IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 8, Issue.10, October 2020 E-ISSN: 2347-2693

NLBSIT: A New Lightweight Block Cipher Design for Securing Data in IoT Devices

Abdulrazzaq H. A. Al-Ahdal[1,2]*, Galal A. AL-Rummana[1], G.N. Shinde[3], Nilesh K. Deshmukh[4]

[1,2]School of Computational Sciences, S.R.T.M. University, Nanded, India
[1]Dept. of Computer Science & Engineering, Hodeidah University, Yemen
[3]Yeshwant College, Former Pro-Vice Chancellor, S.R.T.M. University, Nanded, India
[4]School of Computational Sciences, S.R.T.M. University of organization, Nanded, India

*Corresponding Author: alahdal201211@gmail.com, Tel.: +91-92847-12177

DOI: https://doi.org/10.26438/ijcse/v8i10.164173 | Available online at: www.ijcseonline.org

Received: 20/Oct/2020, Accepted: 25/Oct/2020, Published: 31/Oct/2020

***Abstract*—** Modern applications consist of different types of control devices and sensors that connect to the Internet. These applications are new approved technologies called the Internet of Things. Nowadays, these new technologies have gained a great interest in the field of research because of their existence in several diverse fields and due to the rapid development of these technologies. Communication between these devices generates a large amount of private and sensitive information and data between them. Therefore, maintaining the confidentiality of that data and information in the Internet of Things is of great importance. Mathematical cost (complex mathematical operations) and the number of cycles in traditional cryptographic algorithms leads to a large use of memory and energy waste for devices with limited resources, which makes traditional cipher algorithms inappropriate for Internet of Things devices. A fast and LW algorithm called NLBSIT has been proposed in this regard, which provides the requisite protection and resource constrained confidentiality of data on IoT devices. This algorithm (NLBSIT) uses a 64-bit key to encode 64-bit data, uses simple mathematical operations (XOR, XNOR, shifting, swapping), and uses the features of both the Feistel and SP Network architecture to achieve diffusion and confusion (increasing data security). The FELICS and MATLAB tools are used to simulate the NLBSIT algorithm. To execute this algorithm, various data types are used, such as text and images. The results of the simulation indicate the supremacy of the proposed algorithm in various areas, such as security, efficiency, less cycles (encryption and decryption), and less memory usage.

***Keywords*—**Lightweight Cryptography LWC; FELICS; RFID tags; IoT Security;

I. INTRODUCTION

The Internet of Things is new in the world of technology. It has made many changes in human life. It provides many advanced computing facilities. For this reason, it has become the focus of researchers' attention and focus. IoT defines the network of physical objects — "things" — that are equipped with sensors, apps, and other technologies that communicate and share data over the Internet with other devices and systems [1]. Currently, there is a great jump in IoT applications around the globe. IoT is increasingly guiding the real world to grow smarter across numerous demanding fields of operation such as home and building automation [2], [3], smart traffic [4], smart healthcare, and smart energy grids [5]. IoT is rapidly expanding in this new data world. Nevertheless, insufficient security and privacy measures affect its sustainable development. Via the various sensors, IoT devices become typically responsible for gathering various kinds of sensed data; some of them may be sensitive information from the user's point of view and may not be revealed to several people. According to numerous studies, security is the most exciting issue for IoT users today. Therefore, it is important to incorporate more security functionality into both the hardware and the system software in order to enhance the security of the IoT device layer and ensure the privacy of sensitive sensed information.

In this plan, integrating the principle of cryptography on the functional shortcomings of IoT devices is desirable. Using cryptographic algorithms in IoT devices, the data will be secured and integrated before it is published on the back-end networks (cloud server or other). While security based on cryptography is increasingly necessary. However, the security based on the approved encryption algorithms is a big challenge in the hardware/software of IoT due to having large area and consume overhead of power. In addition, the encryption operations in the Internet of Things must contain robust and effective management for the public key. The strong key management leads to the protection of data integration and security. IoT devices are essentially lightweight, meaning that they should have a little storage capacity. Furthermore, IoT devices are mostly battery-driven; as such, low power consumption is required. Therefore, the symmetric key cryptography algorithms are commonly preferred for the design of IoT devices due to their lower computing capacity, storage space, complexity in comparison to the asymmetric key cryptography algorithms, which take more process and

more memory [6]. This means the current trend towards lightweight encryption is used instead of traditional encryption because traditional encryption takes several mathematical operations and exhausts the devices' energy [7].

Block cipher is a form of secret key cryptography that was introduced in the last decade to develop resource-constrained computing devices for its simpler software and hardware implementation, higher diffusion. Compared with the stream cipher, it requires highly constrained hardware resources [6]. In contrast, when building block ciphers, conventional Feistel ciphers take a lot of time and use a lot of energy because they require much more circular functions than replacement-permutation networks (SPNs). In this respect, the synthesis of both the Feistel and SPN system has been suggested in one system. The synthesis structure provides lightweight and compact circular functions, with reduced energy consumption and storage, contributes to rapidly diffusion [8], [9]. A LW symmetric key cryptographic technique has been proposed here in view of all the challenges and drawbacks of small IoT devices by enhancing all the susceptibilities present in proven literature. Furthermore, the suggested cryptographic technique is developed by integrating the arrangement of Feistel and SPN together to ensure quicker diffusion. In comparison, encryption technique has also followed a circular key management technique.

The remainder of the paper was structured in the following way: Section II presents in depth the relevant works in the required area. The suggested cryptographic lightweight technique was described in depth in Section III the experimental environment is presented in Section IV, V, and VI with findings and multiple forms of analysis. The report was concluded in Section VII.

II. LITERATURE SURVEY

There is a clear need for LWC to cope with the data size, processing capacity and cost of small computer devices to a minimum. Therefore, when developing a cryptography algorithm dedicated to any small computing system, the key objective should make it LW in all respects such as usage of memory, consumption of power [10]. Algorithms such as CLEFIA [11], PRESENT [12], LED [13], KATAN/KTANTAN [14], PICCOLO [15], PRINCE [16], PRIDE [17], SIMON/SPECK [18], TEA [19], and LEA [20] are still being followed and used in inside different fields of software. This segment has outlined some of the most popular recently published lightweight block ciphers.

A group of authors suggest a family of LWC ciphers named CHAM dependent on a 4-branch Feistel structure mechanism performing ARX procedures. CHAM is made up of three algorithms which have various criteria. Both three family algorithms are ideal for resource-constrained environments but have subtly different implementations according to their parameters at the same time. CHAM-64/128 is best suited for low-end applications, CHAM-128/128 is a better option for general use on 32-bit microcontrollers and CHAM-128/256 will work when a higher degree of protection is necessary [21].

In [22] suggested a LWC technique based on SPN, and called it BORON. It supports 128/80 key bits along with a 64-bit plaintext block. It is based on a total of 25 rounds, where the 4-bit to 4-bit S-boxes are used, followed by round shift, permutations, and XOR operations. It is also immune to attacks, differential and linear.

In [23] a LW block cipher algorithm called LiCi was developed. It considers 31 consecutive rounds paired with 4 × 4 LW S-boxes. The cipher LiCi is a balanced network of Feistel structure and its architecture supports 128 bits key for 64-bits plaintext.

The 'RECTANGLE' SPN-based block cipher algorithm was suggested in [24] to allow 64-bit plaintext and 80-bit or 128-bit keys for the production of 64-bit ciphertext. It has 25-round. Each of the 25 rounds is composed of three steps: AddRoundkey, SubColumn, and ShiftRow. Some of this approach's advantages are like very hardware friendly design, better competitive performance of applications.

