply the protective measures in software in the loop simulator with MAVLink, we then designed a case study. We then examine our suggestion's performance to evaluate the technique's effectiveness using five benchmark tests. The remainder of this article is structured as follows: Section II contains a summary of the literature review. The proposed keystream cipher is described in Section III. The implementation and comprehensive simulated experimental findings are presented in Section IV. In Section V, we present concluding remarks.

II. Literature Review

The literature review of how UAVs can be easily intercepted, manipulated and interrupted [3] and their countermeasures provided by various researchers is summarised below.

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

An attacker can take advantage of the flaws in each component of a multi-vector attack, but the overall result might be disastrous. Onboard flight controllers, ground control systems, sensors, actuators, wireless data links, and routing infrastructure can all pose security risks to UAVs. The three types of attacks can be grouped according to the vulnerability: attack on wireless, hardware, and sensors [4]. An attacker can directly access the UAV autopilot components through a hardware attack. While a sensor spoofing attack uses the onboard GPS channels to inject or transfer fake data, a wireless attack uses one of the wireless communication channels to carry out the attack. While the UAV is in use, an attacker can conduct attacks from a great distance. Intentionally disrupting a communication link while using a UAV to film an Australian triathlon is one example of a UAV attack [5]. Another contentious episode had Iranian soldiers claiming to have an RQ 170 Sentinel. According to one explanation, Iranian soldiers disrupted GPS and UAV satellite communications, making it simple to attack the GPS system using sensor spoofing [6]. On the other side, the sensors are constrained by storage and power issues, which reduce the effectiveness of the transmission. Use of portable encryption technology is one potent way to overcome the aforementioned restrictions and achieve excellent security. In recent times, Panagiotou et al. [7] recommended the use of stream ciphers as a simple cryptography method for private information in IoT devices. In work by Salami et al. [8], by recommending a light-weight encryption method based on identity that does not need a certificate for protecting communicating information between the homeowner and smart items in the house, resource-constrained smart home equipment was relieved of its security issues. Recently, Panagiotou et al. [7] developed symmetric cryptography for texts, photos, and electronic data applications in Internet of Things systems, based on the Advanced Encryption Standard (AES). Hammi et al. [9] suggested a light-weight IoT authentication system based on elliptic curve cryptography by adopting OTP as an authentication method, which prevents the reuse of passwords by generating a new one for each authentication session. Lately, Kponyo et al. [10] have suggested resource-constrained IoT devices a host-based and light-weight DoS anomaly detective and defence system. Their strategy focused on addressing DoS attacks on Internet of Things systems. The MAVLink protocol is vulnerable to a variety of attacks, such as message injection, in which a malicious party inserts untrusted MAVLink messages into the channel of communication between the UAV and the Ground Control Station (GCS). The behaviour of the UAV can be changed by an attacker by introducing malicious messages. Message manipulation entails changing the content of genuine MAVLink communications, much like message injection. An attacker can trick the UAV or GCS into acting incorrectly and cause possibly harmful or unintended consequences by altering crucial parameters or orders. Replay attacks include intercepting MAVLink communications sent back and forth between the GCS and the UAV and then playing them back to trick the system. The attacker can cause repetition or system disturbance by forcing the UAV to carry out the same operations repeatedly by repeating previously captured communications. The unintentional interception and observation of MAVLink signals constitute eavesdropping. An attacker can obtain sensitive information, such as mission specifics, telemetry data, or system vulnerabilities, by intercepting the transferred communications and using it for later exploitation. For MAVLink's security, various solutions have been created. For message authentication and encryption of MAVLink data between UAV and GCS, the authors employ the Caesar cypher [11]. One of the study's weaknesses is the absence of its outcomes. Another drawback of their work is that the secret Key is transmitted in plain text. Moreover, no empirical evaluation of the study has been done. In [12], the author proposed employing cryptographic encryption for authentication to ensure data integrity. Yet, these two investigations are only hypotheses. In [13], the authors strengthened the security by including a further layer, which shields the whole packet. To protect the MAVLink communication system, more investigation has been conducted. Unfortunately, the majority of the study is still in its development or consists of of hypotheses. In [14], the authors provided a MAVSec mechanism to secure MAVLink communication. They compared the four encryption methods RC4, ChaCha20, AES-CBC, and AES-CTR. Their research indicates that ChaCha20 looks to be outperforming its rivals. However, with their method, they just encrypt the payload messages. The packet is identical for the rest of it. In order to enable safe MAVLink protocol connection, we integrate the encryption method into the source code of the UAV in this study.

