# Enhancing ChaCha Algorithm to Improve Security

Noshin Sabuwala
*Electrical Department*
*Veermata Jijabai Technological Institute*
Mumbai, India

Rohin D Daruwala
*Electrical Department*
*Veermata Jijabai Technological Institute*
Mumbai, India

***Abstract*—Unmanned Aerial Vehicles (UAVs) are employed in both military and civilian operations for mission-critical activities. Their potential for use in the commercial sector is growing quickly. UAV data should be encrypted because they are very sensitive. In order to give a high level of security, complex operation encryption techniques are developed. These mathematical operations bring forth issues with efficiency and power usage, though. One of these methods, the ChaCha cypher, recently gained notoriety after Google used it in a number of applications. In the current study, a new stream cypher method with low duty cycles is proposed for protecting data in UAVs. By strengthening resistance to cryptanalysis, the suggested method constitutes an enhanced version to the conventional ChaCha algorithm. The adjustment primarily affects the rotation process, which was converted to a variable constant from a fixed constant based on a random value. When compared to normal ChaCha, the cypher inputs also change, first taking the shape of columns, then diagonals, then zigzags, and finally alternate forms. The security results show that a brute-force attack cannot defeat our approach without 2512 probable keys.**

***Index Terms*—ChaCha, Stream cipher, UAV, Internet of Things.**

## I. Introduction

Together with the development and expansion of digital tools and technologies, the necessity for cyber security has increased. The objects we use on a daily basis are now becoming smarter and connected to the Internet of Things (IoT), a global computer network, software platforms, and communication channels. As a result, protecting digital data has grown to be a serious challenge. The dangers to the confidentiality, integrity, and availability of cyber physical systems (CPS) are not sufficiently addressed by traditional computer and network security techniques. They also don't address a cohesive strategy for surviving from hostile intimidations and recovering from attacks [1]. The vulnerability of CPS has been studied in recent years, notably for UAVs and Ground Control Stations(GCSs), but little work has been done on secure message transfer, communication, and trust establishment. Cyber physical systems (CPS) are independent systems that combine control, computation, and communication technologies [2]. The smart grid system, networks for distributing oil and gas, improved communication systems, unmanned aerial vehicles, and intelligent ground vehicles are only a few applications for CPS. UAVs are cyber-physical devices that can either fly autonomously using on-board computers in accordance with pre-programmed flight plans or under remote control from a ground control station. They are also intelligent systems that can connect with their controller, return payload data, and make decisions automatically in response to an event [3]. Control components, wireless and satellite communication links, sensors, and actuators make up a UAV's core components. UAVs are limited-resource devices. They have historically been employed primarily in defensive operations, although they are increasingly widely used for scientific, commercial, and recreational purposes. They are used as a major tool [4] for shippers, law enforcement agencies, farmers, aerial photographers, and, other agencies. Corporations like Amazon and Google intend to deploy UAVs for the delivery of products and services [5]. 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. Additionally, due to their distinctive configuration, which includes open sensor states, wireless networks, serially safe structures, etc., these devices are highly exposed technological systems. There has been some research on cyber security risks to UAVs utilised in the defence industry recently, but not much has been done to investigate further cyber risks for UAVs that are accessible for purchase. Additionally, 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 [6].

Beyond simply assaulting the various system parts, CPS is under threat. A skilled 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 [7]. An attacker can directly access the UAV autopilot components through a hardware attack. While a sensor spoofing attack uses the on-board GPS channels to inject or transfer fake data, a wireless attack uses one of the wireless communication channels to carry out the attack. The wireless attack used to secure wireless data communication channels is the main topic of this study. While the UAV is in use, an attacker can conduct such attacks from a great distance. The biggest danger from wireless attacks is that if the communication protocol is identified, an attacker can take complete control of the UAV and decrypt the channel's encryption. An effective attack must compromise at least one of the following information security goals: confidentiality, integrity, or availability [8]. Intentionally disrupting a communication link while using a UAV to film an Australian triathlon is one example of a UAV attack. The driver claims that a ”channel hop” attacker purposefully interfered with his operation, leading him to lose whole control of the truck and crash into one of the athletes [9]. 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 [10].