In [25] suggested a LW block cipher-based software, and named it ITUbee. It uses 20 rounds and main layers of whitening to ensure greater protection. It supports 80-bit for plaintext block and key-length. It utilizes AES S-box and reduced the memory, strength, and time needed. ITUBEE was believed to have been immune to the associated key attack.

Simeck, a new family of LW block ciphers, combines Speck and Simon's good design components to make block ciphers even more compact and efficient. It is proposed in [26]. For key lengths of 64, 96, and 128 bits, it supports plaintext 32, 48, and 64 bits. SIMECK has also been shown to withstand bit-flip and random-byte risks of failure.

LED signifies (Lightweight Encryption Device), suggested in 2011 by Guo et al. [27]. It is one of the latest ciphers of lightweight. It offered protection against threats of the state-of-the-art. The cipher state uses 4-bit matrix to represent ordered definitions. The LED is similar to the lightweight PHOTON hash function, too.

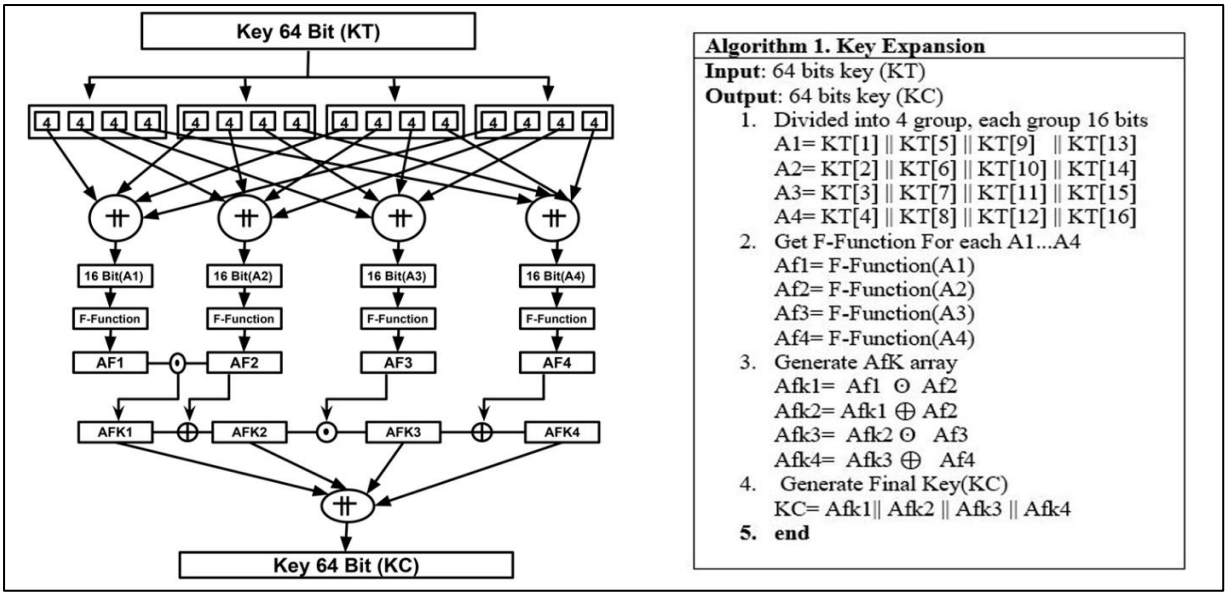
In [8] a lightweight block cipher algorithm for IoT operating is used in just 6 rounds. Each round uses simple mathematical operations. This causes uncertainty for attackers and complicates the encryption process even more. The cipher supports key size of 80-bit for size data of 64-bit. It works with the architecture of SPN and Feistel to gain strong security [28].

III. OVERVIEW OF NLBSIT

The algorithm proposed was applied using basic logical operations such as XOR, Ex-NOR operations, concatenation. Using these operations, the data / information is encrypted. Therefore, only the user or authorized person can open the encryption of data / information easily. Because the key aims to build system of a lightweight encryption, the suggested approach addresses 64-bit data block for 64-bit key-length. The security is high in the NLBSIT algorithm, due to the use of both the SP and Feistel architectures in the construction of the NLBSIT. The algorithm uses uncomplicated mathematical operations with fewer rounds and has the advantages of both SP and Feistel architectures. Therefore, the algorithm achieves high security as well as high speed, uses less memory and less energy consumption for devices with limited resources.

A. Expansion of The Key

The key is the algorithm's principal component (encryption / decryption). For security the size of the encryption key is very important. Thus (key size) becomes a major obstacle for the attacker to be known. The strength of key generation results in increased security, better encryption complexity, and decreased knowledge of the key by the attackers, through the processes of diffusion and confusion that generate the proposed algorithm. The key expansion is required 64-bit length. The key expansion shown in Fig. 1. And illustrated in Algorithm 1.

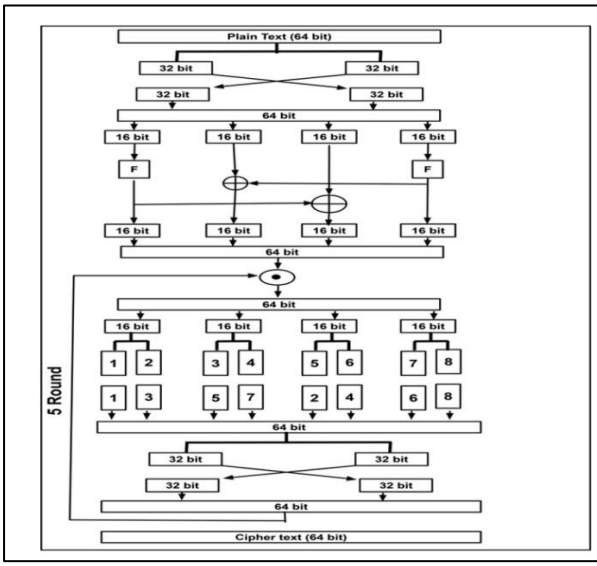


Algorithm 1. Key Expansion

Input: 64 bits key (KT)
Output: 64 bits key (KC)

1. Divided into 4 group, each group 16 bits
   A1= KT[1] || KT[5] || KT[9] || KT[13]
   A2= KT[2] || KT[6] || KT[10] || KT[14]
   A3= KT[3] || KT[7] || KT[11] || KT[15]
   A4= KT[4] || KT[8] || KT[12] || KT[16]
2. Get F-Function For each A1...A4
   Af1= F-Function(A1)
   Af2= F-Function(A2)
   Af3= F-Function(A3)
   Af4= F-Function(A4)
3. Generate AfK array
   Afk1= Af1 ʘ Af2
   Afk2= Afk1 ⊕ Af2
   Afk3= Afk2 ʘ Af3
   Afk4= Afk3 ⊕ Af4
4. Generate Final Key(KC)
   KC= Afk1|| Afk2 || Afk3 || Afk4
5. **end**

Fig. 1: Key Expansion of NLBSIT



Algorithm 2. NLBSIT **Encryption**

Input: 64-bits plaintext (PT) and 64-bits key (KC)
Output: 64 ciphertext (CT)

