A New Approach to Key Generation Using 3D Chaotic Systems in Hybrid Lightweight IoT Based Cloud Security Algorithms

Mohammed Ali Qasem[1], S.B. Thorat[2], Bokare Madhav Motiram[3],
Pritam Rajendra Pati[4], Aymen M. Al-Hejri[5]
[1]*School of Computational Sciences, S.R.T.M. University, Nanded, India.*
[2]*Campus Director, (GPGI), Rashtrasant Tukadoji Maharaj Nagpur University (RTMNU)*
[3,4]*SSBES's Institute of Technology and Management SRTM University, Nanded.*
[5]*School of Computational Sciences, S.R.T.M. University, Nanded, India*
doi.org/10.64643/IJIRTV12I8-191503-459

***Abstract*—In the era of cloud computing and IoT, securing data through encryption is crucial to prevent unauthorized access and protect sensitive information. This study proposes a novel approach to key generation using a 3D chaotic system within a hybrid lightweight encryption algorithm tailored for IoT-based cloud security. The encryption technique combines the Lightweight Encryption Algorithm (LEA) and Advanced Encryption Standard (AES) to create a robust defense mechanism for critical data. Enhancements such as AES S-box, Shift Rows, Mix Columns, and Add Round Key have been integrated into the Z-LEA algorithm, increasing the complexity of the encrypted data and minimizing the risk of unauthorized decryption. The 3D chaotic system plays a key role in generating highly secure, nonlinear, and dynamic keys. Additionally, the proposed system reduces both the number of encryptions rounds and execution time (encryption/decryption) by implementing parallel processing techniques. Experimental results demonstrate that the Z-LEA encryption system is resistant to various attack vectors, significantly reduces computation time, and generates a highly random key stream, validated through the NIST randomness test.**

***Index Terms*—Hyperchaos, three positive Lyapunov, Z-LEA, Diffusion, Confusion, SP- network**

I. INTRODUCTION

With the rapid advancement of information technologies, data exchange has become a daily routine for most people. Since this data often contains vital and sensitive information, ensuring its protection has become more critical than ever. Cryptography, the science of protecting data, works by converting information into an unreadable form during transmission so that only authorized parties can retrieve the original data. With many encryption methods available, the key is selecting the most suitable one. Today's fast-paced technology and resource-limited communication channels require encryption algorithms that balance efficiency with security. This is where lightweight algorithms stand out, offering robust protection without the heavy computational burden associated with traditional algorithms [1].

Lightweight algorithms like RC6, ChaCha, Present, and LEA are known for their simplicity and high security, despite requiring minimal computational resources for encryption and decryption [2], [3]. Their low computation time makes them well-suited for environments such as IoT (Internet of Things), IoE (Internet of Everything), and cloud computing [4], [5]. These algorithms typically fall into two categories: block ciphers and stream ciphers. Symmetric encryption uses the same key for both encryption and decryption, while asymmetric encryption uses two different keys [5], [6].

In mathematics, chaos theory deals with complex dynamic systems that are highly sensitive to initial conditions, resulting in random-like behavior [7]. Chaotic systems can be modeled by nonlinear differential equations with multiple variables, where the number of variables determines the dimensions of the chaotic system. These equations dictate the system's evolution over time [8]. Chaos theory has practical applications in fields such as weather forecasting, population dynamics, and - most importantly for this study - cryptography and secure communication [9], [10]. Understanding chaotic behavior can offer insights into complex processes and help develop more secure systems.

To keep up with modern technology's demands for speed and security, encryption algorithms must process data in parallel, speeding up operations by handling tasks simultaneously. Parallel processing, which takes advantage of multi-core processors, dramatically reduces encryption time and strengthens the overall security system [11].

1.1. Problem Statement

As cloud computing grows and increasingly handles sensitive data, there is a rising need for secure communication networks. Traditional encryption methods, though secure, are computationally intensive and can negatively impact latency and performance in cloud environments. To address this, lightweight cryptographic techniques are required, offering secure data transfer with minimal CPU usage, memory consumption, and power draw, making them ideal for modern cloud systems.

1.2. Motivation

Like other technologies, cloud computing moves large amounts of data between its servers. Given the rapid development of modern data transmission channels, it is crucial to protect this data from unauthorized access. The need arises for a highly secure system that operates at high speeds to keep up with the demands of today's transfer channels. This study aims to develop a hybrid lightweight encryption algorithm, optimized for cloud environments, which operates in parallel to maximize protection while ensuring confidentiality.

1.3. Research Contribution

1. This study proposes a new lightweight hybrid encryption algorithm, named Z-LEA, based on the simple yet powerful ARX operations of LEA and the strong Substitution- Permutation Network (SPN) of AES.
2. While Z-LEA integrates unique components, it builds on the well-known LEA and AES ciphers, using a two-level scheme that includes SPN for diffusion and modified LEA operations at the first level.
3. The algorithm employs parallel processing, which distributes the encryption task across multiple CPU cores, significantly reducing the computational cost when handling large data streams.

Experimental results indicate that Z-LEA offers security comparable to established standards like AES, but with far fewer processing resources, making it ideal for cloud computing.

1.4. Paper Organization

This paper is organized into six sections. Section (1) introduces the study. Section (2) covers the background theory. Section (3) discusses related research. Section (4) presents the proposed algorithm. Section (5) focuses on the results. Finally, Section (6) provides a discussion of the findings.

II. BACKGROUND THEORY

Cryptography is a fundamental tool for securing information by encoding it in a way that hides the original content. Using cryptographic algorithms ensures that data transmitted across cloud networks remains secure from unauthorized access[12]. The choice of encryption method depends on the type of key used in the process, which divides cryptography into two main categories: symmetric and asymmetric encryption [13].

2.1 Symmetric Cryptography

In symmetric encryption, both the encryption and decryption processes use the same key. This means that the sender and receiver must share an identical key to encode and decode the information. Symmetric algorithms are generally faster and more efficient, making them ideal for environments requiring high-speed data transmission. Popular symmetric encryption algorithms include AES (Advanced Encryption Standard), DES (Data Encryption Standard), and 3DES (Triple Data Encryption Standard) [11], [14], [15].

2.2 Asymmetric Cryptography

Asymmetric encryption, on the other hand, uses a pair of keys: a public key and a private key. These keys are mathematically linked but designed in such a way that deriving one key from the other is computationally infeasible. The public key is widely shared, while the private key is kept secret. Any message encrypted with the public key can only be decrypted by the corresponding private key, and vice versa. Asymmetric algorithms tend to be slower and more resource- intensive compared to symmetric algorithms. Common examples include RSA (Rivest-Shamir- Adleman), ECC (Elliptic Curve Cryptography), and DSA (Digital Signature Algorithm) [16], [17], [18]. Figure 1 illustrates the mechanism of both symmetric and asymmetric encryption.

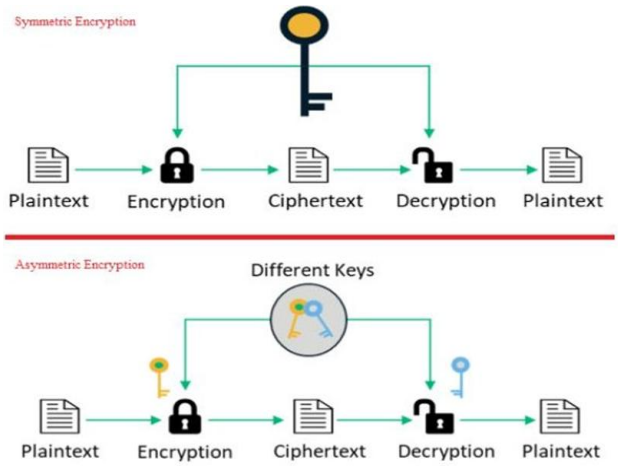


Figure 1: illustrates the mechanism of both symmetric and asymmetric encryption

2.3 Emergence of Lightweight Cryptography

The field of modern cryptography has undergone profound transformation, catalyzed by the varying demands of applications from expansive data centers to diminutive Internet of Things (IoT) devices. While traditional cryptographic algorithms are adept at securing data, their application is often impractical in environments with limited resources. This has led to the ascendance of lightweight ciphers as a pivotal area of research, driven by the imperative for cryptographic mechanisms capable of fluid operation within such constraints. The foundation of lightweight cryptography is built upon the recognition that conventional ciphers, tailored for high-performance environments, do not align with the needs of devices constrained by resources. As digital connectivity extends beyond traditional computing devices to include an array of embedded systems, wearable technology, and IoT sensors, the importance of lightweight ciphers in ensuring efficient operation under conditions of limited computational capacity, memory, and energy becomes increasingly pronounced [4]. Lightweight cryptography aims to address these constraints by developing algorithms that require fewer resources in terms of power, memory, and processing capacity while still providing a sufficient level of security. The need for lightweight cryptographic solutions has grown significantly with the expansion of IoT devices, such as sensors and wearable devices, which operate under stringent power and processing limitations [5].

