Provably-Secure LED Block Cipher Diffusion and Confusion Based on Chaotic Maps

Hussain M. Al-Saadi[1], Imad S. Alshawi*[2]
Department of Computer Science, College of Computer Science and Information Technology,
University of Basrah, Basrah, Iraq
E-mail: hussain.mk1978@gmail.com[1], emadalshawi@gmail.com[2]

Keywords: lightweight block cipher, LED block cipher, chaos, Lorenz system, security, NIST tests

Received: January 4, 2023

Lightweight cryptography algorithms have concentrated on key generation's randomness, unpredictable nature, and complexity to improve the resistance of ciphers. Therefore, the key is an essential component of every cryptography algorithm since it affects the algorithm's level of security. Light Encryption Device (LED) is a high-performance, lightweight block encryption solution that works on resource-constrained devices and considers a lighter version of AES. It employs a 64-bit block cipher with two significant instances using 64-bit and 128-bit keys, respectively. A lack of key scheduling in LED heightens security risks, such as key-related attacks. Specifically, now that LED has been hacked and is no longer secure. Therefore, LED must achieve a high diffusion and confusion level to withstand known attacks. Chaos-based encryption provides an exceptionally high level of security because of the unique characteristics of chaotic systems, which are defined by various nonlinear deterministic dynamic equations. Merge LED algorithm and the advantages of chaotic maps randomness provide successful confusion and diffusion property to improve the LED algorithm's shortcomings by increasing its security. This paper presents a lightweight approach to construct a robust, sufficiently using 3-D Lorenz system chaotic map to generate a one-time pseudo-random bit key to avoid being predicted by adversaries, resulting in achieving sound confusion and diffusion and withstand known assaults. A NIST test suite found that the performance of the LED based on the 3-D Lorenz chaotic map approach in terms of data secrecy is nearly 0.3003 higher than that of the LED and keeps the trade-off between computation cost and security.

Povzetek: Predstavljena je izgradnja robustnega in izvedbeno ugodnega 3-D Lorenzovega sistema za generiranje psevdo-naključnega ključa.

1 Introduction

Information security protects data against unauthorized access, detection, modification, or destruction with upholding confidentiality, integrity, and availability (CIA) [1]-[2]. Cryptography protects data in transit (either electronically or physically) through networks. Thus, it is necessary to use current cryptography techniques to fend off security risks [1], [2]-[3]. In resource-constrained systems, conventional cryptography algorithms are extremely slow, complex, and energy-intensive [4]-[5]. The use of low-cost computational algorithms is growing in popularity. Symmetric and asymmetric lightweight cryptography algorithms are classified into two categories.

The symmetric encryption algorithm encrypts and decrypts data using the same secret key. While asymmetric key encryption, data is encrypted with a public key and decrypted with a private key. Block cipher and stream cipher are two distinct symmetric key encryption methods. Trivium, Grain, and Salsa20 are examples of stream ciphers, while Present, LED, RECTANGLE, and HIGHT are examples of block ciphers [6]-[7]. In block cipher, encryption and decryption occur concurrently on a block of a defined size (64 bits or more). In contrast, stream cipher continuously processes the input information bit by bit (or word by word). Claude Shannon suggested confusion and diffusion as crucial aspects of any cryptography [8]-[9] to strengthen the cipher. Stream ciphers rely primarily on the confusion property, but block ciphers combine confusion and diffusion more straightforwardly than stream ciphers [7]. Except for LED, most ciphers, like AES, SPECK, TWINE, PRESENT, and SIMON, require key scheduling, in which actions are performed on the initial secret key to improve the cipher's security. Each round produces a unique round key. The round keys can be made outside the cipher and downloaded at runtime, or the cipher can make them before it starts and save them in memory or make them "on the fly." [10]. The LED method has a minimum block size of 64 bits, a low hardware cost, and a greater frequency than the AES block cipher [8]-[11]. Therefore, LED is the optimal choice if any application requires the smallest area and the quickest time for encryption and decryption [11].

LED is no longer as secure as the current cryptanalysis techniques. Biclique attack is a technique of meet-in-the-middle cryptanalysis applied to the most common lightweight block ciphers, LED, Piccolo, and PRESENT [12], resulting in slow and limited diffusion of the key schedule and encryption process. The attacks could recover the secret key of target algorithms with less computational complexity than an exhaustive search [12]-[13]. LED must handle this issue to acquire good dissemination and resist known attacks such as related key attacks, side-channel attacks, and meet-in-the-middle attacks [14]-[15]. So, a chaotic system or computational intelligence (CI) is the ideal answer. Several techniques have been invented for chaotic systems [16]**.** The application of chaos theory, a nonlinear system, in cryptography has recently been made to address issues with existing encryption techniques, which are losing their ability to provide quick and secure encryption for large amounts of data simultaneously [17]. Chaotic systems are unexpected over the long term because of their unique characteristics and high sensitivity to their initial states, which allow for a wide range of chaotic sequences. The resulting chaotic patterns are neither periodic nor concurrent [17]-[18].