1. read the byte values, from the input file called plaintext (PT)
2. Divided the plaintext (PT) into two halves and replaced them together such as,
   A1=PT [1:32]; A2=PT [33:64]
   CT=A1|| A2
3. Divided PT into 4 group, each group 16 bits such as,
   CT1 =CT[1:16]; CT2=CT[17:32]
   CT3 =CT[33:48]; CT1=CT[49:64]
4. Get F-Function for all 16 bits of CT such as,
   CTT1= F-Function (CT1) ; CTT4= F-Function (CT4)
   CTT2= CTT4 ⊕ PT2 ; CTT3= CTT1 ⊕ PT3
5. merge CTT1 , CTT2 , CTT3 , and CTT4 into CT 64 bits such as,
   CT= CTT1 || CTT2 || CTT3 || CTT4
6. Initializer R with value 1
7. EX-NOR between ciphertext and key
   CT= CT ʘ KC
8. Split the block, each 8-bit partition, then switch between them such as,
   CT1 =CT[1:8] || CT[17:24] || CT[33:40] || CT[49:56]
   CT2 =CT[9:16] || CT[25:32] || CT[41:48] || CT[57:64]
9. merge CT1 And CT2 into CT64 bits such as ,
   CT=CT2 || CT1
   R = R +1
10. If (R <5)
    Go to Step 6
11. Else
    Go to step 12
12. Generate Final Cipher
    CT

Fig. 2: Encryption Process of NLBSIT

B. Strategy of encryption

The mechanism to turn the data / images from comprehensible form to incomprehensible form using diffusion and confusion techniques called encryption. The mechanism to turn the data / images from non-understandable to comprehensible form for users with authority to do so is called decryption. The proposed NLBSIT algorithm is illustrated in Fig. 2 and explained in Algorithm 2. The algorithm works with five rounds to reduce energy use in devices. Each round uses simple mathematical operations. The output of each round is considered an input to the next round to obtain the ciphertext (CT).

C. F-function

The F-Function strategy consists of many linear and nonlinear processes that ensure dynamic dependence of output bits on input bits [8], [29], [30]. This process also called confusion and diffusion depend on value P and Q that seen in Fig 3 and illustrated in Fig 4 and explained in the algorithm 3.

K_{Ci}	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P(K_{Ci})	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1
Q(K_{Ci})	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

Fig. 3 P and Q Table.

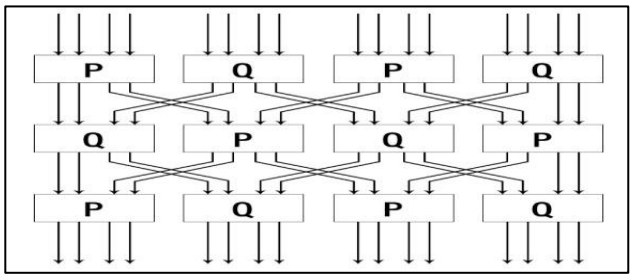


Fig. 4: F-Function [5].

Algorithm 3. F-Function

Input: 16 bits key (KFT), Four noes of 4 bit (C1,C2,C3,C4)

Output: 16 bits key (KFC)

1. **Get Frist Level From F-Function**
   SC1= SBOXP[C1]
   SC2= SBOXQ[C2]
   SC3= SBOXP[C3]
   SC4= SBOXQ[C4]
2. **Get Second Level F-Function**
   SC1= SBOXQ[SC1]
   SC2= SBOXP[SC2]
   SC3= SBOXQ[SC3]
   SC4= SBOXP[SC4]
3. **Get Last Level F-Function**
   SC1= SBOXP[SC1]
   SC2= SBOXQ[SC2]
   SC3= SBOXP[SC3]
   SC4= SBOXQ[SC2]
4. **Generate (KFC)**
   KFC= SC1|| SC2 || SC3 || SC4
5. **end**

D. Strategy of decryption

Having a good decryption process when developing an effective encryption method is undeniably important. The decryption method of the proposed algorithm was built in this study as being essentially the same as the encryption procedure but in the reverse order.

IV. SIMULATION AND IMPLEMENTATION OF ALGORITHM

The proposed NLBSIT algorithm is a lightweight encryption/decryption for IoT devices. The encryption algorithm was built in the 1st phase and is implemented in the Dev-C++ language by compiler C. It is a completely functioning, free, integrated programming development environment licensed under the GNU General Public License. The 2nd step converts the algorithm form C language format to FELICS (Fair Evaluation of Lightweight Cryptographic Systems) and ensure from result C and FELICS same result (plaintext and ciphertext). The FELICS tool measures LWC from some parameters such as execution cycles (encryption/decryption), RAM usage, and program size [31]. It works on operating system Ubuntu Linux and implements various standard and popular forms of LWCs, such as PICCOL, TWIN, AES, HIGHT and others, and compares them with any new algorithms, and its uses different performance evaluation tools (such as AVR, ARM, and MSP).

V. SECURITY ANALYSIS

The security quality of the suggested solution is measured by three forms of analysis, namely: analysis of the attack in section A, the evaluation parameters in section B, and statistical analysis in section C.

A. Attack Analysis

- **Linear and Differential Cryptanalysis:** Linear and differential attacks are completely unsuccessful in full encryption. That is, current f -function does the same work in [8], [29], and [30]. If the linear approximation is performed for two rounds, the input and output association is very high. The circular transformation is often kept uniform, handling any bit in a comparable way and opposing differential attacks.
- **Weak Key Attacks:** Encryption is considered strong if the algorithm produces strong keys for encryption. The encryption key is generated from the actual key. XORing 's actual key defends this attack before its use. The same is used by SIT [30] and cipher in [8], which has proved protection against weak keys. In this sense, the algorithm uses F-Function and after that we use the operations (XNOR, XOR) to increase the complexity (confusion and diffusion). Therefore, the algorithm thwarts this type of attack.
- **Related Keys Attacks:** The attacker guesses several keys to recognize the encrypted text. These attacks are known as related attacks [32]. The attacker's target is to

search for the real hidden keys. Symmetry in the key expansion process and the slowdown in the encryption process are one of the reasons that lead to this type of attack. The proposed algorithm for the main expansion mechanism is intended to be immune to this form of attack because it has high diffusion and non-linearity.

- **Square Attack:** Square attack is presented in [8],[29]. In the last round of the key, the intruder could acquire the last byte. Moreover, the attack is repeated eight times in order to acquire the remainder of the key. key guessing from the attacker takes 2^8 by 2^8 selected plaintexts= 2^{16} S-box lookups to get one byte.

B. Evaluation Parameters

- **Key Sensitive:** For encryption a key-sensitive algorithm needs to be used. This suggests that the algorithm doesn't restore the original data if the key already has a minute difference from the original version. In this scenario, the evaluation is deemed ideal if 50 percent of the bits are modified due to one bit of change, according to the Strict Avalanche Criteria SAC [33]. In order to visually detect this effect, we decrypt the image using a key that differs from the correct one by a single bit.
- **Execution Cycle:** The most significant variable for measuring the algorithm's output is the period of time it takes to encrypt and decrypt the given data. The proposed algorithm is designed to require minimum time consumption and comprehensive protection for the IoT environment.
- **Memory Utilization:** Memory capacity is a key problem in IoT devices which restricts resources. Complex arithmetic operations and an increase in the number of rounds take the greatest attention when designing a lightweight encryption algorithm for IoT devices. This is because these operations require large memory and consume the hardware power. Thus, the proposed algorithm uses fewer rounds and simple mathematical operations. Therefore, it requires less energy consumption and little memory for processing.