2.4 Cryptography Terms

2.4.1. Substitution-Permutation Network (SPN)

Shannon,[17] introduced the concept of layering substitutions and permutations to create strong block ciphers. This design is known as the Substitution-Permutation Network (SPN). The alternating application of substitutions and permutations has an attractive effect: the substitution increases local confusion, while the permutation layer spreads this confusion throughout the sub- blocks within the network, ensuring diffusion [20].

Substitution Layer (or S-box) is a non-linear function that replaces blocks of input bits with blocks of output bits based on a predefined substitution table. These S-boxes are carefully designed to introduce confusion into the cipher, making it resistant to various forms of cryptanalytic attacks [19].

Permutation Layer rearranges the bit stream of the data, disrupting its original sequence to achieve the diffusion property. This rearrangement ensures that each output bit is influenced by a large number of input bits. By spreading the impact of each input bit across the entire block, permutation layers enhance the cipher's security. Figure 2 illustrates the diffusion effect achieved through this process.

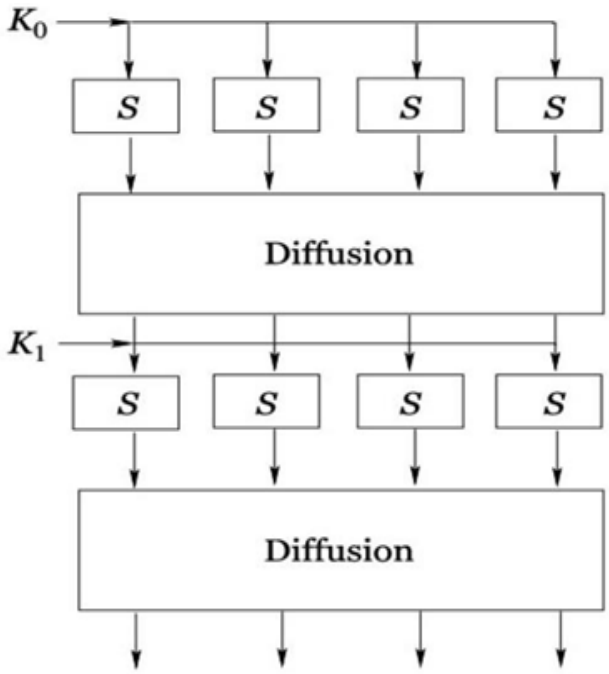


Figure 2: S-box with diffusion impact.

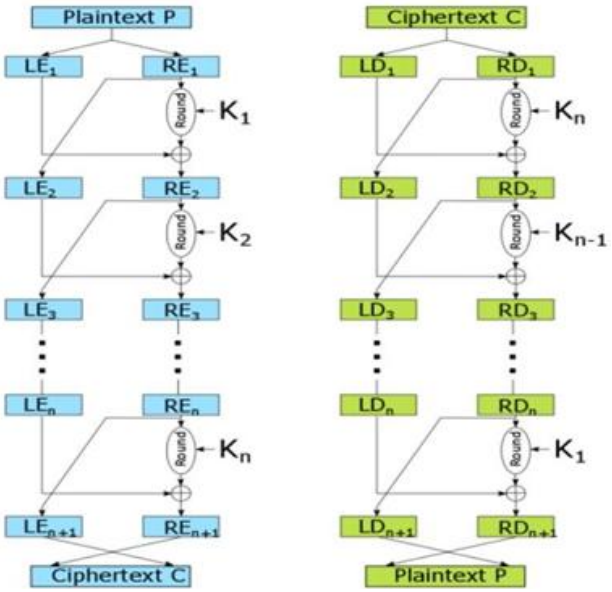


Figure 3: Feistel network with n rounds

2.5 Feistel Network

A Feistel network is a structure used in many encryption algorithms, in which the data block is split into two equal parts. Encryption is then applied in multiple rounds, with each round using a combination of substitution, permutation, and functions derived from the key [21]. The number of rounds can vary depending on the algorithm. One of the strengths of the Feistel network is that it is a reversible process, meaning the same input will always produce the same output. Figure 3 illustrates the Feistel network's working mechanism [22].

2.6 Lightweight Encryption Algorithm (LEA)

Lightweight Encryption Algorithm (LEA) was designed in 2013 by South Korean researchers Seokhie Hong, Jaechul Sung, Sangjin Lee, and Donghoon Chang to meet the needs of constrained environments like IoT devices, smart cards, and embedded systems. LEA operates with a 128-bit block size and supports key lengths of 128, 192, or 256 bits, with the number of encryption rounds varying between 24, 28, or 32, depending on the key size. The algorithm is built on ARX (Add, Rotate, XOR) operations [22].

LEA has proven its efficiency by using minimal memory and power, making it an ideal choice for environments with limited resources. It strikes a good balance between security and performance, making it well-suited for applications involving lightweight cryptography. It has undergone extensive cryptanalysis and has been found resistant to various forms of attacks, including differential and linear cryptanalysis [23].

Some of the main features of LEA:

- Lightweight Design: LEA is optimized for environments with limited computational resources, such as IoT devices and embedded systems, while still providing strong security.
- High Security: Despite its lightweight nature, LEA offers strong protection against a wide range of cryptanalytic attacks.
- Efficiency: LEA is highly efficient, both in terms of speed and memory usage, which makes it ideal for resource-constrained applications.
- Versatility: LEA supports multiple key lengths (128, 192, and 256 bits) and block sizes, making it adaptable to different security requirements.
- Simple Structure: LEA's relatively simple structure allows for easier implementation and analysis, while still maintaining robust security.

Standardization: LEA has been standardized by international organizations like the Korean Agency for Technology and Standards (KATS) and the International Organization for Standardization (ISO), ensuring widespread credibility and interoperability.

Table 1: Basic Details of the LEA Algorithm

Block Size	128 bits
Key Size	128, 192, 256 bits
Round Number	24, 28, or 32 (depending on key size)
Block Number	4 (each 32 bits)
Key Block Number	4, 6, or 8 (depending on key size)

III. RELATED WORK

With the continuous expansion of digital domains, protecting sensitive data has become increasingly crucial, particularly in environments with limited resources such as IoT devices and embedded systems. Traditional cryptographic methods, although reliable, often demand more resources than these applications can afford. This study tackles this issue by introducing a hybrid lightweight cipher that combines the strengths of CLEFIA and RECTANGLE ciphers. This hybrid model integrates the robust security features of CLEFIA with the flexible key scheduling of RECTANGLE, offering a well-balanced and efficient cryptographic solution. Lightweight ciphers are essential in these settings because they ensure data security without excessively taxing the limited computational resources, making them perfect for scenarios where power

consumption and memory usage are critical considerations. Idhalama & Oredo, [24] indicate IoT will face significant challenges, primarily focusing on the complexities of managing diverse devices and ensuring their interoperability. Jebrane & Lazaar, [25] suggested by 2020, the proliferation of tablets and smartphones, estimated to reach 7.3 billion units, has highlighted the need for standardized protocols to handle the vast data flow. With increasing IoT adoption, addressing issues like data privacy, power consumption, and efficient encryption becomes crucial. As various intelligent devices interconnect, the development of adaptive cryptographic solutions is necessary to mitigate potential security vulnerabilities [26]. Studies Mohammed et al., [27], Shannon, [17] indicate that the integration of smartphone technology with IoT has heightened security risks due to expanded network exposure.

In the study by Biryukov, [18] the observed that IoT devices often juggle multiple roles under limited resources, complicating the enforcement of robust security policies. To enhance device security, they proposed an agent-based model that separates IoT operations from device-specific functions, leveraging cloud infrastructure. This approach aims to create virtual replicas of IoT devices in the cloud, improving both functionality and security.

A detailed classification of existing security gaps within IoT infrastructure, communication channels, and cloud applications is presented [27]. This research reviews various proposed security strategies, identifying common challenges in securing cloud-based IoT systems. Similar to Thabit et al., [19]. Berisha & Kastrati, [6] provide a comprehensive analysis of IoT security issues, addressing legal and technical solutions pertinent to both private and public sectors.