Security issues are typically seen as the first major hurdle in all IoT application domains [11]. Additionally, every piece of established, deployed, and synchronised equipment and sensor in an IoT environment could be attacked from anywhere [12]. Data from node sensors are particularly important in many Internet of Things applications and must be secured. To guarantee a high level of security, current encryption techniques are linked to a considerable complexity of mathematical operations. Naturally, this numerical complexity necessitates using more time and energy. 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. [13] recommended the use of stream cyphers as simple cryptography methods for private information in IoT devices.

Lightweight encryption techniques for IoT have been deployed according to several specified procedures. Yao et al. [14] used Elliptic Curve Decisional Diffie Hellman and attribute-based encryption to create a lightweight encryption technique. They also recommended resource-limited equipment that was procedure-resistant and helped to improve performance efficiency in IoT security issues. In work by Salami et al. [15], 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. Baskar et al. [16] described the idea of WSNs, their elements, and their characteristics with a focus on WSNs, which were vulnerable to attacks. Additionally, they proposed a simple encryption method using a chaotic map to generate the key and a FGPA. In work by Yang et al. [17], authors recommended patients' secure data management using keyword search and simple Diffie Hellman encryption of patient health data in the healthcare context. Recently, Panagiotou et al. [13] 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. [18] 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. [19] have suggested a 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 goal of this work is to recommend a lightweight, low-power stream cypher scheme for protecting data transferred between UAVs and GCS.

The rest of this paper is arranged as follows: An overview of ChaCha20 is demonstrated in Section II. Section III presents the proposed keystream cipher. Section IV discusses the implementation of the proposed keystream cipher and the case study involving detailed simulated experimental results. Concluding remarks are provided in Section V.

## II. ChaCha20

In the areas of data storage, telephony, and the Internet, there is a significant evolution between 1990 and 2020. These contemporary evolutions require strong security, in line with strong cryptography method. A method for secure communication when there are attackers is cryptography [20]. Most cryptography techniques fall into one of two categories: The first is a block cypher that demonstrates how it works by dividing each original piece of data into successive blocks, each of which is then encrypted using the same key [21]. The second form, a stream cypher, shows how the cypher works by using the XOR function to combine the original data with a key random sequence to produce the cypher data [22].

For its applicability in areas of equipment and mobile communications, this study only focuses on stream cyphers. In comparison to block cyphers, they use more resources and have limited bandwidth, energy, and processing efficiency.

Google has implemented ChaCha20, a stream cypher that is used in counter mode for symmetric encryption. The ChaCha20 input(I) is 512 bits in size overall, as shown below:

$$I = \begin{bmatrix} 61707865 & 3320646e & 79622d32 & 6b206574 \\ K_{(0)} & K_{(1)} & K_{(2)} & K_{(3)} \\ K_{(4)} & K_{(5)} & K_{(6)} & K_{(7)} \\ Count & Nonce_{(0)} & Nonce_{(1)} & Nonce_{(2)} \end{bmatrix}$$

The bits act as 32 bit seeds, which consist of [23]:

- 256 bit is key size (k1… k8).
- 192 bit is the constants [c1 … c4] and nonce size (n1, n2).
- 64 bit is the block message counter size (b1, b2).

Therefore, three simple techniques are employed to combine the ChaCha20's input to form 512 bits series that symbolise the keystream. The first lightweight method is an addition (adding two 32-bit values), the second is Exclusive-OR (XORing the two 32-bit values), and the third is rotation (rotating 32-bit values by e bit [$y << e$), where e serves as a constant integer). The dual function procedure condenses the three lightweight procedures. The Quarter Round Function (QRF), the core of the dual function, was created to update each round's state

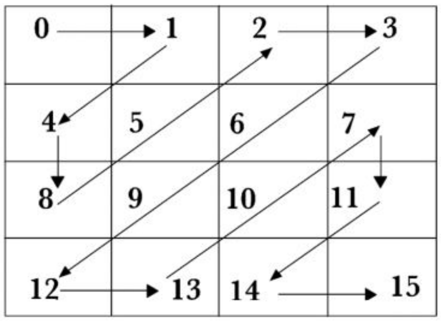


Fig. 1. Zigzag Form

matrix. The state matrix's diagonals are applied after the QRF has been applied to the state matrix's columns.