III. Proposed Keystream Cipher

In this section, we propose a novel keystream cipher algorithm for securing communications. We use this for our experimentation and prove its randomness using five benchmark tests. In order to create a novel keystream for raising security, the basics of the ChaCha20 have been researched. UAV communications are encrypted using the generating keys. The new strategy consists of 10 rounds which makes a Lightweight Stream Cipher, described in Algorithm 1 and is summarized into:

- ChaCha20 changes the rotation method (16, 12, 8, and 7) to a variable constant from a fixed constant based on a random value (y_0, y_1, y_2, and y_3), in each round, depicted in Algorithm 2.
- There has been a change in the application order of the QRF (for inputs updating) in the diagonals form following the columns form to zigzag form followed by alternate form depicted in Figs. 1 and 2. This new updating process order causes a greater dissemination of inputs, which raises the complexity of defence against attacks.

For encryption, 512 bits of generated keystream (Algorithm 1) is XORed with the UAV payload.

IV. Experimental Results

In this section, we will first discuss the implementation of various algorithms in the system, then the standards for evaluating the level of randomness of the keystream generator's generated binary series[27] for the proposed algorithm, and then we compare our alogrithm to existing algorithm

Algorithm 1 Proposed Keystream Cipher Algorithm

```
Require: $Key \in (0,1)^{256}$, $Nonce \in (0,1)^{96}$, $Count \in (0,1)^{32}$, $PlainText \in (0,1)^*$
Ensure: $CipherText = ProposedKeystreamCipher(Key, Nonce, Count, PlainText)$
$I \leftarrow Init(Key, Nonce, Count)$
for $a \leftarrow 1$ to $\left\lceil \frac{\text{length}(PlainText)}{512} \right\rceil$ do
    $O \leftarrow I$
    for $b \leftarrow 1$ to 10 do
        $O_{[0,1,4,8]} \leftarrow QR(O_{[0,1,4,8]})$
        $O_{[5,2,3,6]} \leftarrow QR(O_{[5,2,3,6]})$
        $O_{[9,12,13,10]} \leftarrow QR(O_{[9,12,13,10]})$
        $O_{[7,11,14,15]} \leftarrow QR(O_{[7,11,14,15]})$
        $O_{[0,4,1,5]} \leftarrow QR(O_{[0,4,1,5]})$
        $O_{[8,12,9,13]} \leftarrow QR(O_{[8,12,9,13]})$
        $O_{[2,6,3,7]} \leftarrow QR(O_{[2,6,3,7]})$
        $O_{[10,14,11,15]} \leftarrow QR(O_{[10,14,11,15]})$
    end for
    $Sl \leftarrow Serial(O + I)$
    for $c \leftarrow 1$ to 512 do
        $CipherText_{[512(a-1)+(c-1)]} \leftarrow PlainText_{[512(a-1)+(c-1)]} \oplus S_{[c-1]}$
    end for
    $I_{[12]} \leftarrow I_{[12]} + 1$
end for
return $CipherText$
```

Algorithm 2 Proposed Keystream Cipher Quarter Function

```
Require: Four 32-bit integers ($p$, $q$, $r$, $s$)
Ensure: Updated Four 32-bit integers ($p$, $q$, $r$, $s$)
Let $y_0$ be the first 4 bits of $r$
Let $y_1$ be the first 4 bits of $p$
Let $y_2$ be the first 4 bits of $q$
Let $y_3$ be the first 4 bits of $s$
$p \leftarrow p + q$; $s \leftarrow (s \oplus p) <<< 16$
$r \leftarrow r + s$; $q \leftarrow (q \oplus r) <<< 12$
$p \leftarrow p + q$; $s \leftarrow (s \oplus p) <<< 8$
$r \leftarrow r + s$; $q \leftarrow (q \oplus r) <<< 7$
```

