International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

GEO based Adaptive Dynamic Key Generation Scheme for Speck-R in IoT Applications

M. Abinaya[1], S. Prabakeran[2]
[1]Research Scholar, Networking and Communication
SRM Institute of Science and Technology, Kattankulathur
Chengalpattu, Tamilnadu, India
am8580@srmist.edu.in
[2](Corresponding Author)
Assistant Professor, Networking and Communication
SRM Institute of Science and Technology, Kattankulathur
Chengalpattu, Tamilnadu, India
prabakes@srmist.edu.in

Abstract— By integrating lightweight properties with lightweight cryptography in Intenet of Things (IoT) applications, information confidentiality and dependability can be guaranteed. In order to cut down on the number of rounds and speed up execution, a dynamic key-substitution layer was included to the original Speck's structure to create the ultra-lightweight encryption algorithm Speck-R.In this paper, we propose to design Golden Eagle Optimizer (GEO) based adaptive dynamic key generation scheme for Speck-R. In this algorithm, the block size and key size are adaptively chosen based on the IoT application and data size. Then the dynamic key (DK) is generated using GEO algorithm in which a fitness function is derived in terms of the metrics LPF, DPF and BIC. Then the key with optimum fitness is chosen as the dynamic key for further encryption /decryption process of Speck-R. By simulation results, we show that the proposed technique enhances the security.

Keywords- IoT Security; Lightweight Cryptography; Speck-R; Dynamic Key Generation; Golden Eagle Optimizer.

I. INTRODUCTION

Internet of Things (IoT) is an environment where networked devices are connected to one another to allow for inter-device communication. IoT offers technical answers to societal issues in order to make society more intelligent. As we utilize IoT devices on a regular basis, they have become a part of our lives [1]. Due to its numerous applications, internet of things is currently enticing everyone. IoT is one of the most cutting-edge technological advancements now being used to make everything smarter, but it also has serious security flaws. Our lives have transformed as a result of its quick development and widespread applicability in verifiable apps. The standards that provide adaptable, secure and reliable communication without compromising the resource constraints of the IoT devices, are required [2] [3].

Build user confidence in IoT technology's security and privacy so they may utilize it with confidence that their data integrity, confidentiality, and authority won't be seriously threatened. The IoT is intrinsically exposed to a number of security concerns; if crucial security steps are not performed, there is a danger of information theft which could cause many problems. Those dangers could be viewed as a significant barrier to IoT [4]. IoT is susceptible to attacks as its constituents have a good potential of being physically attacked given how long they are left unattended. Second, eavesdropping is incredibly easy because of the wireless medium. At last, the IoT components exhibit low proficiency in energy consumption and processing power. When IoT is used for monitoring reasons, a sizable volume of data is anticipated to be generated, hence it is crucial to maintain data unification [5]. Data integrity and authentication are the specific issues that need attention.

In general, the term "security" refers to a wide range of characteristics that must be present in any information network in order to guarantee end-to-end security. The secrecy of the data is a crucial security feature. Only personnel who have been authorized by some means before accessing the data may see its contents in order to maintain data confidentiality. At rest (during storage) and in transit (while communication), the confidentiality of the data must be guaranteed. In general, secrecy is provided using encryption; but, because to the abundance of IoT devices and their resource limitations, securing confidentiality becomes difficult. Scalability poses a problem on the one hand, and the resources of the devices make it difficult to directly use traditional encryption techniques in the IoT [6].

The majority of IoT devices is insignificant in size and come with fewer resources. To process the data, they also have less computer power. The bulk of IoT devices also deal with real-time applications, in which it may be challenging to react fast and accurately while maintaining essential security [7]. Energy capacity and data security are two hazards and issues that IoT device designers must deal with. In these conditions, IoT devices' performance may not be acceptable if traditional cryptography standards are used to them [8].