There have been a lot of real-world systems in recent years, including the Internet of Things (IoT), wireless sensor networks (WSNs), smart cities, etc. Information security has faced significant challenges due to the complexity and increasing data. Considerable computational intelligence (CI) approaches have been created to tackle these difficulties [19]-[20] that are challenging to solve manually. CI has been employed to address many information security problems, such as selecting the optimum solution and determining normal and abnormal behavior in systems like data concealing and intrusion detection [21]-[22]. However, a specific computation intelligence technique cannot address all information security challenges. Thus, numerous computation intelligence techniques and applications with chaos theory have been implemented for information security [23].

Recall that LED as a block cipher is weak in security and has been broken [7], [12]-[14]. Therefore, LED must obtain a high diffusion and confusion level to resist known attacks. Consequently, we use Lorenz 3-D chaotic map because of its unpredictable nature and complexity to generate a random key that XORed twice with the LED block cipher state during the encryption process to encrypt the 64-bit block of the data to produce the ciphertext.

In this paper, we present a block cipher encryption method that uses a 3-D Lorenz chaotic map, which is used to generate a highly randomized encryption-decryption key. The cipher form generated from an LED with a 3-D Lorenzo chaotic map has successfully passed the 15 statistical tests specified by the National Institute of Standards and Technology (NIST) SP 800-22, thus confirming its randomness. Furthermore, we maintain optimal performance and the best possible computation cost and security balance. In addition, the proposed method can be used on resource-constrained devices due to the low computational costs.

The rest of this paper is arranged as shown below. The literature review will be addressed in the next section. The Lorenz chaotic maps and the history of LED are covered in Section 3. Section 4 will go over the preferred technique. The fifth section will consist of an evaluation of the proposed method and a discussion of the findings. Finally, in Section 6, the conclusions will be provided.

2 Related work

Many articles were used to develop lightweight cryptography algorithms according to chaos theory. Table 1 covers relevant work by technique, performance, and results. The key weaknesses and efficient attacks have been found, and the authors use differential enumeration, key-bridging, and key-dependent sieve techniques. Recent studies focus on key recovery attacks [24]-[25].

Hamdi et al. [17] proposed a hybrid encryption algorithm (HEA) structure for the stream and block cipher algorithms using the Chirikov Standard Maps (CSM) to reduce computational overhead significantly. The suggested technique employs two primary operations: one to construct a one-time pseudo-random data block for usage in stream cipher and the other to generate substitution and permutation tables in the initial phase and execute rounds for confusion and diffusion processes in the block cipher.

M. Sharafi et al. [26] proposed the Modified Block Cipher depending on a Chaotic (MBCC) algorithm. It has a substitution-permutation structure and uses the principles of chaos theory to make it more resistant to differential and statistical attacks while using the same amount of resources. Lina Ding et al. [27] utilized a chaotic system and two Nonlinear Feedback Shift Registers (NFSRs) to build a new stream cipher for resource-constrained devices and applications. It digitizes the Logistic chaotic sequence and merges it with NFSRs and multiplexers to generate a new lightweight stream cipher that may be actively utilized for encryption in resource-constrained devices.

Ameer N. et al. [24] proposed a new hybrid method for making keys based on chaos theory. They presented a 2-D chaotic system (a hybrid of Henon and Cat chaotic maps) combined with a PRESENT lightweight algorithm to enhance its security. The Proposed Chaotic Key Generator (PCKG) approach outperforms the PRESENT cipher in terms of throughput, processing time, storage space utilization, and memory usage to achieve a high level of security.

Zaid M. et al.[25] applied the chaotic system of the 2D logistic map to create pseudo-random keys for a suggested hybrid system based on two cryptographic algorithms, Salsa20, and PRESENT, to increase the complexity of the recommended method. The proposed technique struck a balance between computational performance and ciphertext complexity.

Lamia A. Muhalhal. et al. [28] presented a lightweight approach to construct a strong, sufficiently random keystream to achieve sound diffusion, avoid being predicted by adversaries, and withstand known assaults. They found that the performance of the Salsa20 approach with chaotic maps outperforms Salsa20 in terms of data integrity and secrecy. R. Ziaur et al. [29] suggested a