C. Statistical Analysis

- **Histograms analysis:** Image histogram analysis involves a graphic portrait of pixel intensity values and illustrates the tonal distribution of an image [34]. In brief, it gives statistical characteristics of picture strength from which the picture can be noticeable [35]. The key purpose of the study of the histogram is to demonstrate the properties of the ciphered data confusions and diffusions. Nonetheless, a histogram can measure randomness when encrypting a file. A cryptographic algorithm refers to adequate protection, unless the calculated histogram is compatible after encryption.
- **Information entropy analysis:** The encryption algorithm adds additional information to the data, such as the difference between the original information and the algorithm implemented, which is difficult for the attacker. The entropy of image is a quantity that is used to characterize the amount of data that an encryption algorithm needs to decrypt the extra information. Therefore, the higher the entropy of image, the higher the security of the algorithm. The maximum entropy of an 8-bit grey scale image can be 8 bits. Its express of formula (high entropy) by equation is as below:

$$Entropy(H) = \sum_{i=0}^{255} P(x_i) \log_2 P(x_i) \qquad (1)$$

The probability that the variance between two neighbouring pixels is equal to i is P (x_i).

- **Correlation analysis:** The Correlation is an efficient method of determining a cryptography algorithm's power. However, the correlation between two values refers to the dependency. In general, there is no correlation between the original image and the encoded image. Therefore, the encoded image is random and is not related to the original image [36]. Using the following equations, correlation coefficients for initial and encrypted messages are determined.

$$CorrCoef = \frac{cov(x,y)}{\sqrt{Var}(x) \times \sqrt{Var(y)}} \qquad (2)$$

$$Var(x) = \frac{1}{N}\sum_{i=1}^{N}[(x_i - E(x))^2] \qquad (3)$$

$$Cov = \frac{1}{N}[(x_i - E(x)) \times (y_i - E(y))] \qquad (4)$$

Where, the correlation coefficient is $CorrCoef$ and the covariance of $Cov(x,y)$ is pixel x and y. $Var(x)$ is the pixel value variance in an image, $E(x)$ is the value operator predicted and N is the total number of pixels in the matrix.

VI. FUNCTIONALITY ANALYSIS

Figure .5 is a test and verification of the suggested algorithm in the FELICS tool. A short overview of some well-known lightweight cryptographic algorithms with their characteristics and performance measurements is presented in Table 2, and implementation in the FELICS tools and platform device is AVR. Based on parameters such as code size, RAM footprint, and the execution times (encryption/decryption), the functional comparative analysis of the other algorithms with the proposed NLBSIT was drawn up. In the bar graph in Figure 6, the proposed algorithm is shown to use less memory, and take fewer cycles in encoding and decoding. The red color characterizes the proposed algorithm in Figures (7, 8, 9) to illustrate the comparison in memory, and cycles in encoding and decoding.

The strength of the algorithm is measured by using the avalanche test of the proposed algorithm. We used two images named p643_l_s2(Hand1) and p709_l_s3(Hand2)

from the database [37]. As in Figure 10, the avalanche experiment of the algorithm means that a single bit shift in the key, the plaintext, produces around 49 percent changes in cipher bits. If even one bit is modified in the original keys, the decryption is not identifiable.

Additionally, on each histogram test, the vertical lines represent the number of pixels and the horizontal lines represent the pixel intensity. Uniform distribution of intensities reveals ideal protection after encryption. As shown in Figure 11. Finally, the correlation test in Figure 12. Illustrates the contrast between encrypted data and original data. The initial image shows strongly correlated meaning while the encrypted image appears to have marginal correlated meaning. Therefore, less correlation provides better protection.

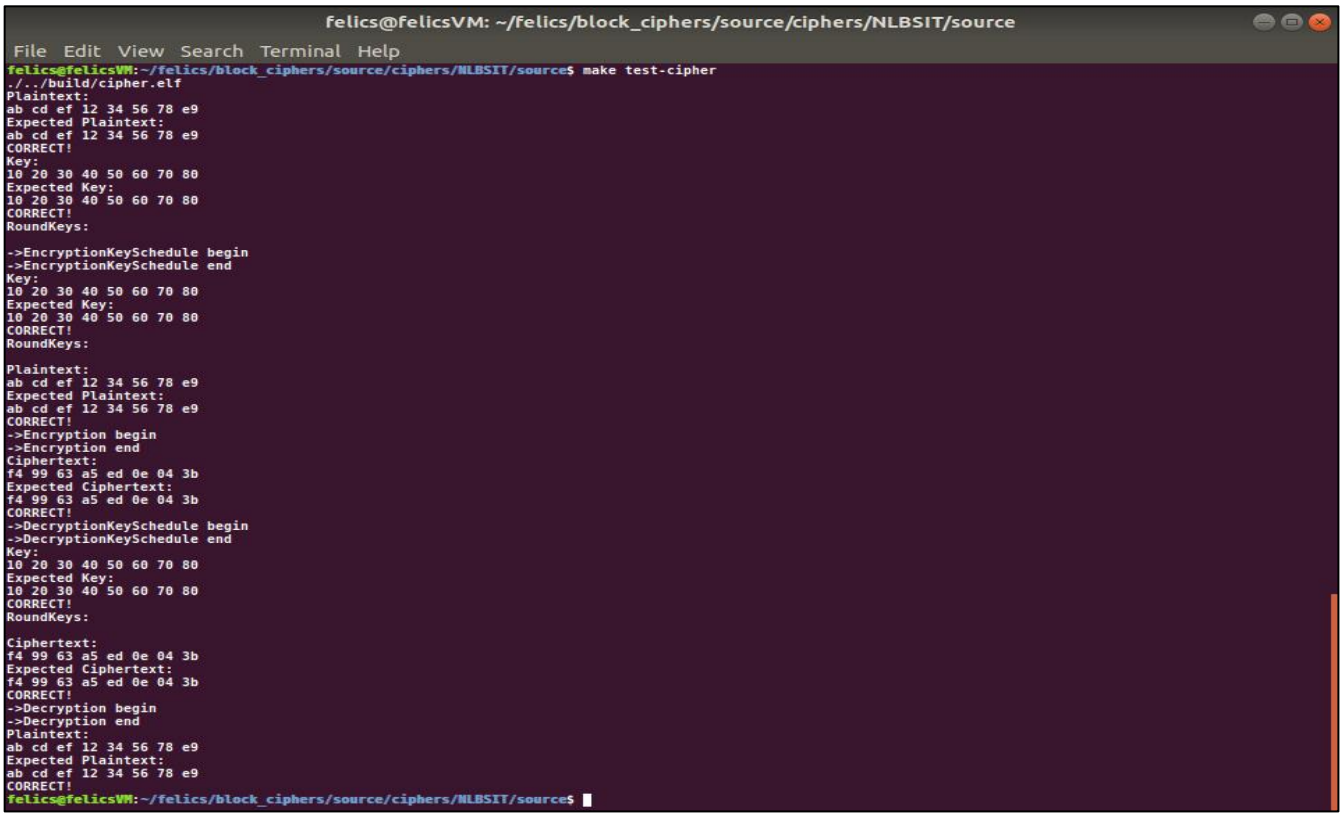
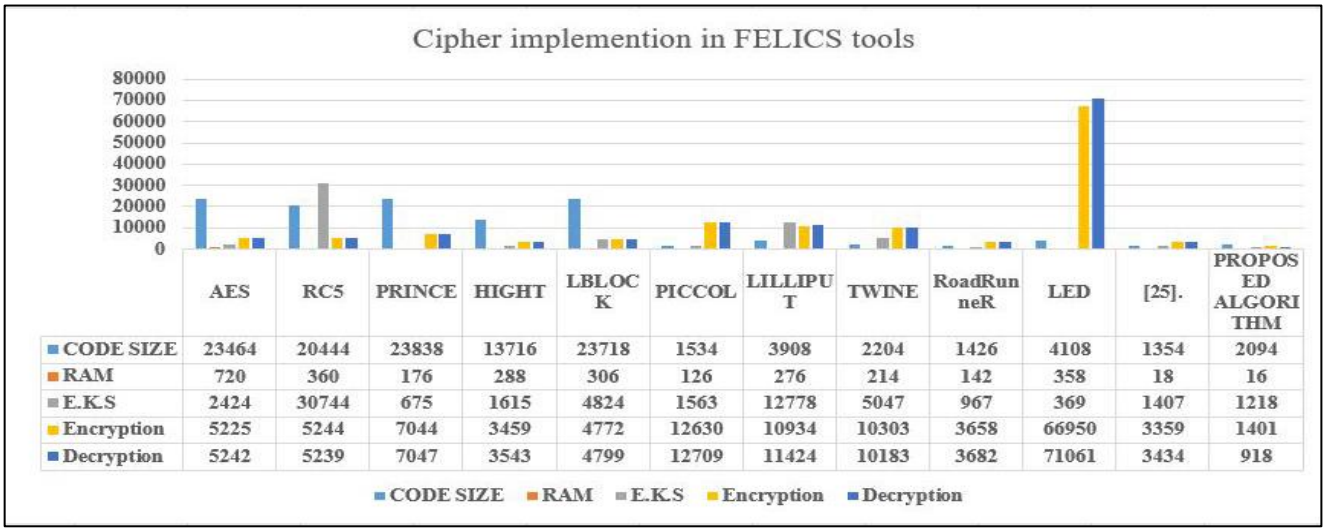