By integrating lightweight properties with lightweight cryptography, the aforementioned problems are effectively addressed. Another crucial feature of lightweight cryptography (LWC) is that it is easily applicable to other resource-rich devices that it interacts with directly or indirectly. Using a LWC technique, information confidentiality and dependability can be guaranteed [9]. The idea of LWC is to create a security system that can run on devices having restricted resources by utilizing less memory, less material, and less power [10] [11]. The security of lightweight encryption is stronger. Although the RSA/Elliptic Bend (marking) and SHA-256 (hazing) of the AES (encryption) have functioned brilliantly with our standard encryption technology, frames with reasonable handling power and capabilities, these don't extend to a world where frames and sensor organizations have been established [12].

A. *Motivation and Objectives*

It can be observed that a lightweight encryption algorithm should meet the subsequent measures:

- Minimum Block Size
- Minimum Key size
- Minimum number of rounds

In Speck-R [15], the following three main goals have to be fulfilled

- Linear Probability approximation Boolean Function (LPF) should be minimum
- Differential Probability approximation Function (DPF) should be minimum
- Output Bits Independence Criterion (BIC) should be low

For meeting these goals, we propose to design Golden Eagle Optimizer (GEO) based adaptive dynamic key generation scheme for Speck-R.

II. RELATED WORKS

A LWC based encryption technique was suggested by Usman et al. [13]. It is a 64-bit block cipher. The architecture of the algorithm is a cross between a uniform substitution-permutation network and a Feistel network. Experimental results have proven that the method provides notable security in just 5 encryption rounds. The outcomes are evaluated with respect to code size, memory consumption, and execution cycles.

This study describes related work for lightweight secure data transmission strategies by Abdul et al. [14]. Handling security vulnerabilities in IoT is still challenging. In order to handle these security issues, LWC schemes were devised. This paper has presented several of these techniques and their equivalent methods. The low power and light weight of the sensors used in Internet of Things devices made it necessary to develop lightweight wireless solutions. In addition to evaluating the performance of various LWC algorithms, this study examined the main security concerns faced by IoT devices.

Based on de Ree et al.'s [16] description of the abilities of cryptographic techniques for ensuring data secrecy, they have evaluated the suitability of different methods for resource utilization of IoT networks. They have addressed potential in the field of Physical Layer Security (PLS), basing our discussion on the shortcomings of cryptographic solutions that have been revealed. They have concluded by providing a summary of PLS techniques designed to improve data secrecy in IoT networks.

A brand-new, extremely lightweight block ciphering technique for IoT (LBC-IoT) is suggested in this research by Ramadan et al [17]. The Feistel structure is the basic foundation for the block length of 32 bits and key length of 80 bits. Small stiff substitution boxes and the usage of straightforward functions are examples of energy-efficient cryptographic features. Additionally, it is adaptable in terms of implementation and resistant to various assaults, including linear, differential, and side-channel attacks. In addition, compared to other modern algorithms, it delivers respectable performance in both hardware and software. The findings of hardware implementation are highly encouraging and competitive with the top lightweight ciphers used today. Additionally, it is perfectly suited for extremely limited devices like RFID tags.

To fix the security vulnerability brought on by using different keys for each round function, Mhaibes et al. [18] suggested upgrading TEA by implementing a key function generation by Linear Feedback Shift Registers (LFSRs) in conjunction. To assess its security performance, Avalanche effect, the key sensitivity, and a completeness test will be utilized. According to experimental findings, the proposed modified TEA performs encryption better than the original TEA.

To categorize the different kinds of encryption techniques, Hasan et al. [19] reported the experimental analysis of cryptographic methods. It offers a thorough evaluation of the temporal complexity, size, encryption, and decryption capabilities of AES, DES, 3DES, RSA, and Blowfish. More bytes an encryption can support is the key size utilized for each encryption technique. The three devices' combined average computation times for the algorithm are utilized for comparison. For IoT applications, a collection of plaintexts are utilized in the experimental test’s simulation, and the results are compared to the performance of the existing real-time deep learning procedures.