Table 1: Summary table of the related works

Ref.	Chaotic Type	Goal	Construction	Results
[17]	Chirikov Chaotic Maps (CSM)	Using (CSM) to increase security and minimize the encryption time.	New encryption method based on stream and block ciphers using chaotic standard map (CSM).	• Has great cryptographic features • Sensitive to small changes in secret key • Resistant to common cryptanalytic attacks
[24]	Hybrid Cat and Henon chaotic maps	Implement two chaotic maps to increase security, processing time, less memory consumption for lightweight PRESENT cipher.	Proposed chaotic key generation (PCKG) using 2-D chaos system to increase PRESENT security.	• Improve PRESENT cipher security • Provide high level of encryption in IoTs.
[25]	2D logistic chaotic map	Implement 2D chaotic maps to generate pseudo-random keys to increase security of PRESENT algorithm.	2D chaotic system applied to generate pseudo-random keys for making the PRESENT and Salsa20 encryption more complexity.	Achieve the tradeoff between the complexity and computation speed for ciphertext.
[26]	Modified Block Cipher based on Chaotic (MBCC)	Employs chaos theory to resist statistical and differential attacks while conserving resources in WSNs.	Modified BCC (MBCC) algorithm, using chaos theory to improve (BCC) security.	MBCC exceeds BCC in time, energy consumption, memory usage, and security.
[27]	Logistic chaotic sequence	Generate cryptosystem with good complexity lightweight stream cipher.	Combining Logistic chaotic sequence with Nonlinear Feadback Shift Registers (NFSRs) to generate new stream cipher.	Obtain effective encryption stream cipher design for resource-constrained devices.
[28]	Lorenz,Henon , Rabinovich Fabrikant, and Chua circuit maps	To increase security and data integrity of Salsa20 stream cipher.	Using four chaotic maps to generate a random keystream for Salsa20 algorithm.	The proposed approach increases the security level of Salsa20 and its speed for limited-resource devices.
[29]	Logistic map	Improve the AES security for IoT devices.	Use logistic map to generate a key scheduling for AES cipher.	• Increase Key generation complexity. • The proposed method can protect the confidentiality of critical IoT data.
[30]	1-D chaotic maps and 2-D cat map	Attain strong cryptosystem based on chaos theory against different attacks.	• Form a Pseudorandom number generator based on four discrete 1-D chaotic map (PRNG-CS). • Strong S-box based on 2-D cat map.	The suggested chaos-based cryptosystem is resistant to statistical and cryptographic attacks.
[31]	3-D continuous chaotic system	Provide ready to use random bits to enhance the security in cryptosystem.	Utilize a chaos-key generator and chaos based true random number generator for secure communications.	Implement a highly integrated analog circuit architecture for image encryption.

method based on an excellent chaotic idea and a logistic map. They have created and tested a key-scheduling mechanism for encrypting massive data volumes. The proposed adjustment to the key-origination matrix and the S-box strategy decreases its chances of being broken. Their method illustrates how seeming disorder escalates the intended key initiation preceding message transmission.

Fethi D. et al. [30] developed a novel, safe chaos-based cryptosystem employing cipher block chaining (CBC) mode. Their system is equipped with a reliable pseudo-random number generator for chaotic sequences

(PRNG-CS). The method uses four separate 1-D chaotic maps to stop the divide-and-conquer attack and make the generated sequences more random and longer. The security analysis and experimental results showed that the proposed cryptosystem achieved high confusion and diffusion effects.

Nguyen N. [31] shows how to design and build a true random number generator based on chaos and a data encryption system based on chaos keys for secure communications. The chaos-based one-time pad encryption method utilizing the chaos-key generator demonstrates the benefits of using such a random number generator for secure communications. In terms of data secrecy and completion time, they compared the encryption and decryption characteristics of the chaos-key-based image encryption system to those of the standard AES128 algorithm.

Discussion: As stated in Table 1. the work in [17] HEA algorithm structure is very similar to traditional AES and has the same complexity and running time, which is not suitable for resource-constrained devices. The method in [24] used a combination of chaotic Cat and Henon maps. In the equation for the 2D Cat map, there is a mod operation, which is a costly mathematical operation. The authors of [25] implement two algorithms Salsa20 with PRESNT along with chaotic maps, increasing the overhead computation cost. In the study [26] using a lot of substitution-permutation operations leading to high consuming time. Using two (NFSRs) high-cost operations increases complexity and cost in the study [27]. The authors [28] used four chaotic maps to increase the security levels of the Salsa20 cipher but decrease performance by causing high computation costs, especially when used with limited resource-constrained devices. The work [29] used a chaotic map for key scheduling for AES, and already it is not suitable for devices with limited resources. The method in [30] applied four discrete 1D chaotic maps for PRNG and a 2D Cat map for S-box, which caused very high computation costs. The technique in [31] limitation and efficiently method not clear. As we have seen, the limitations of literature reviews in computation costs, thus the LWC algorithms, are still challenging.

In this proposed scheme, we use a three-dimensional continuous Lorenz chaotic system to generate the one-time pseudo-random bit key for LED lightweight block cipher because of its aperiodicity, highly randomized, unpredictable nature, and extreme sensitivity to initial values. Moreover, it also has the following three benefits: The complex system structure makes predicting the chaotic output sequence more difficult; the solution space is composed of three parameters and three initial values, and it has a significantly higher density than the low-dimensional discrete chaotic maps, and the use of three chaotic sequences makes cipher design more flexible [32].

3 Background

3.1 The LED block cipher

LED consider one of the states of the art in lightweight cryptography target algorithms for implementation in the report that issues from the working group of Cryptography Research and Evaluation Committees (CRYPTREC) to evaluate and monitor the security of cryptographic techniques used in Japanese e-Government systems. Several lightweight algorithms were presented, like CLEFIA, PRESENT, Piccolo, TWINE, PRINCE, SIMON, and SPECK [32].