Lubna et al. proposed a simple and highly secure encryption/ decryption (SHSED) methods depended on IDEA cipher. Besides shuffling byte, it used XOR logical operation, addition/subtraction mathematical operations to entering diffusion/ confusion to the entire system. The paper adapted variable key size and round values, Strong security with faster encryption times was provided for cloud cryptosystem [29].

Bao et al., [30], [31] adapted a small part of the original data as a key by using data partitioning and scrambling techniques. While shuffled the other parts based on this value. To access scrambled data saved in the cloud, the shuffling can only be undone using this key. Small random keys were generated instead of user-provided data [30]. Consequently, in 2017 Bao et al., [30] developed a new model. A random generator function used to create data rather than user provided. This small data might come from any gadget the user uses or could be generated at random. Once again, the data is jumbled since the small amount data gathered in this manner might not be random [31].

In study Alamari et al., [31] , proposed a new massage authentication algorithm in parallel mode. Two of PNGRs and substitutions box are employed for encrypting message and authenticating. Work average speedup enhanced by 2.99 over to the old work.

IV. PROPOSED ALGORITHM

The Proposed Algorithm to enhance cloud computing security with low processing overhead and high performance, a new hybrid lightweight cryptography algorithm is proposed. This algorithm combines the confusion and diffusion strengths of AES with the simple and efficient structure of the LEA algorithm. The result is a robust, secure encryption system, ideal for the cloud environment. The core of the system is based on the ARX (Addition-Rotation-XOR) structure of LEA, interleaved with the Substitution-Permutation Network (SPN) from AES. This design offers enhanced protection for data transmitted through untrusted channels.

4.1 New 3D Chaotic System:

The mathematical model for the 3D chaotic system can be defined as:

$$\dot{x} = a(y - x)$$
$$\dot{y} = bx - y - xz$$
$$\dot{z} = cz + xy - dz$$

Where:

- x, y, and z are the states of the system.
- a, b, c, and d are positive parameters.

For this system, you can choose parameter values from the original system and adapt them to the 3D context. Let's use:

- a=11a = 11a=11
- b=1.5b = 1.5b=1.5
- c=1.38c = 1.38c=1.38
- d=0.5d = 0.5d=0.5

These values are chosen to maintain some resemblance to the original system's chaotic behavior while simplifying it.

Initial Conditions:

Let the initial conditions be:

- x(0)=3.6x(0) = 3.6x(0)=3.6
- y(0)=1y(0) = 1y(0)=1
- z(0)=2.5z(0) = 2.5z(0)=2.5

4.2 Key Generation Phase

As we know, the strength of any cryptographic system lies in the security of its key. Therefore, generating a strong and dynamic key is crucial for the encryption process. The proposed system introduces a nonlinear, dynamic method for generating the encryption key, utilizing a chaotic system to ensure high variability and unpredictability in key generation.

The key generation phase leverages a new 3D chaotic system (adapted from a 5D model), governed by the following equations:

$$x˙=a(y−x)y˙=bx−y−xzz˙=cz+xy−dz$$

$$x˙y˙z˙=a(y−x)=bx−y−xz=cz+xy−dz$$

Where x, y, and z represent the state variables, and a,b, c, and d are positive parameters controlling the system's chaotic behavior. The specific parameters chosen for the key generation process are:

- a=11a = 11a=11
- b=1.5b = 1.5b=1.5
- c=1.38c = 1.38c=1.38
- d=0.5d = 0.5d=0.5

The initial conditions are set as:

- x(0)=3.6x(0) = 3.6x(0)=3.6
- y(0)=1y(0) = 1y(0)=1
- z(0)=2.5z(0) = 2.5z(0)=2.5

This 3D chaotic system generates a highly random and dynamic key stream, ensuring security across both the encryption and decryption processes. The generated key stream matches the bit size required by the proposed encryption algorithm, enhancing both security and performance. The chaotic system's sensitivity to initial conditions ensures that even the slightest changes produce entirely different key streams, further strengthening the encryption against potential attacks.

V. DYNAMIC PROPERTIES OF THE NEW 3D CHAOTIC SYSTEM

Using Mathematica, we calculated the Lyapunov exponents for the newly proposed 3D chaotic system, described by equation (1), with initial conditions (x (0), y (0), z (0)) = (3.6,1,2.5) and parameters (a, b, c, d) = (11,1.5,1.38,0.5) The results of the Lyapunov exponents are as follows:

$$\{L1=2.6915,2=0.1234, L3=−3.2548\} \quad (2)$$

The Kaplan-Yorke dimension of the 3D system is calculated by arranging the Lyapunov exponents in descending order, as:

$$\Sigma i=1 j\lambda i\geq 0 \quad (3)$$

From this, the Kaplan-Yorke dimension DKYD_{KY}DKY is calculated as:

$$DKY=2+L1+L2|L3|=2.86 \quad (4)$$

This value shows that the system exhibits chaotic behavior with two positive Lyapunov exponents, which indicates strong nonlinearity and sensitivity to initial conditions. The relatively high value of the largest Lyapunov exponent, L1=2.6915, confirms that the system has a rapid exponential divergence in its trajectories, making it highly suitable for applications in cryptographic systems.

5.1 Analysis of the New 3D Chaotic System

In a 3D chaotic system, the complexity of the system's behavior can be gauged through its Lyapunov exponents. Having two positive exponents signifies that the system's dynamics can expand in two distinct directions within the phase space. This characteristic, combined with the large positive Lyapunov exponent, ensures that the system exhibits fast, unpredictable behavior, which is essential for cryptographic applications.

Additionally, the Kaplan-Yorke dimension measures the complexity of the system's attractor [33]. A value close to 3 reflects a highly complex attractor, indicating intricate and robust chaotic behavior. For a 3D chaotic system to be effective in cryptographic systems, the following properties are critical:

1. Two positive Lyapunov exponents, ensuring the chaotic system spreads in two independent directions within the phase space.
2. A large positive Lyapunov exponent, guaranteeing fast and exponentially growing divergence in the phase space.
3. A high Kaplan-Yorke dimension, indicating that the system's attractor is sufficiently complex to support secure cryptographic applications.

Using Mathematica, we demonstrated the chaotic behavior and dynamic properties of this new 3D system. As shown in Figure 4, even minor variations in initial conditions, such as a small change in xxx, lead to significantly different results. Figure 5 shows the system's chaotic and random behavior, confirming that it is well-suited for cryptographic use.

Figure 4: The dynamic properties of the new 3D system.