A. *Research Gap*

To mitigate the threat, researchers began to rethink their approaches and made improvements to the network and application levels. This would not, however, change the reality that the simpler devices are still more susceptible to attacks than the more complex ones. One of the methods is just draining these devices' battery for putting them in a Denial-of-Service mode, which causes them to shut down eventually. The exchanged data must be encoded in order to guarantee its security, privacy, and resistance to various types of attacks. However, in such strict conditions, typical methods of encryption are ineffective and ineffective. LWC was therefore recommended to satisfy the needs in such constrained devices. Numerous studies have focused on using simple techniques to effectively encrypt data. The resources needed for lightweight cryptography must be lower, and any computing overhead brought on by the encryption/decryption process must be eliminated.

III. PROPOSED METHODOLOGY

A. Overview

In this algorithm, the block size and key size is adaptively chosen based on IoT application and data size. Then the dynamic key (DK) is generated using GEO algorithm in which a fitness function is derived in terms of the metrics LPF, DPF and BIC. Then the key with optimum fitness is chosen as the dynamic key for further encryption /decryption process of Speck-R.

B. Selection of Key and Block Size

If the data size is less, the block size is chosen between32 to 64 and if it is beyond that, the block size is chosen between 96 and 128 [14]. In table 1, different combinations of Speck algorithm and its specifications are shown.

TABLE I. DIFFERENT COMBINATIONS OF SPECK AND SPECIFICATIONS

Block and Key Size (bits)	**Rounding Parameter**	**Number of Rounds**
(32,64)	(7,2)	22
(48,72)	(8,3)	22
(48,96)	(8,3)	23
(64,96)	(8,3)	26
(64,128)	(8,3)	27
(96,96)	(8,3)	28
(96,144)	(8,3)	29
(128,128)	(8,3)	32
(128,192)	(8,3)	33
(128,256)	(8,3)	34

C. Derivation of Fitness Function

1) *Linear Probability Approximation Boolean Function (LPF):* This aids in identifying a linear relationship or approximation connecting some plain-text bits with their corresponding ciphered ones. It will be easier to extract the key if there is a linear relationship between the plaintext and the cipher text. LPF needs to be very low to achieve a higher level of resistance to linear attacks.
2) *DPF:* One of the key characteristics of any replacement layer to produce the nonlinear transformation and prevent variant cryptanalysis attacks is differential probability. In this attack, the cryptanalyst tries to take advantage of the high variance of two plaintexts. Differential uniformity must be present in the substitution layer. A minimum DPF is required.
3) *Output BIC:* This condition assesses the degree of dependence among the output bits following their substitution. This criterion states that inverting an input bit modifies the output bits, independently of one another. BIC ought to be little.

Fitness Function:

It is defined as follows:

$$F = \min \{LPF, DPF, BIC\} \tag{1}$$

Position of Eagle:

The location of each eagle *i* is replaced with the new location, if the fitness of the new location superior than the location stored in its memory.

$$P^{t+1} = P^t + \Delta P_I^t \tag{2}$$

D. Dynamic Key Generation using GEO Algorithm

Then the dynamic key (DK) is generated using GEO algorithm in which a fitness function is derived in terms of the metrics LPF, DPF and BIC. Then the key with optimum fitness is chosen as the dynamic key for further encryption /decryption process of Speck-R.

The movement of the eagles consists of attack and a vector.

Golden Eagle Optimizer (GEO)

The GEO algorithm draws its inspiration from the ability of golden eagles to alter their speed as they spiral through the air in search of prey. The golden eagle follows a circular path during the cruising and hunting stages. In the early stages of hunting, it exhibits a strong propensity to move around looking for the prey, and a stronger propensity to attack.

Each eagle memorizes the optimum position it has attained so far. It is interested in attacking the prey and searching for better food [19].

The dynamic key (DK) is generated using GEO algorithm in which a fitness function is derived in terms of the metrics LPF, DPF and BIC (explained in section III.C).