Fig. 5: Implementation Test of NLBSIT on FELICS.

TABLE 1: Results for NLBSIT Cipher with common cipher Implementations on AVR Architecture in FELICS tools.

Cipher	Device	Block Size (bit)	Key Size (bit)	Code Size (byte)	RAM (byte)	Encrypted-Key Schedule	Encryption (cycles)	Decryption (cycles)
AES	AVR	128	128	23464	720	2424	5225	5242
RC5	AVR	64	128	20444	360	30744	5244	5239
PRINCE	AVR	64	128	23838	176	675	7044	7047
HIGHT	AVR	64	128	13716	288	1615	3459	3543
LBLOCK	AVR	64	80	23718	306	4824	4772	4799
PICCOL	AVR	64	80	1534	126	1563	12630	12709
LILLIPUT	AVR	64	80	3908	276	12778	10934	11424
TWINE	AVR	64	80	2204	214	5047	10303	10183
RoadRunneR	AVR	64	80	1426	142	967	3658	3682
LED	AVR	64	80	4108	358	369	66950	71061
[28]	AVR	64	80	1354	18	1407	3359	3434
PROPOSED ALGORITHM	**AVR**	**64**	**64**	**2094**	**16**	**1218**	**1401**	**918**



	AES	RC5	PRINCE	HIGHT	LBLOCK	PICCOL	LILLIPUT	TWINE	RoadRunneR	LED	[25].	PROPOSED ALGORITHM
CODE SIZE	23464	20444	23838	13716	23718	1534	3908	2204	1426	4108	1354	2094
RAM	720	360	176	288	306	126	276	214	142	358	18	16
E.K.S	2424	30744	675	1615	4824	1563	12778	5047	967	369	1407	1218
Encryption	5225	5244	7044	3459	4772	12630	10934	10303	3658	66950	3359	1401
Decryption	5242	5239	7047	3543	4799	12709	11424	10183	3682	71061	3434	918

Fig. 6: Comparison analysis in terms of Code Size, RAM, and Encryption/Decryption (Cycles).

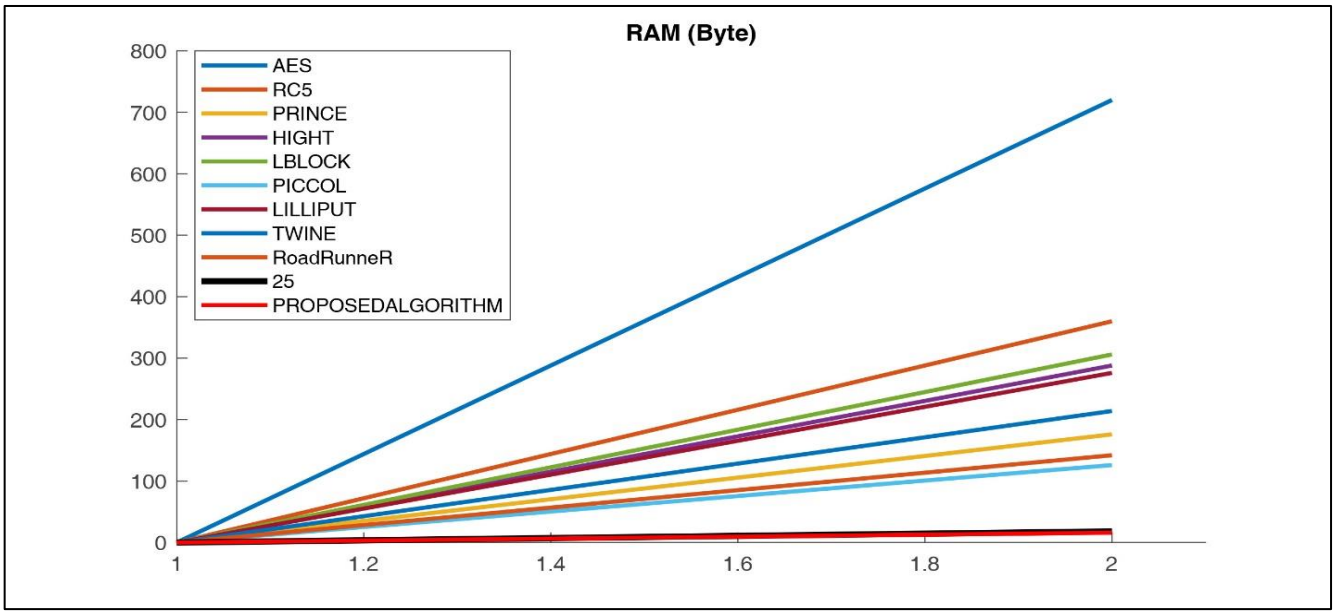


Fig. 7: Curve of RAM in byte for different block implementation.