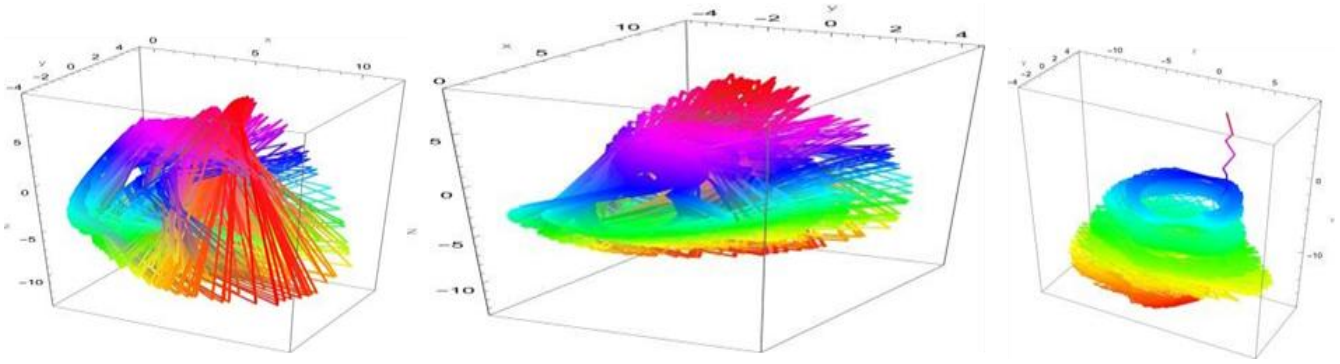
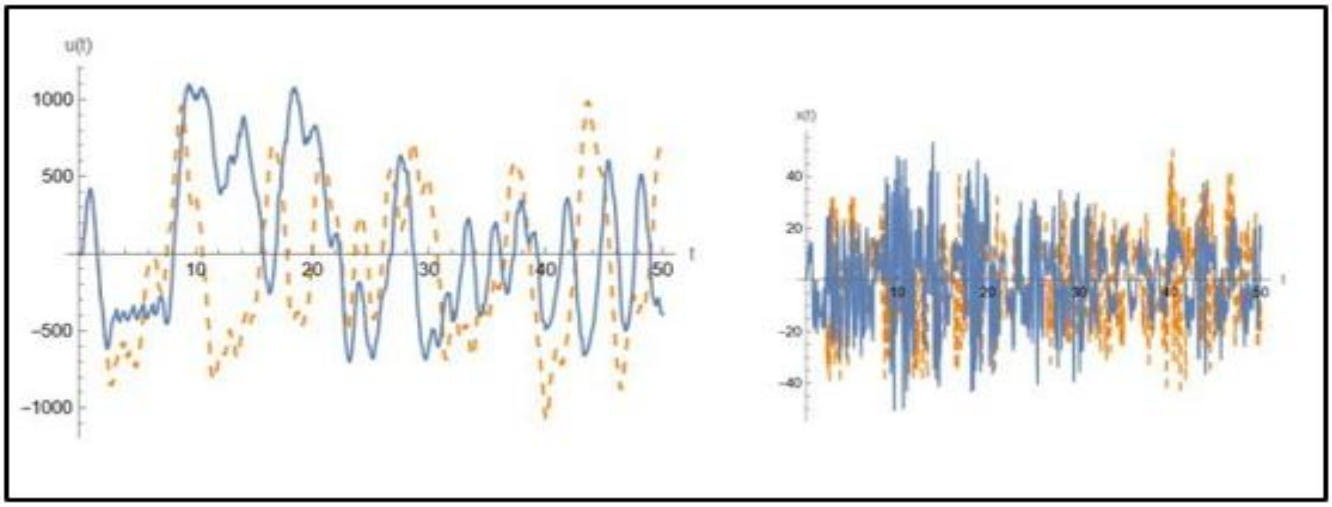


Figure 5: The system's chaotic and random behavior.



5.2 Encryption Phase

This section presents a proposed method for enhancing data security in a cloud environment by hybridizing two lightweight cryptographic algorithms. The purpose of this hybrid approach is to combine the strengths of both LEA (Lightweight Encryption Algorithm) and AES (Advanced Encryption Standard) to maximize security. Specifically, we leverage LEA's simplicity and high security, while incorporating AES's four S-boxes and permutation stages for additional robustness.

First Stage: LEA Algorithm

In this hybrid model, the enhanced LEA algorithm processes 128-bit data blocks, divided into four 32-bit sub-blocks. The LEA algorithm operates with the ARX structure, which includes modular Addition, bitwise Rotation, and bitwise XOR operations. The basic parameters of the proposed Z-LEA algorithm are outlined in Table 2.

Table 2: Basic Parameters of the Z-LEA Algorithm

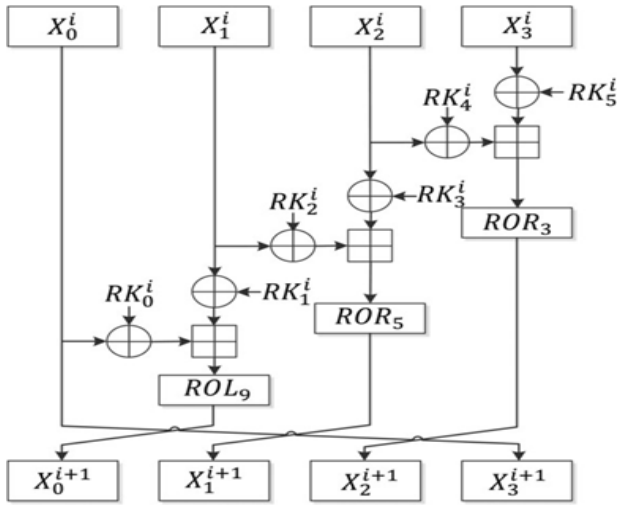
Block Size	128 bits
Key Size	128 bits
Block Number	4 (each 32 bits)
Round Number	7

The encryption process begins by dividing the data stream into four blocks (x_0^0, x_1^0, x_2^0, x_3^0), each 32 bits in size. These blocks are XORed with the chaotic key values generated by the 3D chaotic system. After this initial XOR step, the core LEA function is applied as shown in Figure 4. The core steps of the algorithm are as follows:

1. Bitwise XOR: Each block of plain text is XORed with the corresponding key bit (chaotic key).
2. Modular Addition: Addition is applied between two adjacent blocks (e.g., X_0^0 and X_1^0).
3. Bitwise Rotation: The result of the addition is rotated either left or right by 5, 9, or 3 bits.

These operations are repeated for the number of rounds specified by the encryption algorithm. Each round strengthens the security by further obfuscating the data

with chaotic key values and ARX operations.



Second Stage: AES Algorithm

After processing the data blocks through the first stage (LEA), the next phase involves applying the Advanced Encryption Standard (AES) sub-steps to further enhance the security. The processed data from the LEA stage is formed into an array of 16 elements, which then undergoes the following AES transformations [32].

SubBytes

In this step, substitution is applied to each byte of the data block. Each byte is replaced with a new one using a pre-built substitution table known as the S-box. This 8-bit lookup table swaps out each byte from the LEA stage for another based on its corresponding row/column values. The substitution is represented by the following operation:

bij=S(aij)...(5)b_{ij} = S(a_{ij}) \quad \text{...(5)}bij=S(aij)...(5)

The substitution process ensures that no byte is replaced by itself, providing further complexity and confusion to the data. This step is visualized in the figure below.

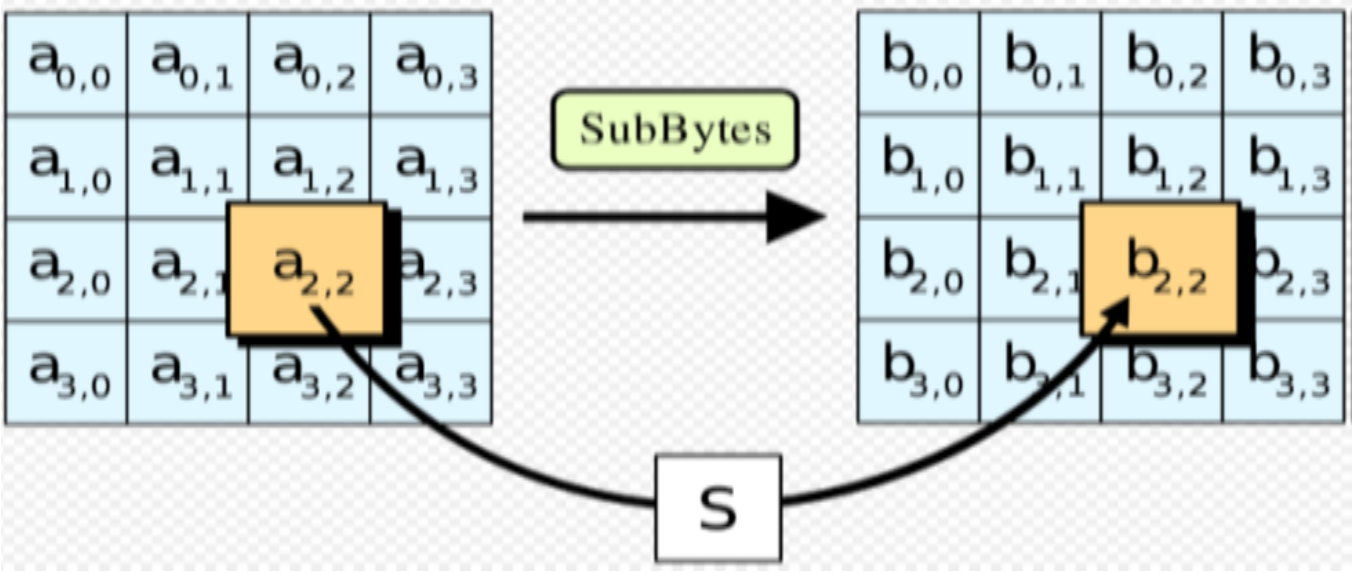


Figure 6: 8-bit lookup table (S-box) used in the SubBytes step.