Algorithm Steps:

Step 1: Golden Eagles population is initialized

Step 2: Fitness function is estimated

Step 3: Population memory is initialized

Step 4: Set w_a and w_t

$$w_a = w_a^t + \frac{t}{T}\,|w_a^T - w_a^0| \tag{3}$$

$$w_c = w_c^t + \frac{t}{T}\,|w_c^T - w_c^0\,| \tag{4}$$

***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

where, t is the current iteration, T is the maximum iteration, w_a^0 and w_a^T denotes initial and final values for propensity to attack, respectively, w_c^0 and w_c^T denotes initial and final values for propensity to cruise, respectively.

Step 5: For each eagle i

Step 5.1 : Randomly select a prey from the population

Step 5.2 : Calculate the LPF, DPF and BIC

Step 5.3 : Update the position

Step 5.4 : Evaluate the fitness for the new position

Step 5.5 : Choose the key with optimum fitness, as the dynamic key

Step 5.6 : Update the new position with the earlier position of i

Step 6: End for.

E. Encryption Process

Any type of data, including texts, photos, and other types of data, can be encrypted using Speck-R. When a picture is encrypted, it has the following dimensions: A× B × C, where A, B, and C indicate the column, row, and plane numbers.

A single block Nonce (64 bits) is used by Speck-R, which divides it into two blocks of 32 bits. After going through Speck-R, the 32 bits input block will be stored along with the encrypted N[0] and N[1] results of 32 bits.

All of the blocks in the input plain-text are still covered by the encryption.

If n stands for the number of bytes selected for the block in each Speck version, then Z represents the total number of blocks, given by

$$Z = \frac{(A \times B \times C)}{n} \tag{5}$$

where i indicates the block index for i ∈ {0, 1, ..., Z}.

The block size and key size are adaptively chosen depending on the type of IoT application and data size. (Explained in section 3.2)

The following operations are included in this process:

- Encryption of Nonce
- Crossing through the substitution layer
- Plain text is XoRed with the final substituted value.

Encrypting the Nonce

Let N[0] be represented as U and N[1] be represented as V

Each block will go through the identical round Speck function, which is denoted by the equations given below:

$$O_0 = (U >>> 8) + V) \oplus K_p \tag{6}$$

$$O_1 = (U <<< 3) \oplus O_0 \tag{7}$$

K_p = key used for every round and generated using GEO

Speck-R switches the two outputs O_0 and O_1 following the encryption of U and V. To put it another way, U will replace V and vice versa.

Substitution Layer

In this technique, the three different substitution boxes are generated (K1, K2, K3). To achieve the confusion attribute, a non-linear element called a substitution table is introduced to the cypher.

The substitution layer is set to be dynamic, which implies that it is constructed based on a previously selected key. Unlike existing cypher methods, the suggested method is based on a dynamic key that is variable and varies in a pseudo-random manner for each new session. The periodic interval of a session is determined by the application or by the needs of the user. Changing the key each time results in a different substitution layer that changes as the number of iterations increases. Not only did adding a dynamic layer result in a more secure cypher, but it also lowered the number of iterations from 26 to 7.

XoR the Palin-Text

The plain text will be split into 64 bits of block each, after completing the substitution layer. Then, the O_0 and O_1 replacement values will be XoRed into the first and second 32 bits, R_0 and R_1, respectively. This is expressed by the equation that goes like this:

$$O_L = R_0 \oplus K_1 [O_0] \tag{8}$$

$$O_R = R_1 \oplus K_1 [O_1] \tag{9}$$

All of the block are then joined and aligned to produce the final result.

Decryption Process

There are seven rounds required for decryption. With encrypted Nonces, the ciphered scheme will be XoRed. Since the identical substitution tables will be utilized, an inverse substitution layer is not required. The decryption procedure is identical to the encryption procedure, which gives the system a significant benefit with respect to software/hardware deployment.