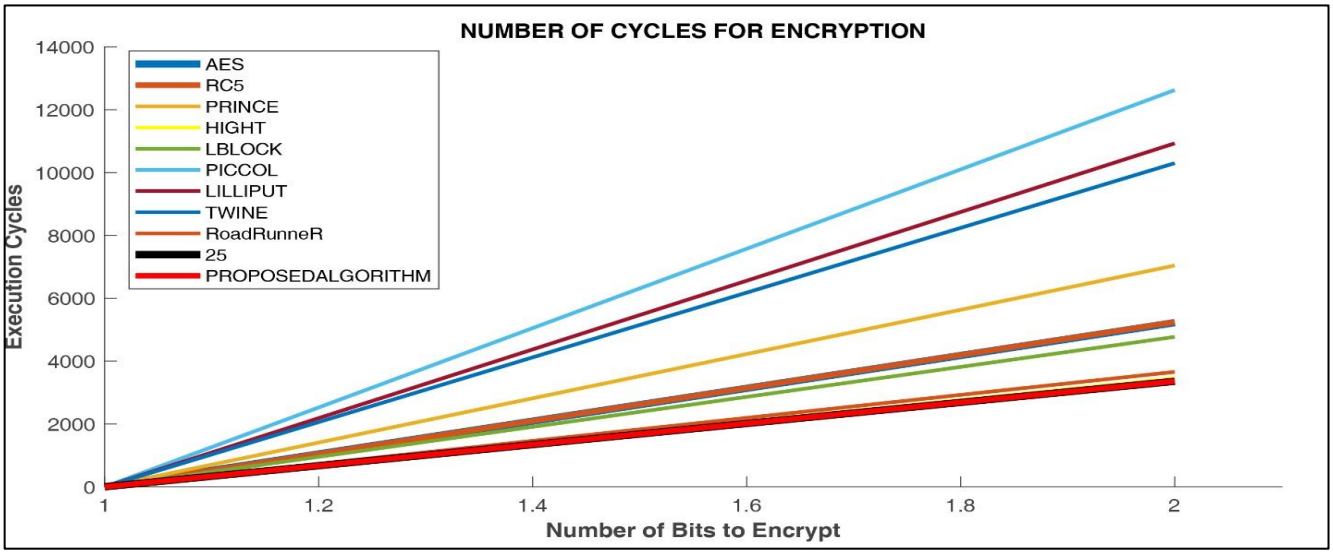


Fig.8: Curve of execution time (cycle) for multiple ciphers in varying block for encryption.

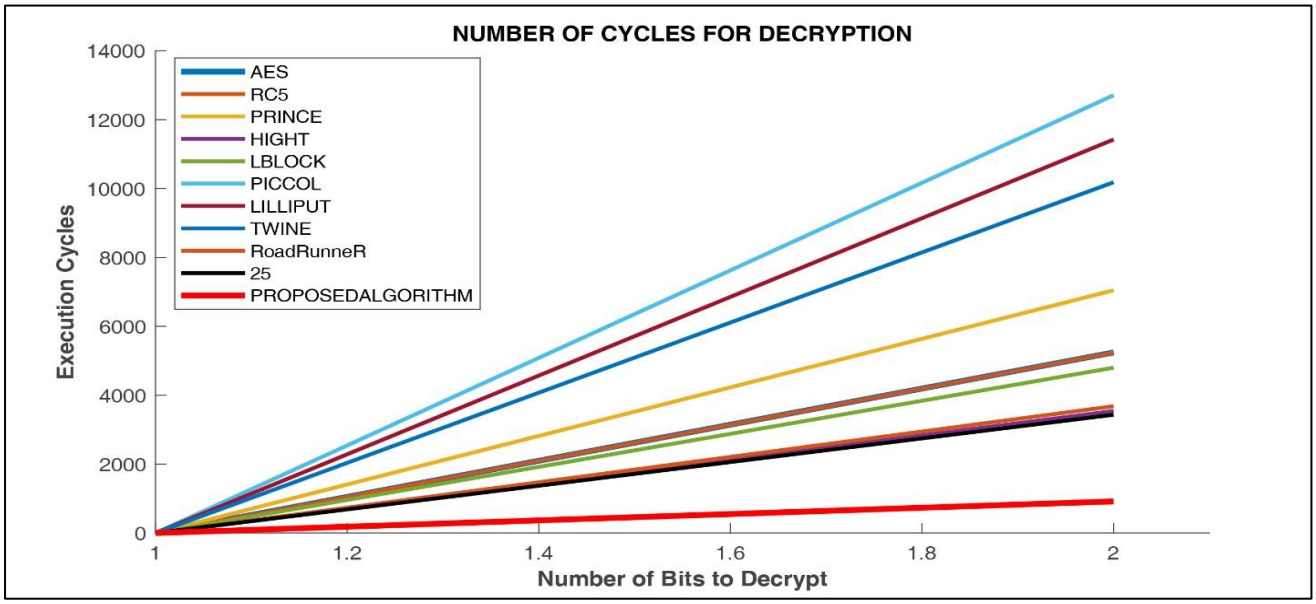


Fig.9: Curve of execution time (cycle) for multiple ciphers in varying block for decryption.

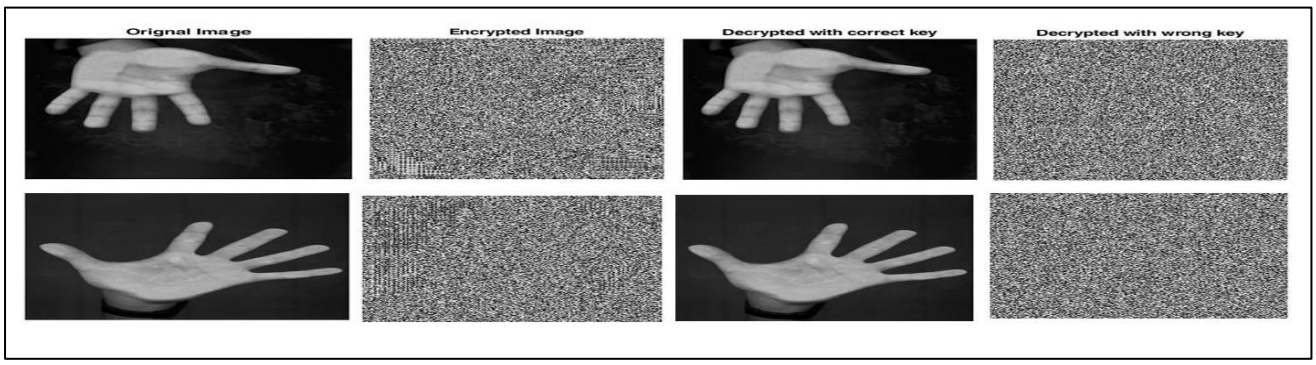


Fig.10: Key Sensitivity of hand image analysis.