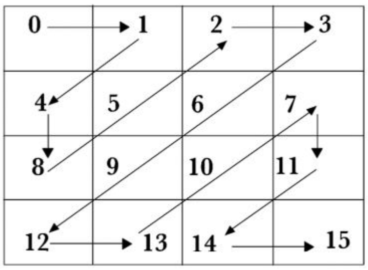


Fig. 1. Zigzag Form

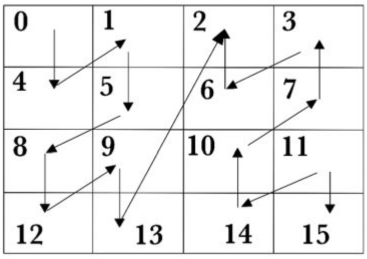


Fig. 2. Alternate Form

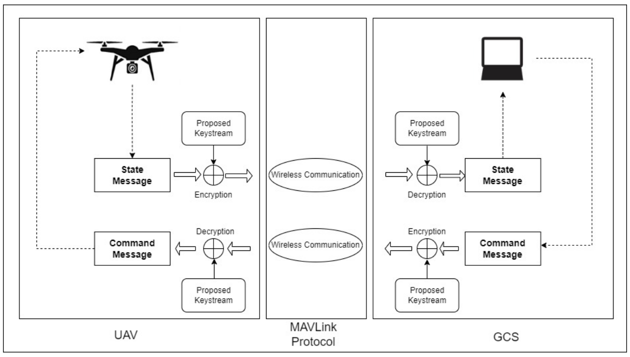
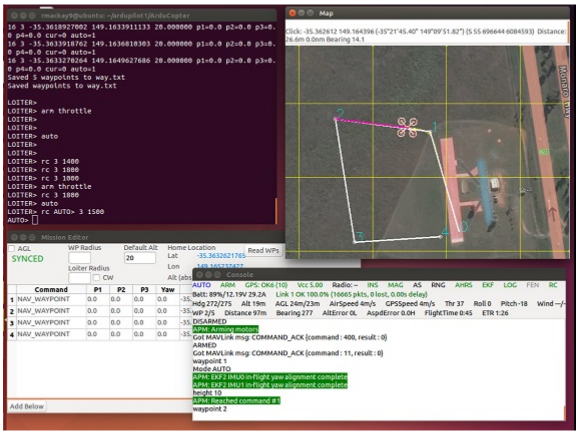


Fig. 3. Block Diagram of Implemented System

integrated MAVLink and the original MAVLink for metrics. The block diagram of implemented system presented in 3.

A. Setup

Ardupilot, a Software-In-The-Loop (SITL), which uses the same autopilot and MAVLink communication protocol as a real UAV, is used with a model UAV as the testbed for the experiment. A virtual UAV's use directly generalises to a real UAV's use. Without any hardware, we can fly a helicopter, a plane, or a rover using the SITL simulator. We have put together the Ardupilot source code in order to add all the keystream cipher encryption techniques one by one to the communication stream delivered between the GCS and UAV's. In addition, we used a C++-based GCS application, Lorenz Meier's open-source QGroundControl ground station. In order to enable secure communication between the Ardupilot and the QGroundControl, we also added the keystream cipher encryption techniques one by one to the QGroundControl. This allows it to decrypt the received cipher stream and extract the genuine MAVLink message. The GCS and the fictional UAV are connected through a free GCS application called MAVproxy. We have used the gazebo for simulations. A snapshot of the Ardupilot simulation (UAV) is presented in 4 and a snapshot of QGrounDControl simulation is presented in 5. To connect to the SITL, we used the UDP protocol with desirable port. The output is a keystream with good robustness. The proposed keystream cipher's time consumption in microseconds and attack difficulty are examined with those of the standard ChaCha versions (8), (12), and (20) [15]. The comparison shows that the suggested keystream cipher outperforms the common ChaCha of versions 8 and 12 with a very little time increase of 1 to 2 microsecond as can be seen in Fig 6.



```
rmackay9@ubuntu: ~/ardupilot1/ArduCopter
16 3 -35.3618927002 149.1633911133 20.000000 p1=0.0 p2=0.0 p3=0.
0 p4=0.0 cur=0 auto=1
16 3 -35.3633918762 149.1636810303 20.000000 p1=0.0 p2=0.0 p3=0.
0 p4=0.0 cur=0 auto=1
16 3 -35.3633278264 149.1649627686 20.000000 p1=0.0 p2=0.0 p3=0.
0 p4=0.0 cur=0 auto=1
Saved 5 waypoints to way.txt
Saved waypoints to way.txt

LOITER>
LOITER> arm throttle
LOITER>
LOITER>
LOITER> auto
LOITER>
LOITER>
LOITER> rc 3 1400
LOITER> rc 3 1800
LOITER> rc 3 1000
LOITER> arm throttle
LOITER> rc 3 1800
LOITER> auto
LOITER> rc AUTO> 3 1500
AUTO>
```

Map

Click: -35.362612 149.164396 (-35°21'45.40" 149°09'51.82") (S 55 696644 6084593) Distance: 26.6m 0.0m Bearing 14.1

Mission Editor

□ AGL SYNCED | WP Radius | Default Alt: 20 | Loiter Radius | □ CW | Home Location Lat -35.3632621765 Lon 149.165237427 Alt (abs) | Read WPs

	Command	P1	P2	P3	Yaw	
1	NAV_WAYPOINT	0.0	0.0	0.0	0.0	-35.
2	NAV_WAYPOINT	0.0	0.0	0.0	0.0	-35.
3	NAV_WAYPOINT	0.0	0.0	0.0	0.0	-35.
4	NAV_WAYPOINT	0.0	0.0	0.0	0.0	-35.