The three lightweight procedures' three 32-bit outputs are modified based on the QRF's four 32-bit input values as shown below:

$$\left\{\begin{array}{ll} p = p + q & s = (s \oplus p) <<< 16 \\ r = r + s & q = (q \oplus r) <<< 12 \\ p = p + q & s = (s \oplus p) <<< 8 \\ r = r + s & q = (q \oplus r) <<< 7 \end{array}\right\}$$

The final step in ChaCha20 involves adding the most recent matrix modification to the initial seed of the input ChaCha20 [24].

## III. Proposed Keystream Cipher

In order to create a novel keystream for raising security, the basics of the ChaCha20 have been researched. UAV communications will be encrypted using the generating keys. The new strategy consists of 10 rounds which makes a Lightweight Stream Cipher, described in Algorithm 1and 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. Proposed Keystream Cipher Implementation

Ardupilot, a Software-In-The-Loop (SITL), which uses the same MAVLink communication protocol and autopilot as a real drone, is used with a model UAV as the testbed for the experiment. A virtual drone's use directly generalises to a real drone'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 the proposed keystream cypher encryption to the communication

**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 $a \leftarrow 1to([PlainText/512])$ do
    O ← I
    for b ← 1 to 10) do
        O[0,1,4,8] ← QR(O[0,1,4,8])
        O[5,2,3,6] ← QR(O[5,2,3,6])
        O[9,12,13,10] ← QR(O[9,12,13,10])
        O[7,11,14,15] ← QR(O[7,11,14,15])
        O[0,4,1,5] ← QR(O[0,4,1,5])
        O[8,12,9,13] ← QR(O[8,12,9,13])
        O[2,6,3,7] ← QR(O[2,6,3,7])
        O[10,14,11,15] ← QR(O[10,14,11,15])
    end for
    $Sl \leftarrow Serial(O + I)$
    for c ← 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 (p, q, r, s)
Ensure: Update Four 32-bit (p, q, r, s)
$y_0$=$1^{st}$ 4 bits of r
$y_1$=$1^{st}$ 4 bits of p
$y_2$=$1^{st}$ 4 bits of q
$y_3$=$1^{st}$ 4 bits of s
$p = p + q \rightarrow s = (s \oplus p) <<< 16$
$c = r + s \rightarrow q = (q \oplus r) <<< 12$
$p = p + q \rightarrow s = (s \oplus p) <<< 8$
$c = r + s \rightarrow q = (q \oplus r) <<< 7$
```

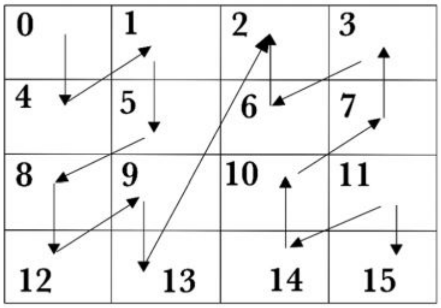
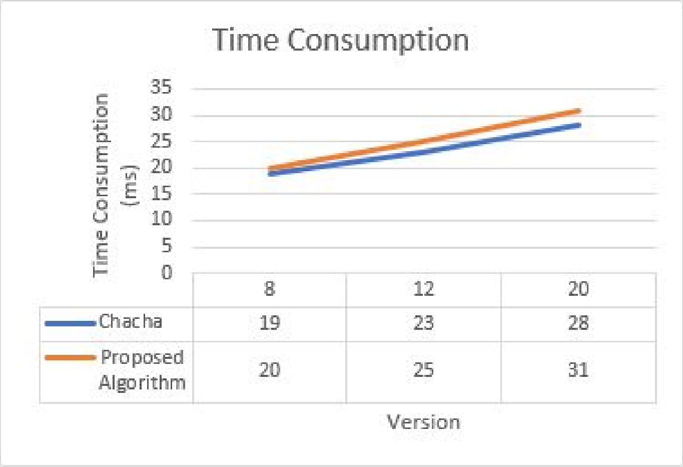


Fig. 2. Alternate Form



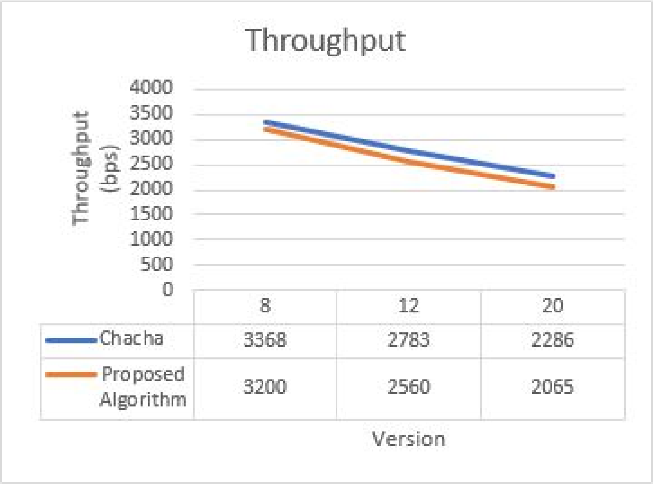
| | 8 | 12 | 20 |
|---|---|---|---|
| Chacha | 19 | 23 | 28 |
| Proposed Algorithm | 20 | 25 | 31 |

Fig. 3. Time Consumption Comparision

stream delivered between the GCS and drone's autopilot. In addition, we used Lorenz Meier's open-source QGroundControl ground station, a C++-based GCS application. In order to enable secure communication between the Ardupilot and the QGroundControl, we also added the proposed keystream cypher encryption to the QGroundControl. This allows it to decrypt the received cypher 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. To connect to the SITL, we used the UDP protocol and port 14551. The suggested keystream cypher model's 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). The three ChaCha versions are modified according to the findings of this experiment. The comparison shows that the suggested keystream cypher 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 3. Throughput and power consumption metrics are measured:

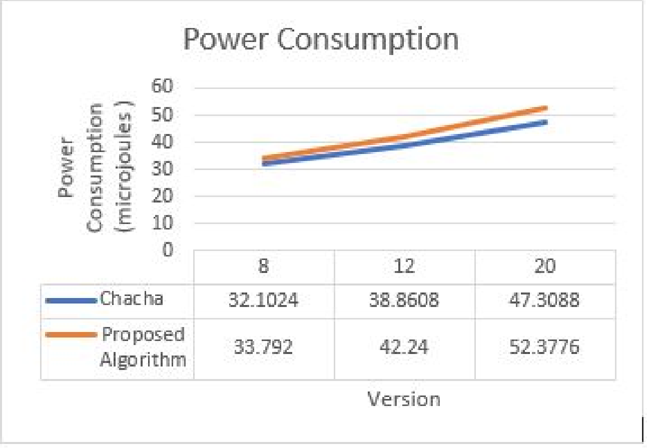
### A. Throughput

In accordance with the encryption method given for the proposed keystream cypher approach i.e., Algorithms 1 and 2 and for the standard ChaCha20, throughput had been taken. The total duration of encryption is the length of time it takes to transform the original data into cypher data. By dividing the payload length by the encryption time in seconds, we calculate the encryption process' speed (throughput). The keystream cypher presented is coupled with a very modest time increase (Fig 3) that causes a very slight reduction in throughput. It is anticipated that the throughput will drop as security is raised. It is important to note that the substantial increase in security causes a very tiny drop in throughput as can be seen in Fig 4.



| | 8 | 12 | 20 |
|---|---|---|---|
| Chacha | 3368 | 2783 | 2286 |
| Proposed Algorithm | 3200 | 2560 | 2065 |

Fig. 4. Throughput Comparision



| | 8 | 12 | 20 |
|---|---|---|---|
| Chacha | 32.1024 | 38.8608 | 47.3088 |
| Proposed Algorithm | 33.792 | 42.24 | 52.3776 |

Fig. 5. Power Consumption Comparision

### B. Power Consumption

The current, voltage, and clock cycles have been used to calculate the amount of power consumed in microjoules. The outcomes demonstrate a very modest increase in energy usage related to the proposed algorithm (with slight increase in time of less than 3 ms). In essence, the obvious improvement in security results in a very slight rise in energy use as seen in Fig 5.

## V. Conclusion

Depending on the number of rounds, there were many variations of the ChaCha stream cypher. There have been a few effective attacks against ChaCha 6, 7, and 8, but no entirely successful attacks on ChaCha 12 and 20 have yet been documented in the literature. A new stream cypher that differs from ChaCha cypher is proposed in the current work. The standard variants of ChaCha (8, 12, and 20) have been compared to the proposed keystream cypher based on a number of factors, including time, throughput, and power usage. Results show that with very modest increases in time, power usage, and slight reductions in throughput, the security is substantially raised. In terms of security, it takes 2512 likely keys, as opposed to 2248 probable keys for ChaCha8, to crack it using a brute-force attack. We think the suggested keystream cypher is appropriate for the security of unmanned aerial vehicles (UAVs), which need strong security yet have little energy and little storage capacity.

## References

[1] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Modeling security in cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 5, pp. 118–126, 12 2012.