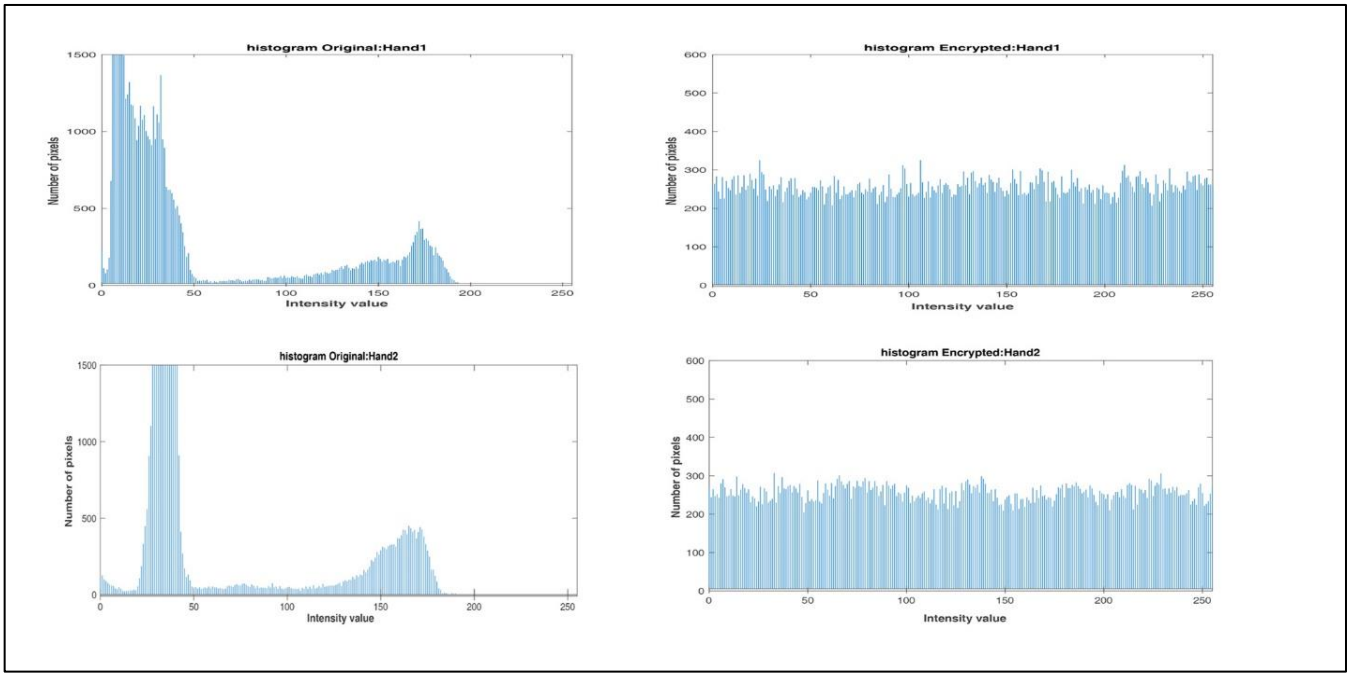


Fig.11: Analysis of hand image histogram

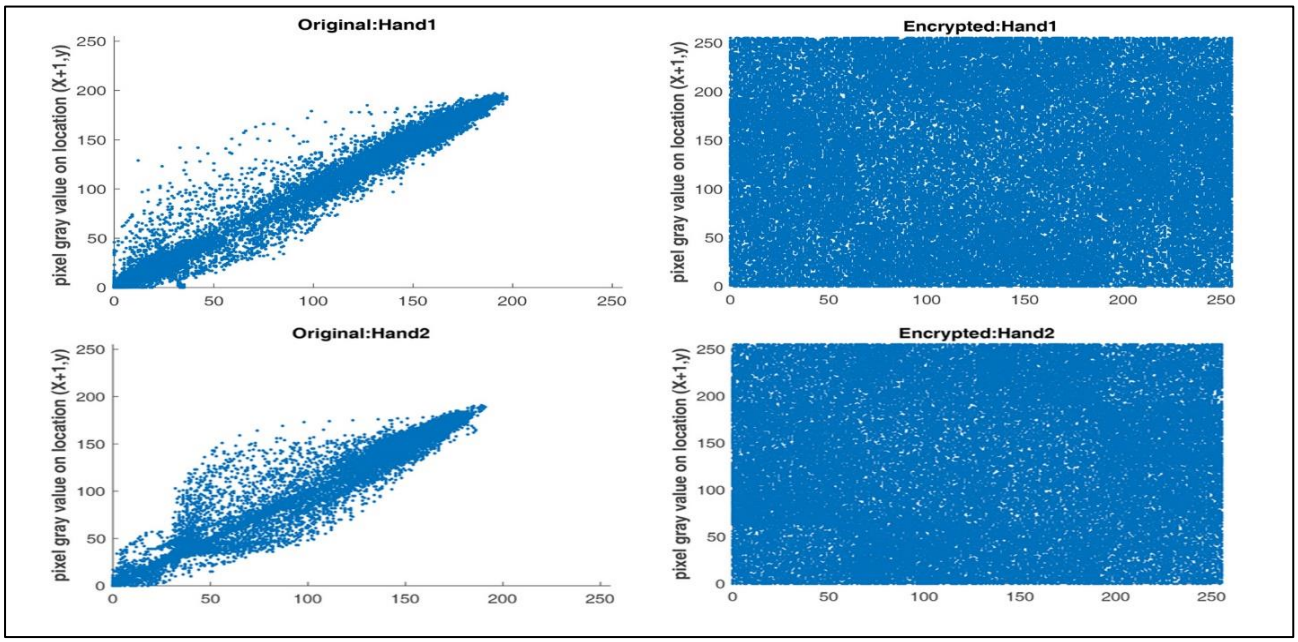


Fig.12: Hand image of correlations for encrypted/decrypted.

VII. CONCLUSION

For resource restricted IoT computers, this work designs and implements a new LW block cipher called NLBSIT. By combining the advantages of both feistel and SPN architectures along with an additional linear box concept, the proposed cipher improves data protection. Also, the avalanche, Histogram, and Entropy tests were performed to measure the strength of this algorithm. The NLBSIT algorithm consumes less energy than other algorithms (encryption / decoding cycles), and less memory consumption. This makes it a component for encryption in IoT devices.

REFERENCES

[1] Prajakta P. Deshpande, "IoT Based Fleet Management Systems : A Review", International Journal of Computer Sciences and Engineering, Vol.7, Issue.5, pp.436-443, 2019.

[2] Hui, T. K., Sherratt, R. S., & Sánchez, D. D. (2017). Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies. Future Generation Computer Systems, 76, 358-369.

[3] Anjum Sheikh, "IoT Based Security System for Smart Homes", International Journal of Computer Sciences and Engineering, Vol.07, Special Issue.11, pp.35-38, 2019.

[4] Zhou, G., Liu, Z., Shu, W., Bao, T., Mao, L., & Wu, D. (2017). Smart savings on private car pooling based on internet of vehicles. Journal of Intelligent & Fuzzy Systems, 32(5), 3785-3796.

[5] Majumdar, A., Debnath, T., Sood, S. K., & Baishnab, K. L. (2018). Kyasanur forest disease classification framework using novel extremal optimization tuned neural network in fog computing environment. Journal of medical systems, 42(10), 187.

[6] Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., & Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers, 24(6), 522-533.

[7] Al-ahdal, A. H., & Deshmukh, N. K. A Systematic Technical Survey Of Lightweight Cryptography On lot Environment.

[8] Al-ahdal, Abdulrazzaq HA, and Nilesh K. Deshmukh, and Galal A. AL-Rummana. "A Robust Lightweight Algorithm for Securing Data in Internet of Things Networks." .Under Publication.

[9] Biswas, A., Majumdar, A., Nath, S., Dutta, A., & Baishnab, K. L. (2020). LRBC: a lightweight block cipher design for resource constrained IoT devices. Journal of Ambient Intelligence and Humanized Computing, 1-15.

[10] Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Journal of Ambient Intelligence and Humanized Computing, 1-18.

[11] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., & Iwata, T. (2007, March). The 128-bit blockcipher CLEFIA. In International workshop on fast software encryption (pp. 181-195). Springer, Berlin, Heidelberg.

[12] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., ... & Vikkelsoe, C. (2007, September). PRESENT: An ultra-lightweight block cipher. In International workshop on cryptographic hardware and embedded systems (pp. 450-466). Springer, Berlin, Heidelberg.

[13] Guo, J., Peyrin, T., Poschmann, A., & Robshaw, M. (2011, September). The LED block cipher. In International workshop on cryptographic hardware and embedded systems (pp. 326-341). Springer, Berlin, Heidelberg.

[14] De Canniere, C., Dunkelman, O., & Knežević, M. (2009, September). KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 272-288). Springer, Berlin, Heidelberg.