Add Below

Console

```
AUTO  ARM  GPS: OK6 (10)  Vcc 5.00  Radio: --  INS  MAG  AS  RNG  AHRS  EKF  LOG  FEN  RC
Batt: 89%/12.19V 29.2A   Link 1 OK 100.0% (16665 pkts, 0 lost, 0.00s delay)
Hdg 272/275  Alt 19m  AGL 24m/23m  AirSpeed 4m/s  GPSSpeed 4m/s  Thr 37  Roll 0  Pitch -18  Wind --/-
WP 2/5  Distance 97m  Bearing 277  AltError 0L  AspdError 0.0H  FlightTime 0:45  ETR 1:26
DISARMED
APM: Arming motors
Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
ARMED
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
waypoint 1
Mode AUTO
APM: EKF2 IMU0 in-flight yaw alignment complete
APM: EKF2 IMU1 in-flight yaw alignment complete
height 10
APM: Reached command #1
waypoint 2
```

Fig. 4. Snapshot of the Ardupilot Simulation

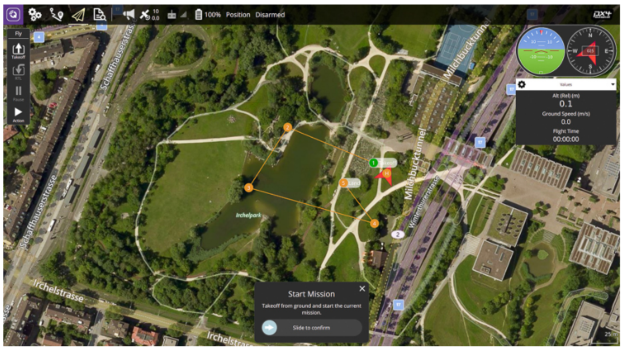


Fig. 5. Snapshot of the QGroundControl Simulation

B. Performance Evaluation

Using the five fundamental tests (benchmark tests), the keystream generator's randomness performance is assessed as illustrated in Table I. This test, which is an empirical test, only looks at the keystream generator's output sequences.

The model's outputs, which are keys with a simpler technique and a solid keystream of robustness, effectively outperformed the five benchmark tests, as shown in Table II.

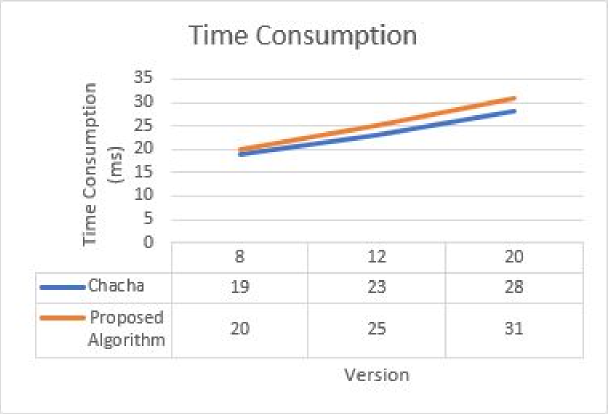


Fig. 6. Time Consumption Comparision

TABLE I
FIVE BENCHMARK TESTS EQUATIONS AND INFORMATION

Benchmark Tests	
Test Equation	Information on Test
$T1 = (M0 - M1)^2/M$	M0: No. of 0's in keystream M1: No. of 1's in keystream M: total size of keystream
$T2 = (4/M - 1)((M11)^2 + (M00)^2 + (M01)^2 + (M10)^2) - (2/M)(M1^2 + M0^2) + 1$	M11: number of 11's in keystream M00: No. of 00's in keystream M01: No. of 01's in keystream M10: No. of 10's in keystream
$P = \frac{M}{N}$, $\frac{M}{N} \geq (5 * 2^N)$ T3 = $(2^N/P)(\sum_{j=1}^{2^N} M_j^2) - P$	M_j: No. of appearance of the jth of length N
$P_j = \frac{M-j+3}{2^{j+2}}$ T4 = $(\sum_{j=1}^{N}((B_j - P_j)^2/P_j))+$ $(\sum_{j=1}^{N}((G_j - P_j)^2/P_j)))$	N: maximum j for which $P_j \geq 5$ B_j: No. of blocks (sequences of 1's) of length j in M G_j: No. of gabs(sequences of 0's) of length j in M
$A(k) = \sum_{j=0}^{M-k-1}(S_j + S_{+k})$ mod 2 T5 = 2 (A(k)-(M-k)/2)/ $\sqrt{M-k}$	$k : 1 \leq k \leq [m/2]$