ShiftRows

This step rearranges the rows of the array by shifting them to the left by a certain number of positions. Each row is shifted according to the following rules [30,31]:

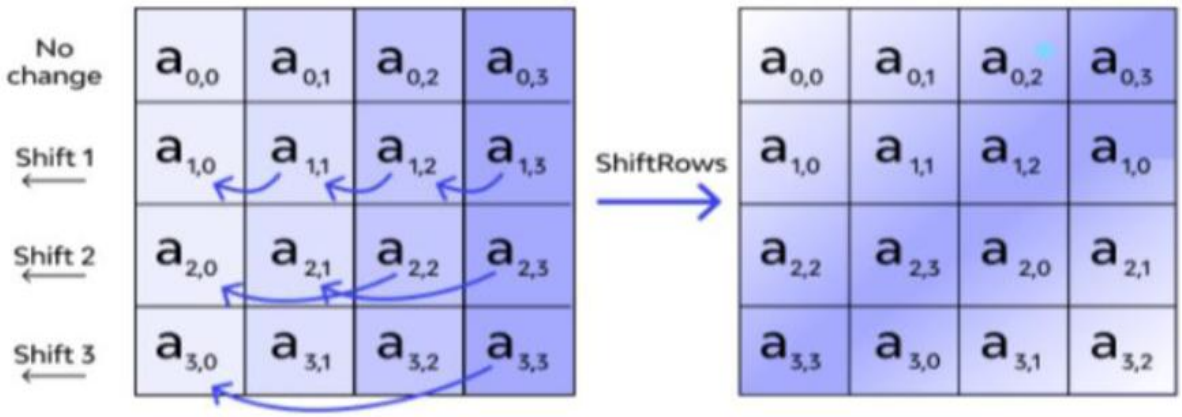
- Row 1: No shift.
- Row 2: Shift left by 1 position.
- Row 3: Shift left by 2 positions.
- Row 4: Shift left by 3 positions.

This operation is represented as follows:

$ShiftR(State)$...(6)

As seen in equation (6):

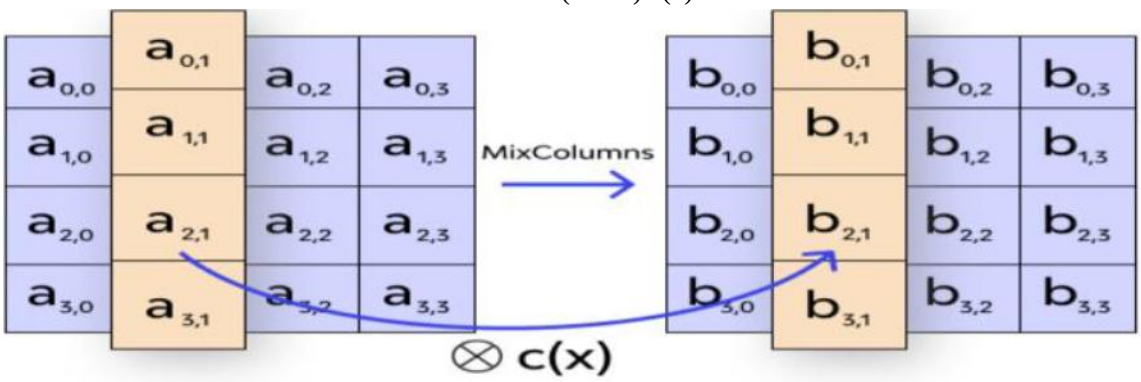
$$ShiftR(State)\text{...(6)}$$



MixColumns

In this step, each column undergoes a transformation using matrix multiplication. The columns of the data are multiplied by a fixed matrix, which modifies each byte's location within the column. This transformation spreads the influence of each input byte across multiple output bytes, enhancing diffusion. The matrix multiplication can be represented as follows:

$$MixColu(State)...(7)$$



By introducing the AES transformations of SubBytes, ShiftRows, and MixColumns, the system significantly increases the complexity of the encrypted data stream. Each step adds layers of confusion and diffusion, making it exceedingly difficult for attackers to reverse-engineer the original message. This combination of Z-LEA and AES strengthens the overall cryptographic system, providing robust protection against various attacks, including differential and linear cryptanalysis [34], [35].

Fourth Stage: LEA Algorithm

In this final stage, the procedures from the first LEA stage are repeated with a reduced number of rounds. As LEA is a lightweight cryptography (LWC) method, it uses predefined round numbers based on the key space. In this stage, the number of rounds is reduced to balance security and efficiency.

In Z-LEA, five rounds are selected from the first stage, and five rounds are chosen from the third stage. The goal of reducing the number of rounds is to minimize computational overhead while maintaining strong security. The 10th round of the LEA algorithm helps increase efficiency by reducing the computation time of the encryption process.

At the same time, introducing this intermediate stage ensures that confusion and diffusion are thoroughly dispersed throughout the system, preserving the overall strength of the encryption system. The structure and stages of the proposed Z-LEA system are illustrated in Figure 8.

Decryption

To decrypt the data, the exact reverse of the encryption process is performed. All the previously described procedures are repeated in reverse order, using the same number of rounds and the encryption key generated by the

3D chaotic system. This ensures the secure recovery of the original plaintext data.

Parallelism

Parallelization can be applied to both the encryption and decryption processes to enhance performance. This approach enables different parts of the encryption algorithm to run simultaneously on multiple cores or processors, significantly speeding up the operations.

In the Z-LEA system, encryption is executed twice in a single pass, with each core handling a portion of the data. For example, one core could process 128 bits of data, 128 bits of the key, and 10 rounds while applying all encryption stages in parallel. This reduces the time required for the encryption process.

Figure 7, 8 illustrates the parallelism process in Z-LEA. Similarly, decryption can be performed using the same parallelization technique to optimize the system's performance.