IV. Experimental Results

Here, the proposed Speck-R-GEO scheme's performance is evaluated to enhance the security parameters of Speck-R cipher, by implementing in IoT network.

The proposed Speck-R-GEO scheme is compared with the existing Speck-R and traditional Speck encryption schemes. The tests are conducted for set of 64/128 and 128/256 block and key sizes. To validate the efficiency of Speck-R-GEO, we have considered plain text data of various sizes on different IoT devices. The throughput and execution time parameters are measured.

A. Throughput

In this section, the throughput obtained for different data sizes of length 500 bytes to 5000 bytes are presented.

TABLE II. THROUGHPUT FOR SIZE 64/128

Data Size (Bytes)	Speck	Speck - R	Speck – R GEO
500	1035	2100	2350
1000	656	1480	1750
1500	624	1458	1747
2000	598	1432	1733
2500	581	1416	1720
3000	579	1401	1686
3500	566	1390	1682
4000	550	1386	1675
4500	534	1372	1671
5000	530	1370	1665

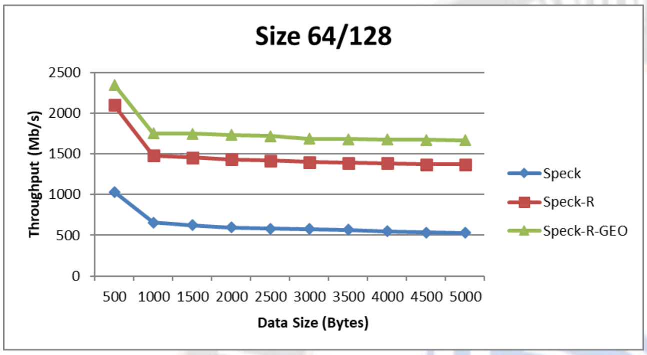

Figure 1. Throughput for size 64/128

TABLE III. THROUGHPUT FOR SIZE 128/256

Data Size (Bytes)	Speck	Speck - R	Speck – R GEO
500	1287	2475	2852
1000	756	1650	1980
1500	734	1638	1977
2000	728	1632	1973
2500	721	1626	1970
3000	719	1623	1966
3500	716	1620	1962
4000	710	1616	1960
4500	709	1612	1958
5000	706	1610	1955

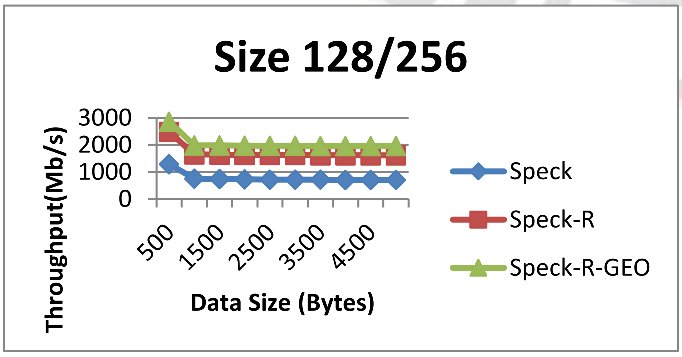

Figure 2. Throughput for Size 128/256

Figure 1 and 2 show the throughput measured in Mb/s for the sizes of 64/128 and 128/256 , respectively. As it is seen from the figures, the throughput of Speck-R-GEO is 65% and 16% higher than Speck and Speck-R, respectively, for both the sizes.

B. *Execution Time*

In this section, the execution times (in ms) for encrypting different data sizes of length 500 bytes to 5000 bytes are presented.