LED is a substitution permutation network (SPN) symmetric cipher and is regarded as a lighter version of AES cipher, whose block size is 64-bit plaintext. LED-64 and LED-128 are its two primary versions, with 64-bit and 128-bit keys, respectively. The cipher state x, the 64-bit key, 128-bit key are categorized as 16 four-bit nibbles in a 4×4 array matrix, as depicted in (1), (2), and (3), respectively. Also, Figure 1. shows both (Top) 64-bit key arrays and (Bottom) 128-bit key arrays. We concentrate on LED-64 as shown in Algorithm 1 pseudocode for lightweight applications and refer to it as LED [33].

LED's encryption technique consists of two fundamental operations: addRoundKey and step. In the first operation, the plaintext (state) is XORed with the secret key (K), then passed through the second operation (step), which comprises four rounds of state encryption. The operations addRoundKey and step are done eight times for the 64-bit key array matrix. The result is XORed with K once more to generate the ciphertext. Each round function consists of AddConstants (AC), SubCells (SC), ShiftRows (SR), and MixColumnSerial (MC) operations, which compute a 4 × 4 matrix multiplication in GF(2^4). 32 rounds resemble the structure of AES [33].

When the user-supplied key is frequently utilized as-is, the absence of key scheduling in LED is a distinctive characteristic. LED is very compact in hardware and has a good software implementation performance profile.

$$X = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix} \quad (1)$$

$$k_1 = \begin{bmatrix} K_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \quad (2)$$

$$k_{1,2} = \begin{bmatrix} K_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \begin{bmatrix} K_{16} & k_{17} & k_{18} & k_{19} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{24} & k_{25} & k_{26} & k_{27} \\ k_{28} & k_{29} & k_{30} & k_{31} \end{bmatrix} \quad (3)$$

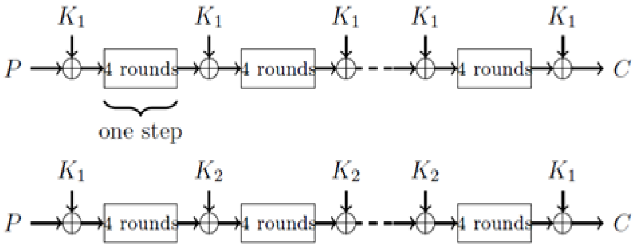

Figure 1: (Top) 64-bit key array k_1 of LED (Bottom) 128-bit key array $k_{1,}$ k_2 of LED [33]

Remark 1: Practically, LED is no longer secure and has been broken [7], [12]-[14]. Therefore, LED must achieve a high diffusion and confusion level to withstand known attacks. Chaos-based encryption provides a very high level of security because of the distinctive properties of chaotic systems defined by a group of nonlinear deterministic dynamic equations. Thus, we suggest using 3-D Lorenz chaotic map to increase the security of LED, as stated in the forthcoming section.

Algorithm 1: LED Algorithm (64-bit Key) [33]

```
Input:   Key (k), Plaintext (State)
Output: Ciphertext
For i =1 to 8                              ▷ LED encryption
    State ← State ⊕ K
    for m=0 to 3
        Add Constants(State)
        Sub Cells(State)
        Shift Rows(State)
        Mix Columns Serial(State)
    end for
End For
Ciphertext ← State ⊕ K
For i =8 to 1                              ▷ LED decryption
    State ← Ciphertext (State) ⊕ K
    for m=3 to 0
        Mix Columns Serial(State)
        Shift Rows(State)
        Sub Cells(State)
        Add Constants(State)
    end for
End for
Plaintext ← State ⊕ K
```

3.2 The 3-D Lorenz chaotic system

Chaos is regarded as a tremendous advancement in data security due to its many applications in numerous fields, such as computer science. As chaotic systems are unpredictable, ergodic, random, and very sensitive to initial conditions, they are well-suited for encryption, decryption, and secure transmission. Multiple chaotic system models were coupled to generate pseudo-random sequences, thus expanding the space for the secret key and enhancing the algorithm's security [34].

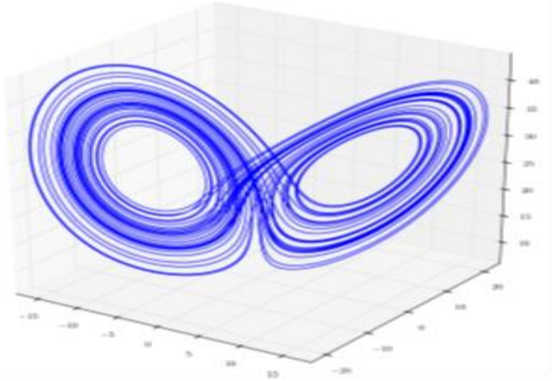

Figure 2 :The Lorenz attractor

Minor changes to the model parameters of the system could cause it to go into a chaotic state. The chaotic behavior of these kinds of systems is crucial for constructing cryptographic methods. The Lorenz system is a chaotic dynamical map in three dimensions. Edward Lorenz invented the coupled differential equation in 1963. When the Lorenz system is plotted, it produces a Butterfly-like attractor, as illustrated in Figure 2. [27]-[35].