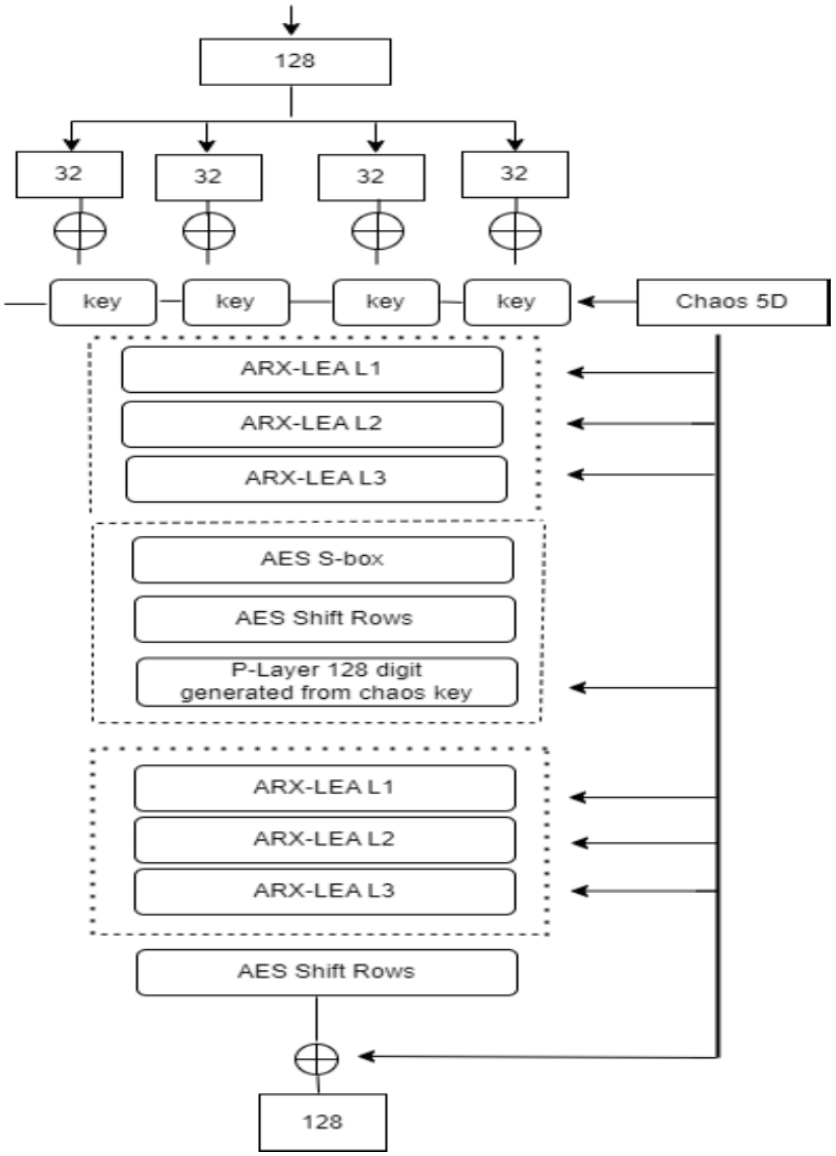


Figure 7: The structure and stages of the proposed Z-LEA system for 128

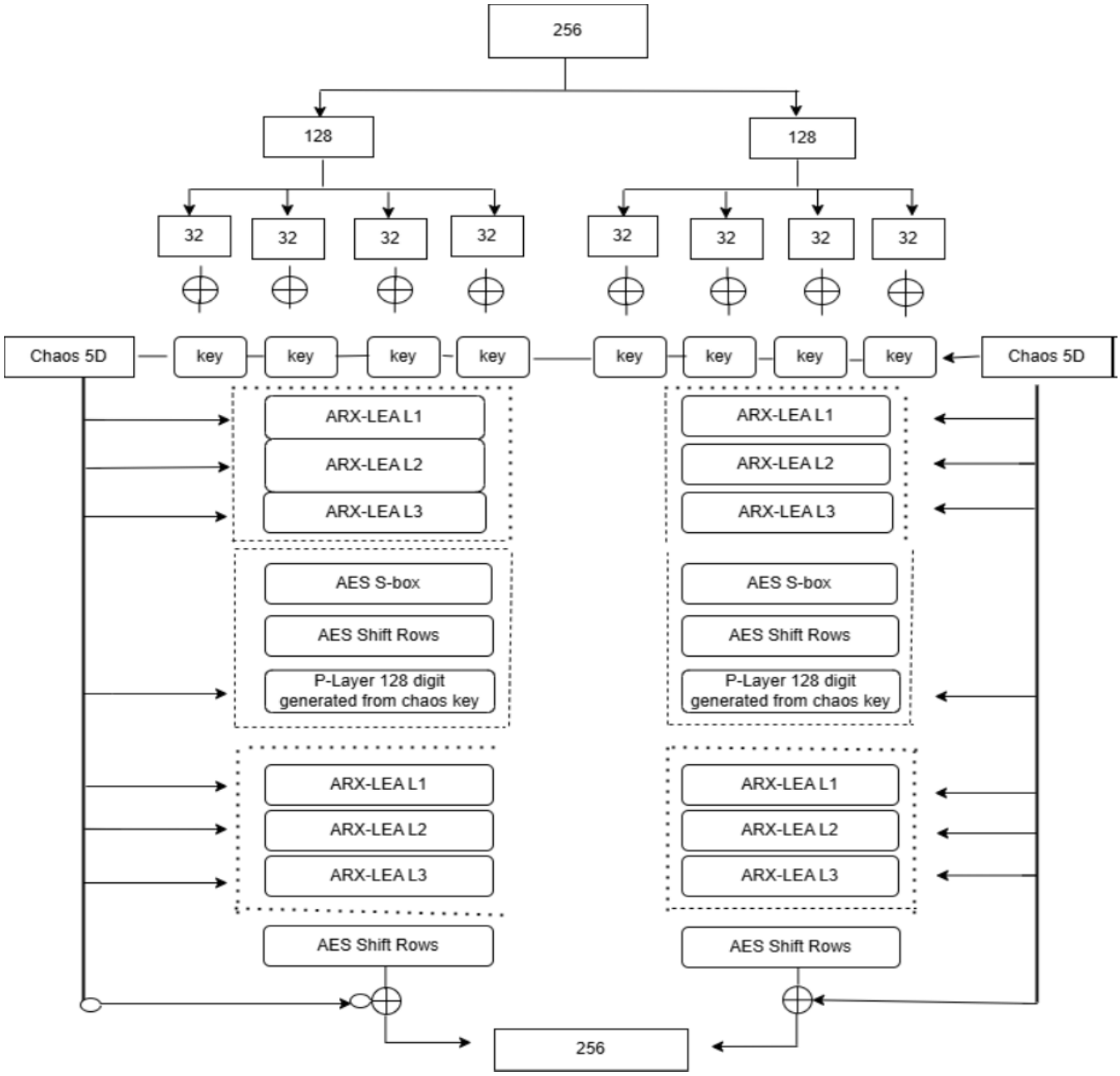


Figure 8: The structure and stages of the proposed Z-LEA system for 256

VI. RESULTS AND DISCUSSION

A. Experimental Environment

This section presents the practical implementation of the proposed Z-LEA algorithm and an analysis of its effectiveness compared to various symmetric lightweight cryptographic techniques. The cloud network framework used for the experimental environment involves a client accessing virtual machines hosted on a Xen Server (version 6.1a).

The server has a Core i7 processor (4.8 GHz) with 16 GB of RAM, and the client uses a Core i3 processor with 4 GB of RAM. Parallelism in the encryption process is a crucial component of this experiment. The aim is to handle large data efficiently by employing parallel processing across multiple cores, ensuring high encryption rates even with substantial datasets.

The results of this implementation are expected to demonstrate the superior performance of the Z-LEA algorithm in real-world cloud computing applications,

aided by the use of parallel processing. Figure 10 shows the Python code used to implement Z-LEA.

B. Comparative Analysis of Parameters

This section provides a comparative analysis of various lightweight symmetric cryptographic algorithms executed in a steady parallel environment. Table 3 presents key features of popular LWC cryptographic techniques, including Present, AES, Simon, SIMECK, RC5, RC6, HEIGHT, RECTANGLE, SEA, and Blowfish, alongside the recently proposed Z-LEA algorithm.

Table 3. A Comparison among some Symmetric Algorithms

Algorithm	Structure	Blocksize	Keysize	Round No.	Possible key	Mathematical Operations	S-P Structure	S-Box Size	Security Rate
Present [36]	SPN	128 bits	64, 128 bits	24	2^64, 2^128 bits	Feistel	8 S-Boxes	8 * 8 (8 bits)	Secure
AES [37]	SPN	128 bits	128, 192, 256 bits	10, 12, 14	2^128, 2^192, 2^256 bits	XOR, Mixing, Substitution, Shifting, Multiplication, Addition	1 S-Box	16 * 16 (16 bits)	Secure
Simon/SIMECK [38]	FN	128 bits	128 bits	20, 27, 35	2^128 bits	ARX function, Rotational- XOR	N/A	N/A	Secure
RC5 [39]	FN	32 bits	128 bits	12	2^128 bits	XOR, Subtraction, Multiplication, Shifting	N/A	N/A	Secure
RC6 [40]	FN	32 bits	64 bits	Variable	2^64 bits	Addition, Subtraction, XOR, Left/Right Rotation	4 S-Boxes	4 * 4 (4 bits)	Secure
HEIGHT [41]	GFS	64 bits	128 bits	32	2^128 bits	XOR, Addition mod 2	N/A	N/A	Secure
RECTANGLE [42]	SPN	64 bits	128 bits	25	2^128 bits	AddRoundKey, SubColumn, ShiftRow	1 S-box	4-bit	Secure
SEA [43]	SPN	16 bits	256 bits	4	2^256 bits	OR, Rotations, Mod 2 Addition, Substitution	N/A	N/A	Secure
LED [44]	FN	64, 128 bits	64, 128 bits	Variable	2^64, 2^128 bits	XOR, Rotations, Mod 2 Addition, Substitution	1 S-Box	6 * 6 (6 bits)	Secure
Blowfish [45]	FN	64 bits	32–448 bits	16	2^32–2^448 bits	XOR, Shifting, Mixing, Substitution	4 S-Boxes	8 * 4 (32 bits)	Secure
Proposed Algorithm (Z-LEA)	SPN	256 bits	128, 192, 256 bits	10	2^128, 2^256 bits	XOR, ARX function, Substitution	4 S-Boxes	4 * 4 (4 bits)	Highly Secure

Table 4 outlines the flexibility, architectural design, security strengths, and limitations of these algorithms. Z-LEA stands out for offering strong security with minimal computational overhead, making it highly suitable for a wide range of cryptographic applications.

The comparative analysis demonstrates how Z-LEA excels in terms of both security and efficiency. The architecture of Z-LEA is designed to minimize computational complexity and resource requirements, using only 10 rounds of processing. Simple mathematical operations (ARX) are applied in each round, ensuring that the algorithm remains lightweight while maintaining robust security.