TABLE IV. EXECUTION TIME FOR 64/128

Data Size (Bytes)	Speck	Speck - R	Speck – R GEO
500	60	35	30
1000	75	48	37
1500	89	55	46
2000	104	62	54
2500	115	70	65
3000	123	83	72
3500	138	96	84
4000	153	103	90
4500	165	118	105
5000	170	135	118

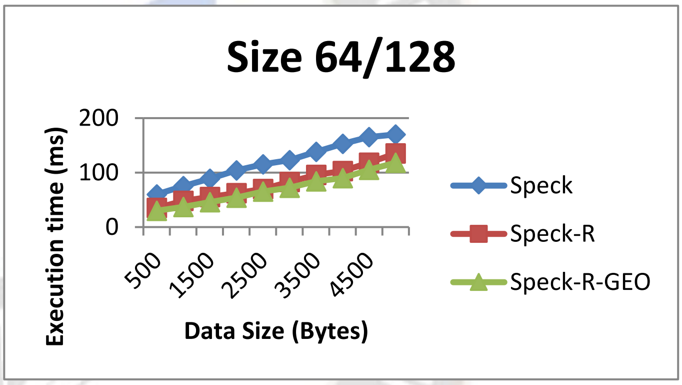

Figure 3. Execution Time for Size 64/128

TABLE V. EXECUTION TIME FOR SIZE 128/256

Data Size (Bytes)	Speck	Speck - R	Speck – R GEO
500	85	50	40
1000	97	64	54
1500	106	75	62
2000	120	82	70
2500	134	93	78
3000	156	110	90
3500	178	122	105
4000	210	135	117
4500	228	150	128
5000	240	166	140

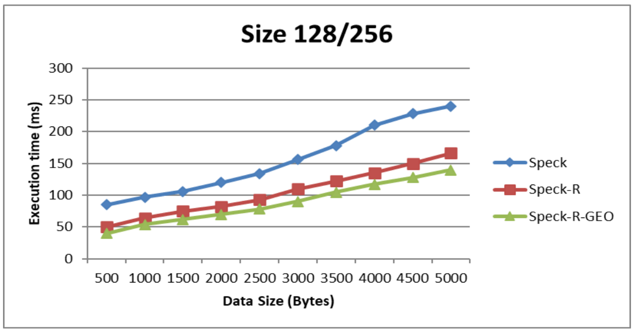

Figure 4. Execution Time for Size 128/256.

Figure 3 and 4 show the execution times of encryption, for the sizes of 64/128 and 128/256, respectively. As it is seen from the figures, the execution time of Speck-R-GEO is 43% and 14% lesser than Speck and Speck-R, respectively, for both the sizes.

V. CONCLUSION

In this paper, we have proposed to design Golden Eagle Optimizer (GEO) based adaptive dynamic key generation scheme for Speck-R. In this algorithm, the block size and key size is adaptively chosen based on the IoT application and data size. Then the dynamic key (DK) is generated using GEO algorithm in which a fitness function is derived in terms of the metrics LPF, DPF and BIC. Then the key with optimum fitness is chosen as the dynamic key for further encryption /decryption process of Speck-R. The proposed Speck-R-GEO scheme is compared with the existing Speck-R and traditional Speck encryption schemes. The tests are conducted for set of 64/128 and 128/256 block and key sizes on plain text data of various sizes on different IoT devices. Experimental results have shown that Speck-R-GEO attains higher throughput and lesser execution time, compared to the existing Speck versions.

REFERENCES

[1] P. V. Dudhe, N. V. Kadam, R. M. Hushangabade and M. S. Deshmukh, ”Internet of Things (IOT): An overview and its applications,” 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 2650-2653.

[2] Abinaya.M and Prabakeran S, ”Lightweight Block Cipher for Resource Constrained IoT Environment—An Survey, Performance, Cryptanalysis and Research Challenges”, 3rd International Conference on IoT Based Control Networks and Intelligent Systems, ICICNIS 2022, Kottayam, 2022.

[3] S. R. J. Ramson, S. Vishnu and M. Shanmugam, ”Applications of Internet of Things (IoT) – An Overview,” 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2020, pp. 92-95.

[4] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf and Y. A. Bangash, ”An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security,” in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10250-10276, 2020.