In the past two decades, there has been an increase in interest in chaos-based cryptography. The primary characteristics of Lorenz chaotic systems (sensitivity to initial values, mixing property, easy analytic description, and highly complicated behavior) make them particularly desirable for creating novel cryptosystems [34]. The convergence characteristics of chaotic systems depend on the nonlinearity and instability of particular states in dynamic systems [17]. Where: the system state (x, y, z), the control parameters values for which it is chaotic are σ=10, ρ=28, β=2.667. These are the control parameters, and these values are critical and influential because they define the system’s behavior. A simple Ordinary differential equation (ODE) can explain the system (4), which is a three-dimensional system whose dynamic changes concerning time:

$$\begin{aligned} x' &= a\,(y - x\,) \\ y' &= (\sigma - z)x - y \\ z' &= xy - bz \end{aligned} \qquad (4)$$

Remark 2: The 3-D Lorenz system generates one-time pseudo-random numbers. These chaos numbers are used to create the key. This highly randomized key generated by chaotic is used as input to the LED algorithm to encrypt data securely by increasing the randomness of the key, which reflects positively on the data encryption process.

4 LED with 3-D Lorenz chaotic map

The proposed approach builds a block cipher using chaotic maps and the LED algorithm. The key generated by the Lorenz chaotic map is XORed twice with the state during the encryption process to encrypt the 64-bit block of data to create the ciphertext. Every block cipher has the benefit of being easy to use. Conversely, the key generation process ultimately determines how strong these ciphers are. This research aims to enhance the LED algorithm to build a strong enough unexpected key that attackers won't expect, like meet in a middle attack and scan-based attack. The cipher system's complete recommended block diagram is illustrated in Figure 3.

We modify the LED algorithm to make it more diffusion and randomization using 3-D Lorenz chaotic map as shown in Algorithm 2 because it has good randomness suited for a key generation utilized to modify the LED algorithm. Therefore, our work adopted this chaos to create an algorithm for the input LED to retain the characteristics of effective randomness. Now LED algorithm accepts a 64-bit key chaotic map to enhance its level of security.

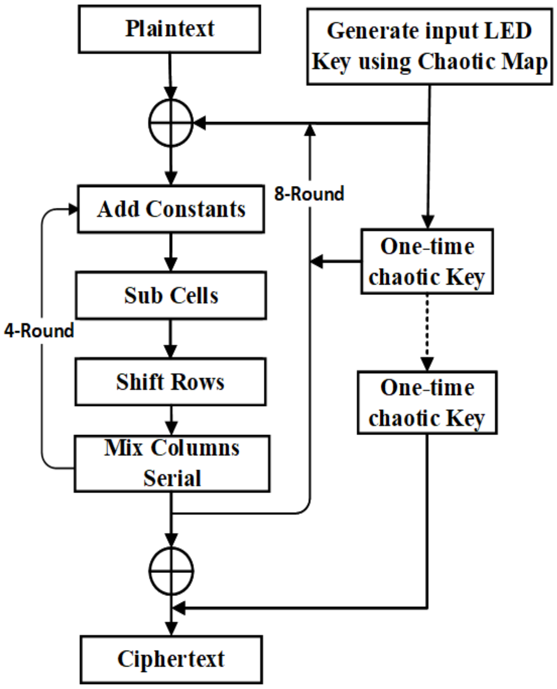

Figure 3: LED-3-D Lorenz chaotic system

Algorithm 2: LED key generation using 3-D Lorenz chaos theory (64-bit)

```
Input: $x_0 = 0,\ y_0 = 1,\ z_0 = 20$, σ=10, ρ=28, β=2.667 //
        as a parameter and initial conditions of the Lorenz
        chaotic system
Output: Key (64-bit)
For n= 1 to 32
    $x_n = a(y_0 - x_0)$ // generate pseudo-random
            numbers using Lorenz chaos equations
    $y_n = (\sigma - z_0)\, x_0 - y_0$
    $z_n = y_0 - bz_0$
    $x_n = num2str\ (x_n\ (n, 1), 5)$ //convert numbers to
            string
    $y_n = num2str\ (y_n\ (n, 2), 5)$
    $z_n = num2str\ (z_n\ (n, 3), 5)$
    $outdx_n$ = dec2bin(str2num($x_n$ (4)) //convert decimal
            numbers to binary
    $outdy_n$ = dec2bin(str2num($y_n$ (4))
    $outdz_n$ = dec2bin(str2num( $z_n$ (4))
End For
Key =$outdx_n$+$outdy_n$+$outdz_n$ //apply concatenate
        process after converting the results to binary
Return Key(64-bit)
```

5 Results and discussion

The proposed approach was built and implemented in a MATLAB R2021a environment on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz running at 1.99GHz, and 8GB of RAM running on Windows 10.

We assess our work using the 15 statistical NIST Test suite designed to test the randomness of ciphertext generated by LED 3-D Lorenz chaotic map. These tests are a good starting point for deciding whether or not a generator is adequate for a specific cryptographic application. In addition, we evaluate the proposed method's computation cost, which determines its good performance.