The 3D chaotic system used for dynamic key generation in Z-LEA is another major advantage. This approach avoids mathematically complex procedures, relying on fundamental arithmetic operations that do not introduce unnecessary overhead. Unlike RC6, which uses complex multiplication, Z-LEA leverages simple ARX operations for enhanced performance.

Furthermore, Z-LEA incorporates S-boxes to enhance diffusion, though it avoids using large S-boxes like those in AES. The smaller 4x4 S-box used in Z-LEA saves both time and memory compared to Present's 8x8 S-box and AES's 16x16 S-box.

Table 4. 3D Chaotic System Values

T = time	X	Y	Z
T = 1	1.030747167	0.1312896	2.816411
T = 2	0.574923075	0.34748328	2.224571
T = 3	0.228897078	0.25912469	1.614655
T = 4	2.88204096	-3.4871125	-4.72783
T = 5	1.136513114	-1.6069506	0.012611
T = 6	-0.04384695	-0.8707766	-0.35863

The Z-LEA algorithm's use of fewer rounds (5 rounds for encryption and 5 for decryption) contributes to its efficiency while ensuring strong security. This feature makes Z-LEA especially suitable for handling large amounts of data, as it processes two 128-bit data blocks simultaneously, encrypting them in a fraction of the time required by other LWC algorithms.

C. Experimental Evaluation

The proposed Z-LEA algorithm utilizes a 3-dimensional chaotic system as a dynamic key generator, producing highly random key streams. A sample of 20 values produced by this new chaotic system is shown in Table 5. The X dimension is selected for key generation during the encryption and decryption processes. The uniqueness of the key values in each instance further demonstrates the advantage of using the chaotic system as a key generator, as it significantly increases the difficulty of predicting the correct key.

Table 5. A sample of 20 values produced by this new chaotic system

Index	X Dimension Value	Index	X Dimension Value
1	0.736492817345	11	0.284615937402
2	0.218374965821	12	0.769203845617
3	0.984615372046	13	0.503847162930
4	0.457193826504	14	0.138947205684
5	0.129584736291	15	0.962817345092
6	0.873615204987	16	0.421659283746
7	0.392748561930	17	0.689472038615
8	0.651947283615	18	0.074615283940
9	0.047381926504	19	0.847293615028
10	0.915273648120	20	0.316584720193

Table 5 presents samples of 20 values generated from the X dimensions of the proposed 3D chaotic system. The values exhibit strong randomness, non-linearity, and non-periodicity, confirming the suitability of the chaotic system as a dynamic key generator. The diversity and unpredictability of the generated sequences significantly enhance the cryptographic strength of the Z-LEA algorithm and increase resistance to statistical, differential, and brute-force attacks.

Table 6 provides the results of statistical tests using the NIST framework, which evaluates the randomness of the binary key stream produced by the 3D chaotic system. The National Institute of Standards and Technology (NIST) randomness test measures the proportion of randomness in the key stream, with the default P-value set at 0.01. A P-value greater than 0.01 indicates that the key stream is random, while a smaller value suggests non-randomness.

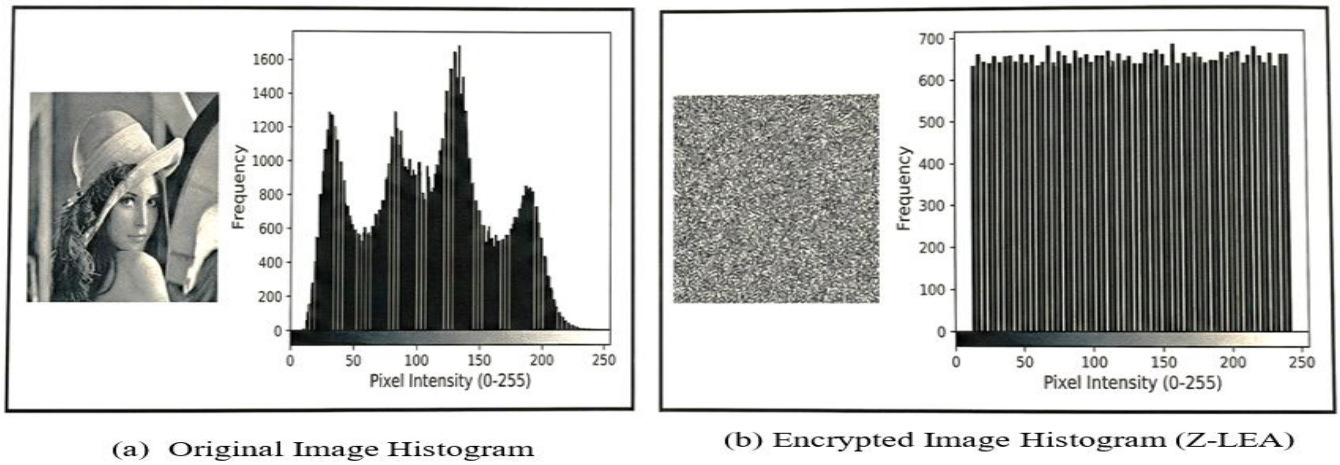
Table 6. Nist Experiment Results of Z-LEA

No.	NIST Statistical Test Name for Chaotic Key	P-Value	Result
1	Frequency Test (Monobit)	0.9619	Random
2	Frequency	0.49817	Random
3	Runs	0.5878	Random
4	Longest Run	0.61479	Random
5	Binary Matrix Rank Test	0.0391	Random
6	Discrete Fourier Transform (Spectral) Test	0.6617	Random
7	Non-Overlapping Template Matching Test	0.2255	Random
8	Overlapping Template Matching Test	0.02073	Random
9	Maurer's Universal	0.4110	Random
10	Linear Complexity Test	0.8866	Random
11	Serial	0.5954	Random
12	Cumulative Sums (Forward) Test	0.19040	Random
13	Cumulative Sums (Reverse) Test	0.19040	Random
14	Random Excursions Test	0.3993	Random
15	Random Excursions Variant Test	0.4414	Random

For performance evaluation, Z-LEA is tested using multiple standard cryptographic assessments, including the Avalanche Test, Correlation Analysis, Time Complexity, and Execution Time (Encryption and Decryption Time). The results demonstrate that the proposed algorithm performs well in all these tests, making it suitable for use in various environments, particularly in cloud computing, which is the target application of this study.

VII. IMAGE HISTOGRAM

The image histogram provides insight into the distribution of randomly encrypted images. It displays how encryption alters the intensity of the data in an image, showing the difference between encrypted and unencrypted images. This metric helps to analyze the average intensity shift, which is caused by encryption. Figure 9 illustrates the fluctuation in the encrypted image's histogram using Z-LEA compared to the original distribution of the image.



(a) Original Image Histogram (b) Encrypted Image Histogram (Z-LEA)

Figure 9. Illustrates the Encrypted

Image Entropy

Image entropy measures the randomness and unpredictability introduced by the encryption process. A higher entropy difference between the plain and cipher images indicates a more effective encryption method. The formula to calculate image entropy is:

$$E=\Sigma i=1(log\ 2(Xi))...(6)$$

Where:

- E is the image entropy,
- X is the probability of a specific intensity level in the image,
- N is the total number of intensity levels.

Table 7 shows the entropy values for various images, such as "Baboon", "Lena", and "Peppers". The Z-LEA algorithm achieves an average entropy increase of 9.92%, demonstrating its effectiveness in image encryption.

Table 7. IMAGE ENTROPY TEST FOR Z-LEA

No.	Image	Dimension	Entropy (ENC)	Entropy (ORG)
1	Baboon	128 * 128	7.9891	7.2608
		220 * 220	7.9958	7.1662
		256 * 256	7.9973	7.2091
2	Lena	128 * 128	7.9885	7.4810
		220 * 220	7.9962	7.4618
		256 * 256	7.9970	7.4436
3	Banda	256 * 256	7.9969	7.5966
		512 * 512	7.9982	7.5217
4	Peppers	256 * 256	7.9970	7.5519
		512 * 512	7.9992	7.5555

Correlation Analysis

This analysis evaluates the correlation between the original image and the encrypted image. A strong encryption algorithm results in encrypted data with little to no correlation to the original data. The correlation can be calculated using the following formula:

$$\begin{cases} r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \\ cov\,(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E\,(x))\,(y_i - E\,(y)) \\ D\,(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E\,(x))^2 \\ E\,(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \end{cases}$$

Where:

- N is the number of pixel pairs,
- x and y are the gray values of two adjacent pixels,
- E(x) is the mean of x,
- D(x) is the variance of x,
- Cov(x, y) is the covariance of x and y.