[15] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., & Shirai, T. (2011, September). Piccolo: an ultra-lightweight blockcipher. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 342-357). Springer, Berlin, Heidelberg.

[16] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knudsen, L. R., Le, G., ... & Rombouts, P. (2012). PRINCE–A Low-latency Block Cipher for Pervasive Computing Applications Full version.

[17] Albrecht, M. R., Driessen, B., Kavun, E. B., Leander, G., Paar, C., & Yalçın, T. (2014, August). Block ciphers–focus on the linear layer (feat. PRIDE). In Annual Cryptology Conference (pp. 57-76). Springer, Berlin, Heidelberg.

[18] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015, June). The SIMON and SPECK lightweight block ciphers. In Proceedings of the 52nd Annual Design Automation Conference (pp. 1-6).

[19] Wheeler, D. J., & Needham, R. M. (1994, December). TEA, a tiny encryption algorithm. In International Workshop on Fast Software Encryption (pp. 363-366). Springer, Berlin, Heidelberg.

[20] Hong, D., Lee, J. K., Kim, D. C., Kwon, D., Ryu, K. H., & Lee, D. G. (2013, August). LEA: A 128-bit block cipher for fast encryption

on common processors. In International Workshop on Information Security Applications (pp. 3-27). Springer, Cham.
[21] Koo, B., Roh, D., Kim, H., Jung, Y., Lee, D. G., & Kwon, D. (2017, November). CHAM: a family of lightweight block ciphers for resource-constrained devices. In International Conference on Information Security and Cryptology (pp. 3-25). Springer, Cham.
[22] Bansod, G., Pisharoty, N., & Patil, A. (2017). BORON: an ultra-lightweight and low power encryption design for pervasive computing. Frontiers of Information Technology & Electronic Engineering, 18(3), 317-331.
[23] Patil, J., Bansod, G., & Kant, K. S. (2017, February). LiCi: A new ultra-lightweight block cipher. In 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI) (pp. 40-45). IEEE.
[24] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., & Verbauwhede, I. (2015). RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences, 58(12), 1-15.
[25] Karakoç, F., Demirci, H., & Harmancı, A. E. (2013, May). ITUbee: a software oriented lightweight block cipher. In International Workshop on Lightweight Cryptography for Security and Privacy (pp. 16-27). Springer, Berlin, Heidelberg.
[26] Yang, G., Zhu, B., Suder, V., Aagaard, M. D., & Gong, G. (2015, September). The simeck family of lightweight block ciphers. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 307-329). Springer, Berlin, Heidelberg.
[27] Guo, J., Peyrin, T., Poschmann, A., & Robshaw, M. (2011). The LED block cipher. In Cryptographic Hardware and Embedded Systems–CHES 2011 (pp. 326-341). Springer Berlin Heidelberg.
[28] A. H. A. Al-Ahdal, G. A. AL-Rummana, G. N. Shinde, and K. D. Nilesh, "Security Analysis of a Robust Lightweight Algorithm for Securing Data in Internet of Things Networks," .Under Publication.
[29] Barreto, P. S. L. M., & Rijmen, V. (2000). The Khazad legacy-level block cipher. Primitive submitted to NESSIE, 97, 106.
[30] Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. SIT: a lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688. 2017 Apr 27.
[31] D. Dinu, A. Biryukov, J. Großschädl, D. Khovratovich, Y. L. Corre, L. Perrin, "FELICS – Fair Evaluation of Lightweight Cryptographic Systems", University of Luxembourg, July 2015.
[32] Biham, E. (1994). New types of cryptanalytic attacks using related keys. Journal of Cryptology, 7(4), 229-246.
[33] A. Webster and S. E. Tavares, "On the design of s-boxes," in Conference on the Theory and Application of Cryptographic Techniques. Springer, 1985, pp. 523–534.
[34] Kanso, A., & Ghebleh, M. (2018). An efficient lossless secret sharing scheme for medical images. Journal of Visual Communication and Image Representation, 56, 245-255.
[35] Hodeish, M. E., Bukauskas, L., &Humbe, V. T. (2019). A new efficient TKHC-based image sharing scheme over unsecured channel. Journal of King Saud University-Computer and Information Sciences.
[36] Ahmad, M., Doja, M. N., & Beg, M. S. (2018). Security analysis and enhancements of an image cryptosystem based on hyperchaotic system. Journal of King Saud University-Computer and Information Science.
[37] Magalh˜aes, F., Oliveira, H. P., Matos, H., Campilho, A.: HGC 2011 - Hand Geometric Points Detection Competition Database, http://www.fe.up.pt/~hgc2011/.

AUTHORS PROFILE

Abdulrazzaq H. A. Al-Ahdal

Currently, he is a Ph.D candidate in the School of Computational Sciences, at Swami Ramanand Teerth Marathwada University, Nanded. He has received his M.Sc. degree in Computer Networking from the School of Computational Sciences at Swami Ramanand Teerth Marathwada University, in 2016, Nanded, India. He has received his B.E degree in Computer Science from the faculty of Computer Science & Engineering, at Hodeidah University, Yemen, 2004. He was worked as a lecturer until 2014 .He is currently working on Development Lightweight Cryptographic schema for IoT.

Galal A. AL-Rummana

Currently, he is a Ph.D candidate in the School of Computational Sciences, at Swami Ramanand Teerth Marathwada University, Nanded. He has received his M.Sc. degree in Computer Networking from the School of Computational Sciences at Swami Ramanand Teerth Marathwada University, in 2017, Nanded, India. He has received his B.E degree in Computer Engineering form the faculty of Computer Science & Engineering , at Hodeidah University, Yemen, 2014. He is currently working on Big Data Security.

G. N. Shinde

is working as Principal, Yeshwant College, Nanded (India). Earlier he was Pro-Vice Chancellor, SRTM University, Nanded, Maharashtra, INDIA. He has received "Ideal State Teacher Award" from Government of Maharashtra, India for 2008-09 and "Best Principal Award" for 2009-2010 from S.R.T.M. University, Nanded, Maharashtra. He has received M. Sc. & Ph.D. degree from Dr. B.A.M. University, Aurangabad. He has awarded Benjonji Jalnawala award for securing highest marks at B.Sc. Seventeen research scholars were awarded Ph.D. degree under his guidance. He has published more than 90 papers in the International Journals and presented more than 50 papers in International Conferences. He was more than the five times Chairperson for International Conference in abroad. In his account one book is published, which is reference book for different courses in different Universities. He is also member of different academic & professional bodies such as IAENG (Hon Kong), ANAS (Jordan). He is in reviewer panel for different Journals such as IEEE (Transactions on Neural Networks), International Journal of Physical Sciences (U.S.A.), Journal of Electromagnetic Waves and Applications (JEMWA, U.S.A.). His abroad Visit includes U.S.A., Thailand, Portugal, Germany, Switzerland, Italy, Vatican City, Monaco, France, Maldives, Sri Lanka, U. K., Scotland, China , New Zealand and Hong Kong, Singapore. His research interest includes Filters, Wireless Sensor Network System, Image processing and Multimedia analysis and retrieval system and Data mining.

Dr. Nilesh K. Deshmukh

In 2001, he joined the School of Computational Sciences, Swami Ramanand Teerth Marathwada University, Nanded as a Lecturer. Since June 2001, he has been with the School, where he was an Assistant Professor. His current research interests include Geo-informatics, Data Mining and Data Analytics. Under his guidance 08 Students awarded Ph.d. Degree and 10 students awarded M.Phil. Degree.