5.1 NIST Statistical test

In this paper, we propose lightweight block encryption based on the LED algorithm and chaotic maps to achieve a high level of randomness and propagation to resist known attacks. A good fusion merges the qualities of both the LED algorithm and the random maps to become more random to improve the weaknesses of the LED algorithm. Especially after the LED algorithm was broken and LED is no longer as safe as before. As a result, LED 3-D Lorenz achieves a high level of diffusion and confusion to resist known attacks.

In addition, many statistical tests exist to evaluate the random properties of cryptographic algorithms. NIST SP 800-22 is used to consider statistical analyses. The NIST tests use the significant value to determine if the succession rate is random. The sequence is regarded as random if the P-value is less than 0.01 or non-random if it is greater than 0.01. The suggested key generation method and the LED cipher algorithm are put through the 15 NIST tests. A discussion of the test results will follow.

- **Frequency (Monobit) Test:** According to NIST testing, the suggested technique generally outperforms the LED, as indicated in Table 2. , It increases by almost 0.6998 more than the LED algorithm.
- **Frequency within a Block Test:** According to NIST testing, the suggested technique generally outperforms the LED algorithm, as in Table 2. , It increases by almost 0.4992 more than the LED algorithm.
- **Overlapping Template Matching Test:** As indicated in Table 2. , the suggested method often outperforms the LED algorithm, increasing by roughly 0.3376 higher than the LED algorithm.
- **Maurer's Universal Test:** According to statistical NIST tests, the suggested technique often outperforms the LED by about 0.0280 points, as shown in Table 2.
- **Linear Complexity Test:** The suggested method often outperforms the LED, as indicated in Table 2. , According to NIST testing, it increases by roughly 0.7723 more than the LED algorithm.
- **Serial Test:** The suggested method often outperforms the LED, as indicated in Table 2. , NIST testing

shows that it increases by roughly 0.4570 more than the LED algorithm.

- **Approximate Entropy Test:** According to NIST testing, the suggested technique outperforms the LED algorithm, as shown in Table 2. , This increase is almost 0.0706 higher than the LED algorithm.
- **Cumulative Sums (Cusums) Test:** The proposed technique, which increases by almost 0.7143 more than the LED algorithm, according to NIST testing, is generally higher than the LED, as indicated in Table 2.
- **Random Excursions Variant Test:** As indicated in Table 2. , the proposed technique is generally superior to the LED, which increases by almost 0.1662 higher than the LED algorithm, according to tests NIST suite.
- **Random Excursions Test:** The suggested technique often outperforms the LED, as indicated in Table 2. , which rises by approximately 0.0371 higher than the LED algorithm, based on the NIST tests suite.
- **Runs Test:** According to NIST testing, the suggested technique generally outperforms the LED algorithm, as shown in Table 2. , This increase is almost 0.1821 higher than the LED algorithm.
- **Longest-Run-of-Ones Test:** The suggested method often outperforms the LED, as indicated in Table 2. , According to NIST testing, it increases by roughly 0.7723 more than the LED algorithm.
- **Binary Matrix Rank Test:** The proposed method is generally less than the LED, as indicated in Table 2. , which decreases by nearly 0.0977 compared to the LED algorithm, according to NIST tests.
- **Discrete Fourier Transform Test:** The proposed technique is generally superior to the LED, as demonstrated in Table 2. , which increases by almost 0.8696 times more than the LED algorithm, according to NIST tests.
- **Non-overlapping Template Matching Test:** The proposed technique is often less than the LED, as indicated in Table 2. , This decreases by around 0.6661 compared to the LED algorithm, as determined by NIST tests.

Table 2: Statistical NIST test suite

NIST tests	LED	Proposed algorithm
Frequency (Monobit)	0.1563	0.8561
Frequency within a Block	0.2720	0.7713
Cumulative Sums (Cusums)	0.1956	0.9099
Runs	0.0051	0.1872
Longest-Run-of-Ones	0.4338	0.8685
Rank test	0.8264	0.7288
Discrete Fourier Transform	0.1304	1.0000
Non-overlapping Template	0.7656	0.0994
Overlapping Template	0.6507	0.9883
Maurer's “Universal’’	0.0313	0.0593
Approximate Entropy	0.1352	0.2058
Random Excursions	0.5499	0.5869
Random Excursions Variant	0.2892	0.4554
Serial	0.1089	0.5659
Linear Complexity	0.0862	0.8585

Remark 3: 13 NIST statistical tests out of 15 indicate that the LED with the key generated by using three dimensional Lorenz chaotic system outperforms the LED cipher, as shown in Table 2. , given the highly randomized and nonlinear output ciphertext generated by the proposed scheme.

5.2 Computation cost

As noted in Table 3., the encryption and decryption times are very close because of our use of computationally inexpensive key generation based on the 3-D Lorenz chaotic method. This additional unnoticed time in the encryption and decryption processes has little effect compared to the increased security of data encrypted by random chaos based on the 3-D Lorenz chaotic map.