[5] S. El-Gendy and M. A. Azer, ”Security Framework for Internet of Things (IoT),” 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2020, pp. 1-6.

[6] E. Valea, M. Da Silva, M. -L. Flottes, G. Di Natale, S. Dupuis and B. Rouzeyre, ”Providing Confidentiality and Integrity in Ultra Low Power IoT Devices,” 2019 14th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), Mykonos, Greece, 2019, pp. 1-6.

[7] P. Kaur and S. Aggarwal, ”Cryptographic algorithms in IoT - a detailed analysis,” 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), Mohali, India, 2021, pp. 45-50.

[8] S. Surendran, A. Nassef and B. D. Beheshti, ”A survey of cryptographic algorithms for IoT devices,” 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2018, pp. 1-8.

[9] P. Shah, M. Arora and K. Adhvaryu, ”Lightweight Cryptography Algorithms in IoT – A Study,” 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 332-336.

[10] A. I. Regla and E. D. Festijo, ”Performance Analysis of Light-weight Cryptographic Algorithms for Internet of Things (IoT) Applications: A Systematic Review,” 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-5.

[11] Vishal A. Thakor, Mohammad Abdur Razzaque, and Muhammad R. A. Khandaker, “Lightweight Cryptography Algorithms forResource-Constrained IoT Devices: A Review, Comparison and Research Opportunities”, IEEE Access, 2021.

[12] El-hajj, M., Mousawi, H., Fadlallah, A.,“Analysis of LightweightCryptographic Algorithms on IoTHardware Platform,” Future Internet, 2023, 15, 54.

[13] Muhammad Usman, Irfan Ahmed, M. Imran Aslam, Shujaat Khan and Usman Ali Shah, “SIT: A Lightweight Encryption Algorithm for Secure Internet of Things”, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1, 2017.

[14] Ashu Abdul, Garlapati Narayana, R. Sudha Kishore, B. Srikanth, K. Kranthi Kumar, D.N.V.S.L.S. Indira, “LWC: Efficient Lightweight Block Ciphers For Providing Security To Constrained Devices A Solution For IoT Devices,” Journal of Theoretical and Applied Information Technology, 2023, Vol.101. No 7.

[15] Lama Sleem and Raphael Couturier, “Speck-R: An ultra light-weight cryptographic scheme for Internet of Things”, Multimedia Tools and Applications (2021) 80:17067–17102.

[16] M. de Ree et al., ”Data Confidentiality for IoT Networks: Cryptographic Gaps and Physical-Layer Opportunities,” 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Porto, Portugal, 2021, pp. 1-6, doi: 10.1109/CAMAD52502.2021.9617779.

[17] Rabie A. Ramadan, Bassam W. Aboshosha, Kusum Yadav, Ibrahim M. Alseadoon, Munawar J. Kashout and Mohamed Elhoseny, “LBC-IoT: Lightweight Block Cipher for IoT

Constraint Devices", Computers, Materials & Continua, CMC, 2021, vol.67, no.3.

[18] Hakeem Imad Mhaibes, May Hattim Abood, and Alaa Kadhim Farhan, "Simple Lightweight Cryptographic Algorithm to SecureImbedded IoT Devices", IJIM, Vol. 16, No. 20, 2022.

[19] Mohammad Kamrul Hasan,MuhammadShafiq,Shayla Islam, Bishwajeet Pandey,Yousef A. Baker El-Ebiary,Nazmus Shaker Nafi, R. Ciro Rodriguez,and Doris Esenarro Vargas, "Lightweight Cryptographic Algorithms for Guessing Attack Protection in Complex Internet of Things Applications", HindawiComplexity, 2021.

[20] Mohammadi-Balani, A., Dehghan Nayeri, M., Azar, A., Taghizadeh-Yazdi, M., "GoldenEagle Optimizer: A nature-inspired metaheuristic algorithm", Computers & Industrial Engineering (2020), doi:https://doi.org/10.1016/j.cie.2020.107050,