[2] G. Dini and M. Tiloca, "A simulation tool for evaluating attack impact in cyber physical systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8906, pp. 77–94, 2014. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-13823-7_8

[3] R. Austin, "Unmanned aircraft systems: Uavs design, development and deployment," *Aeronautic Aerospace Engineering*, 2011. [Online]. Available: $https://www.wiley.com/en-us/Unmanned+Aircraft+Systems\%3A+UAVS+Design\%2C+Development+and+Deployment-p-9780470058190$

[4] "Mcafee labs 2017 threats predictions: "dronejacking" places threats in the sky," 2016. [Online]. Available: https://www.mcafee.com/au/resources/reports/rp-threats-predictions-2017.pdf

[5] 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/abs/10.1177/1548512915617252

[6] K. Mansfield, T. Eveleigh, T. H. Holzer, and S. Sarkani, "Unmanned aerial vehicle smart device ground control station cyber security threat model," *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pp. 722–728, 2013.

[7] 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/10.2514/6.2012-2438

[8] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pp. 585–590, 2012.

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

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

[11] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *J Netw Comput Appl*, vol. 84, pp. 25–37, 4 2017.

[12] A. Giaretta, S. Balasubramaniam, and M. Conti, "Security vulnerabilities and countermeasures for target localization in bio-nanothings communication networks," *IEEE Trans Inf Forensics Secur*, vol. 11, pp. 665–676, 4 2016.

[13] 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.

[14] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Gener Comput Syst*, vol. 49, pp. 104–112, 8 2015.

[15] 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.

[16] C. Baskar, C. Balasubramaniyan, and D. Manivannan, "Establishment of light weight cryptography for resource constraint environment using fpga," *Procedia Comput Sci*, vol. 78, pp. 165–171, 2016.

[17] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health internet of things," *J Netw Comput Appl*, vol. 89, pp. 26–37, 7 2017.

[18] 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.

[19] 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.

[20] N. H. M. Mahdi, "Design of keystream generator utilizing firefly algorithm," *J Al Qadisiyah Comput Sci Math*, vol. 10, p. 91, 2018.

[21] M. S. Mahdi, N. F. Hassan, and G. H. Abdul-Majeed, "An improved chacha algorithm for securing data on iot devices," *SN Applied Sciences*, vol. 3, pp. 1–9, 4 2021. [Online]. Available: https://link.springer.com/article/10.1007/s42452-021-04425-7

[22] M. Yao and J. Ma, "Stream ciphers on wireless sensor networks," *Proceedings - 3rd International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2011*, vol. 3, pp. 358–361, 2011.

[23] P. Yadav, I. Gupta, and S. K. Murthy, "Study and analysis of estream cipher salsa and chacha," *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, pp. 90–94, 9 2016.

[24] D. J. Bernstein, "ChaCha, a variant of Salsa20."