TABLE II
BENCHMARK TEST RESULTS

Benchmark Tests Performance			
Five Benchmark Tests	Test value	Threshold	Result
T1 (Frequency Test)	0.08	3.841	Success
T2 (Serial Test)	0.865	5.991	Success
T3 (Poker Test)	13.133	14.067	Success
T4 (Runs Test)	5.216	9.487	Success
T5 (Autocorrelation)	1.563	1.96	Success

V. CONCLUSION

Increasing applications of UAVs has ushered in a new epoch of unmanned aerial vehicles in civilian as well as military sectors, with numerous advantages like economic and industrial benefits, owing to their easy-to-use, flexible, and autonomous nature, as well as energy and cost efficiency. Their usage, however, resulted in a slew of security, safety, and privacy concerns, which expressed themselves in the form of a slew of cyber attacks, threats, and difficulties, all of which are mentioned and discussed in this study. The vulnerability and security hazards of the MAVLink protocol are then discussed. A new keystream cipher is proposed in the current work. Results show that with very modest increase in time consumption, the security is substantially raised. We think the suggested keystream cipher is appropriate for the security of unmanned aerial vehicles (UAVs), which need strong security yet have little energy and little storage capacity. During performance testing, we showed that the suggested keystream technique can be utilised to improve MAVLink's security because it maintains message secrecy without sacrificing performance.

REFERENCES

[1] 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, aug 2018.

[2] C. Rani, H. Modares, R. Sriram, D. Mikulski, and F. L. Lewis, "Security of unmanned aerial vehicle systems against cyber-physical attacks:," *http://dx.doi.org/10.1177/1548512915617252*, vol. 13, pp. 331–342, 11 2015. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1548512915617252

[3] N. A. Sabuwala and R. D. Daruwala, “Drones: Architecture, vulnerabilities, attacks and countermeasures,” in *International Conference on Intelligent Vision and Computing*. Springer, 2022, pp. 220–232.

[4] A. Kim, B. Wampler, J. Goppert, I. Hwang, and H. Aldridge, “Cyber attack vulnerabilities analysis for unmanned aerial vehicles,” *AIAA Infotech at Aerospace Conference and Exhibit 2012*, 2012. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2012-2438

[5] “Triathlete injured by “hacked” camera drone — ars technica.” [Online]. Available: https://arstechnica.com/information-technology/2014/04/triathlete-injured-by-hacked-camera-drone/

[6] K. Hartmann and C. Steup, “The vulnerability of uavs to cyber attacks-an approach to the risk assessment,” 2013.

[7] P. Panagiotou, N. Sklavos, E. Darra, and I. D. Zaharakis, “Cryptographic system for data applications, in the context of internet of things,” *Microprocessors and Microsystems*, vol. 72, p. 102921, 2 2020.

[8] S. A. Salami, J. Baek, K. Salah, and E. Damiani, “Lightweight encryption for smart home,” *Proceedings - 2016 11th International Conference on Availability, Reliability and Security, ARES 2016*, pp. 382–388, 12 2016.

[9] B. Hammi, A. Fayad, R. Khatoun, S. Zeadally, and Y. Begriche, “A lightweight ecc-based authentication scheme for internet of things (iot),” *IEEE Syst J*, vol. 14, pp. 3440–3450, 9 2020.

[10] J. J. Kponyo, J. O. Agyemang, G. S. Klogo, and J. O. Boateng, “Lightweight and host-based denial of service (dos) detection and defense mechanism for resource-constrained iot devices,” *Internet Things J*, vol. 12, p. 100319, 12 2020.

[11] B. S. Rajatha, C. M. Ananda, and S. Nagaraj, “Authentication of MAV communication using Caesar Cipher cryptography,” *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings*, pp. 58–63, aug 2015.

[12] R. Altawy and A. M. Youssef, “Security, Privacy, and Safety Aspects of Civilian Drones,” *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 2, nov 2016.

[13] N. A. Khan, N. Z. Jhanjhi, S. N. Brohi, A. A. Almazroi, and A. A. Almazroi, “A Secure Communication Protocol for Unmanned Aerial Vehicles,” *Computers, Materials & Continua*, vol. 70, no. 1, pp. 601–618, sep 2021.

[14] A. Allouch, O. Cheikhrouhou, A. Koubaa, M. Khalgui, and T. Abbes, “MAVSec: Securing the MAVLink protocol for Ardupilot/PX4 unmanned aerial systems,” *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, pp. 621–628, jun 2019.

[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.