Table 3: Comparison computation time cost in (ms)

Size in block (64-bit)	Size in bits	LED encryption time	Proposed algorithm encryption time	LED decryption time	Proposed algorithm decryption time
1	64	0.1140	0.1914	0.0584	0.0591
10	640	0.3682	0.4484	0.3700	0.3771
100	6400	2.4569	2.5844	3.3927	3.3980
250	16000	6.0363	6.0735	8.4943	8.4292

From Table 3., we observe that there is a slight difference in encryption and decryption computation costs between the LED and our proposed scheme. For example, it takes about 0.0372 ms in the encryption process when the block size is 16000 bits, as registered in the above table, which is unremarkable compared with gaining high randomization. Besides, we attain high-randomization ciphertext to resist known attacks.

6 Conclusion

LED as a block cipher was broken and needed to meet the security requirements, and it is no longer as safe as it once was. As a result, LED requires greater randomness, confusion, and diffusion. This paper describes a modified LED that adopted a three-dimensional Lorenz chaotic map to achieve a satisfactory level of confusion and diffusion. The difference in computational complexity is slight and almost not noticed between the proposed method and LED. On the other hand, our approach has achieved an impressive security increase, enabling it to prevent attacks. The proposed method improves security by using more randomization with key generation, and it could be used with other techniques like a lightweight block cipher. Key generation raises the complexity of the algorithm and adds greater flexibility. LED has been modified to be high-security and robust enough to use as a lightweight block cipher on devices with limited resources. The performance of random ciphers was evaluated using 15 statistical NIST tests suite developed to assess pseudo-random numbers in cryptographic systems applications. It successfully bypassed the randomness of the proposed method. According to the statistical NIST test suite, the performance acquired by the proposed method increases by nearly 0.3003, higher than that reached by the traditional LED in terms of data confidentiality and integrity.

References

[1] H. Wu and H. Wu, "Research on Computer Network Information Security Problems and Prevention Based on Wireless Sensor Network," in *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2021, pp. 1015–1018. doi: 10.1109/IPEC51340.2021.9421303.

[2] M. A. Latif, M. Bin Ahmad, and M. K. Khan, "A Review on Key Management and Lightweight Cryptography for IoT," in *2020 Global Conference on Wireless and Optical Technologies (GCWOT)*, 2020, pp. 1–7. doi: 10.1109/GCWOT49901.2020.9391613.

[3] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: a solution to secure IoT," *Wirel. Pers. Commun.*, vol. 112, no. 3, pp. 1947–1980, 2020, doi: 10.1007/s11277-020-07134-3.

[4] R. Anusha, M. J. Dileep Kumar, V. S. Shetty, and N. Prajwal Hegde, "Symmetric Key Algorithm in Computer security: A Review," *Proc. 4th Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2020*, pp. 765–769, 2020, doi: 10.1109/ICECA49313.2020.9297547.

[5] H. H. Al-badrei and I. S. Alshawi, "Improvement of RC4 Security Algorithm," *Adv. Mech.*, vol. 9, no. 3, pp. 1467–1476, 2021.

[6] K. Gupta, D. Gupta, S. K. Prasad, and P. Johri, "A Review on Cryptography based Data Security Techniques for the Cloud Computing," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*, 2021, pp. 1039–1044. doi: 10.1109/ICACITE51222.2021.9404568.

[7] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021, doi: 10.1109/ACCESS.2021.3052867.

[8] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptogr. Eng.*, vol. 8, no. 2, pp. 141–184, 2018, doi: 10.1007/s13389-017-0160-y.

[9] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 1, pp. 142–151, 2015, doi: 10.1109/TIFS.2014.2365734.

[10] W. Diehl, A. Abdulgadir, J. P. Kaps, and K. Gaj, "Comparing the cost of protecting selected lightweight block ciphers against differential power analysis in low-cost FPGAs," *Computers*, vol. 7, no. 2, 2018, doi: 10.3390/computers7020028.

[11] H. Mestiri, Y. Salah, and A. A. Baroudi, "A Secure Network Interface for on-Chip Systems," *Proc. - STA 2020 2020 20th Int. Conf. Sci. Tech. Autom. Control Comput. Eng.*, pp. 90–94, 2020, doi: 10.1109/STA50679.2020.9329296.

[12] K. Jeong, H. Kang, C. Lee, J. Sung, and S. Hong, "Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT , Piccolo and LED," *IACR Cryptol. ePrint Arch.*, vol. 2012, p. 621, 2012.

[13] T. Isobe and K. Shibutani, "Security analysis of the lightweight block ciphers XTEA, LED and Piccolo," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7372 LNCS, pp. 71–86, 2012, doi: 10.1007/978-3-642-31448-3_6.

[14] W. Diehl, A. Abdulgadir, J. P. Kaps, and K. Gaj, "Side-channel resistant soft core processor for lightweight block ciphers," *2017 Int. Conf. Reconfigurable Comput. FPGAs, ReConFig 2017*, vol. 2018-Janua, pp. 1–8, 2018, doi: 10.1109/RECONFIG.2017.8279819.

[15] S. Subramanian, M. Mozaffari-Kermani, R. Azarderakhsh, and M. Nojoumian, "Reliable Hardware Architectures for Cryptographic Block Ciphers LED and HIGHT," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1750–1758, 2017, doi: 10.1109/TCAD.2017.2661811.

[16] A. Ali, M. A. Khan, R. K. Ayyasamy, and M. Wasif, "A novel systematic byte substitution method to design strong bijective substitution box (S-box) using piece-wise-linear chaotic map," *PeerJ Comput. Sci.*, vol. 8, pp. 1–38, 2022, doi: 10.7717/peerj-cs.940.

[17] M. Hamdi, J. Miri, and B. Moalla, "Hybrid encryption algorithm (HEA) based on chaotic system," *Soft Comput.*, vol. 25, no. 3, pp. 1847–1858, 2021, doi: 10.1007/s00500-020-05258-z.

[18] R. Anandkumar and R. Kalpana, "Analyzing of chaos based encryption with Lorenz and Henon map," *Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018*, pp. 204–208, 2019, doi: 10.1109/I-SMAC.2018.8653652.

[19] C. D. McDermott and A. Petrovski, "Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks," *Int. J. Comput. Networks Commun.*, vol. 9, no. 4, pp. 45–56, 2017, doi: 10.5121/ijcnc.2017.9404.

[20] R. Iqbal, F. Doctor, B. More, S. Mahmu, and U. Yousuf, "Faiyaz Doctor," *Technol. Forecast. Soc. Change*, pp. 1–25, 2018.

[21] A. H. Jabbar and I. S. Alshawi, "Spider monkey optimization routing protocol for wireless sensor networks.," *Int. J. Electr. \& Comput. Eng.*, vol. 11, no. 3, 2021, doi: 10.11591/ijece.v11i3.pp2432-2442.

[22] A. Abdaoui, A. Erbad, A. K. Al-Ali, A. Mohamed, and M. Guizani, "Fuzzy Elliptic Curve Cryptography for Authentication in Internet of Things," *IEEE Internet Things J.*, 2022, doi: 10.1109/JIOT.2021.3121350.

[23] R. Wang and W. Ji, "Computational Intelligence for Information Security: A Survey," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 5, pp. 616–

629, 2020, doi: 10.1109/TETCI.2019.2923426.

[24] A. N. Abdulraheem and B. M. Nema, "Secure IoT Model Based on PRESENT Lightweight Modified and Chaotic Key Generator," *Proc. 2020 1st Inf. Technol. to Enhanc. E-Learning other Appl. Conf. IT-ELA 2020*, pp. 12–18, 2020, doi: 10.1109/IT-ELA50150.2020.9253079.

[25] Z. M. J. Kubba and H. K. Hoomod, "A Hybrid Modified Lightweight Algorithm Combined of Two Cryptography Algorithms PRESENT and Salsa20 Using Chaotic System," in *1st International Scientific Conference of Computer and Applied Sciences, CAS 2019*, 2019, pp. 199–203. doi: 10.1109/CAS47993.2019.9075488.

[26] M. Sharafi, F. Fotouhi-Ghazvini, M. Shirali, and M. Ghassemian, "A low power cryptography solution based on chaos theory in wireless sensor nodes," *IEEE Access*, vol. 7, no. c, pp. 8737–8753, 2019, doi: 10.1109/ACCESS.2018.2886384.

[27] L. Ding, C. Liu, Y. Zhang, and Q. Ding, "A new lightweight stream cipher based on chaos," *Symmetry (Basel).*, vol. 11, no. 7, pp. 1–12, 2019, doi: 10.3390/sym11070853.

[28] L. A. Muhalhal and I. S. Alshawi, "Improved Salsa20 Stream Cipher Diffusion Based on Random Chaotic Maps," *Informatica* vol. 46, pp. 95–102, 2022.

[29] Z. Rahman, X. Yi, I. Khalil, and M. Sumi, "Chaos and Logistic Map based Key Generation Technique for AES-driven IoT Security," International *Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 177-193. Springer, Cham, 2021.

[30] F. Dridi, S. El Assad, W. El Hadj Youssef, M. Machhout, and R. Lozi, "Design, Implementation, and Analysis of a Block Cipher Based on a Secure Chaotic Generator," *Appl. Sci.*, vol. 12, no. 19, 2022, doi: 10.3390/app12199952.

[31] N. Nguyen, L. Pham-Nguyen, M. B. Nguyen, and G. Kaddoum, "A Low Power Circuit Design for Chaos-Key Based Data Encryption," *IEEE Access*, vol. 8, pp. 104432–104444, 2020, doi: 10.1109/ACCESS.2020.2998395.

[32] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," *Natl. Inst. Stand. Technol.*, vol. NISTIR 811, p. 26, 2017, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf%0Ahttp://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf

[33] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," *International workshop on cryptographic hardware and embedded systems*, pp. 326-341. Springer, Berlin, Heidelberg, 2011.

[34] T. Li, B. Du, and X. Liang, "Image Encryption Algorithm Based on Logistic and Two-Dimensional Lorenz," *IEEE Access*, vol. 8, pp. 13792–13805, 2020, doi: 10.1109/ACCESS.2020.2966264.

[35] E. Lorenz, "Deterministic Nonperiodic Flow," Universality in Chaos. pp. 367–378, 2017. doi: 10.1201/9780203734636-38.