Table 8 present the results of the correlation analysis for the "Lena" image, showing a significant decrease in correlation between adjacent pixels after encryption.

Table 8. Results for Correlation

Image	Size	Correlation (Original)	Correlation (Encrypted)
Baboon	256 * 256	0.9000	0.0026
Banda	256 * 256	0.9764	0.0012
Lena	256 * 256	0.9576	0.0055
Peppers	256 * 256	0.9309	-0.0031

Execution Time

Execution time is a crucial metric for evaluating the performance of an encryption algorithm, alongside its security features. It is defined as the total time required to encrypt and decrypt a specific dataset. Table 9 lists the total encryption and decryption times (in milliseconds) for various images, highlighting the speed of the Z-LEA lightweight algorithm.

Table 9. Results for Execution Time

No.	Image	Image Size	Total Enc/Dec Time (ms)
1	Baboon	128	1.1810
		220	3.7104
		256	4.6078
2	Lena	128	1.2910
		220	3.4531
		256	5.0388
3	Peppers	256	6.0214
		512	25.220
4	Banda	256	4.7363
		512	19.880

Avalanche Test (Key Sensitivity)

The avalanche test is used to evaluate the sensitivity of an encryption algorithm to small changes in the key or plaintext. This test measures how a slight modification in the key stream (such as changing a single bit) affects the ciphertext. Z-LEA is particularly sensitive to such changes due to its use of the 3D chaotic system, which ensures that even small variations result in significantly different ciphertexts.

Table 10 provides the results of the avalanche test, showing that Z-LEA achieves a bit shift of 52.56% and 46.71%, respectively, when a single bit is changed in the key or plaintext. By comparison, AES achieves a bit shift of 45.92%.

Table 10**.** Provides the avalanche test results.

No	plaintext	KEY	Key Cipher	cipher- text	Avalanche Analysis
1	0A BB CB CC DF 22 AF AC 00 00 00 55 FF AA BB FF 0A BB CB CC DF 22 AF AC 00 00 00 55 FF AA BB FF	FF FF BB AA 00 00 00 00 00 BB 00 00 AA AA AA AA FF FF BB AA 00 00 00 EE 00 BB 00 00 AA AA AA AA	42 42 29 40 4A 27 4A 27 BC E5 4A 27 E6 80 E6 80 42 42 29 40 4A 27 AE 29 BC E5 4A 27 E6 80 E6 80	B8 37 5C CC 29 E4 98 16 F5 6B FF 3A 9A 38 90 2B B8 37 80 38 29 E4 84 3A F5 6B 81 A2 9A 38 35 71	0.0363
2	0A 0B 0C 0D 0F 01 02 03 04 05 06 07 08 09 1A 2B 0A 0B 0C 0D 0F 01 02 03 04 05 06 07 08 09 1A 2B	01 02 04 05 06 AA BB CC 44 DD EE 88 09 04 05 06 01 02 04 05 06 AA BB CB 44 DD EE 88 09 04 05 06	4F 29 4C 71 D3 AB 29 D0 EB 79 AC 69 A2 73 AC 7B 4F 29 4C 71 D3 AB 26 99 EB 79 AC 69 A2 73 AC 7B	64 25 87 52 81 32 D6 63 A6 0D CF 1D FD 67 E5 30 B8 37 5C CC 29 E4 98 16 F5 6B FF 3A 9A 38 90 2B	0.0472
3	BB BB CC CC DD DD EE EE FF FF AA AA AB AC AD AF BB BB CC CC DD DD EE EE FF FF AA AA AB AC AD AF	FF FF FF FF FF FF FF FF AA AA AA AA AA AA AA AA FF FF FF FF FF FF FF F7 AA AA AA AA AA AA AA AA	42 42 42 42 42 42 42 42 E6 80 E6 80 E6 80 36 80 42 42 42 42 42 42 42 40 E6 80 E6 80 E6 80 36 80	38 11 DA 6A 8F 51 9A 82 86 27 75 0E A7 F2 26 78 38 11 D6 AB 8F 51 8D 84 86 27 85 7B A7 F2 26 47	0.0334

4	FF FF FF FF FF FF FF FF AA AA AA AA AA AA AA AA FF FF FF FF FF FF FF FF AA AA AA AA AA AA AA AA	BB BB CC CC DD DD EE EE FF FF AA AA AB AC AD AF BB BB CC CC DD DD EE 07 FF FF AA AA AB AC AD AF	26 95 E4 3B C9DA 63B4 42 42 E6 80 2A 1F B9 82 26 95 E4 3B C9DA 6C46 42 42 E6 80 2A 1F B9 82	A3 7F 08 3F C1 EA F9 7B 9B 7F 8C 0A 12 D5 C7 EE A3 7F 0F 0A C1 EA F5 78 9B 7F 4D C0 12 D5 16 E9	0.0366

VIII. CONCLUSION

This study presents a novel lightweight encryption algorithm for enhancing cloud data security, named Z-LEA. Several improvements have been made to the original LEA algorithm to make it more suitable for protecting outsourced data in semi-trusted cloud channels. Z-LEA applies five rounds of LEA in the first stage, integrates AES sub-steps (SubBytes, ShiftRows, MixColumns), and then concludes with five additional rounds of LEA.

The proposed Z-LEA algorithm demonstrates superior qualities in terms of confusion and diffusion compared to alternative methods. Notably, Z-LEA achieves this with only 10 rounds, making it computationally lightweight and efficient. The algorithm utilizes simple arithmetic operations in each round, which helps reduce the encryption workload without compromising security. This efficiency is further enhanced by the dynamic key generation of the 3D chaotic system, which strengthens the encryption by ensuring the key stream is highly unpredictable and resistant to attacks.

Our approach to lightweight cryptography emphasizes simplicity and efficiency. Z-LEA avoids computationally complex methods, instead relying on ARX functions, which are characterized by their straightforward mathematical operations. In addition, Z-LEA uses a smaller 4x4 S-box instead of larger S-boxes, which saves memory and processing time, while still providing robust security. The use of five encryption and decryption rounds strikes an optimal balance between performance and security, significantly improving the overall efficiency of the algorithm.

Z-LEA leverages parallelism to handle two separate 128-bit data streams simultaneously, enabling it to process large amounts of data quickly. This parallelism, combined with the reduced number of rounds, makes Z-LEA highly suitable for applications that require both security and efficiency. It has demonstrated resilience against differential and brute force attacks, while also exhibiting low memory usage and minimal computational overhead.

In conclusion, Z-LEA offers a strong and efficient solution for cloud data security, with a focus on simplicity, performance, and the ability to handle large datasets in a short amount of time. Its lightweight design and security features make it a versatile choice for a wide range of cryptographic applications.

REFERENCES

[1] B. Ghosh, I. K. Dutta, M. Totaro, and M. Bayoumi, “A Survey on the Progression and Performance of Generative Adversarial Networks,” in 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE., 2020, pp. 1–8.

[2] V. A. Thakor, M. A. Razzaque, A. D. Darji, and A. R. Patel, “A novel 5-bit S-box design for lightweight cryptography algorithms,” J. Inf. Secur. Appl., vol. 73, p. 103444, 2023, doi: https://doi.org/10.1016/j.jisa.2023.103444 Available.

[3] A. Fotovvat, G. M. E. Rahman, Graduate, S. S. Vedaei, and K. A. Wahid, “Comparative Performance Analysis of Lightweight Cryptography Algorithms for IoT Sensor Nodes,” IEEE Internet Things J., vol. 8, no. 10, pp. 8279–8290, 2021.

[4] Z. A. Mohammed and K. A. Hussein, “Lightweight Cryptography Concepts and Algorithms: A Survey,” in Second International Conference on Advanced Computer Applications (ACA). IEEE, 2023, pp. 1–7. doi: 10.1109/ACA57612.2023.10346914.

[5] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," J. Cyber Secur. Technol., vol. 00, no. 00, pp. 1–15, 2018, doi: 10.1080/23742917.2017.1384917.

[6] A. Berisha and H. Kastrati, "Parallel Implementation of RC6 Algorithm," J. Comput. Sci. Technol. Stud., vol. 3, no. 2, pp. 1–9, 2021, doi: 10.32996/jcsts.2021.3.2.1.

[7] A. M. Al-Hejri et al., "A hybrid explainable federated-based vision transformer framework for breast cancer prediction via risk factors," Sci. Rep., vol. 15, no. 1, p. 18453, May 2025, doi: 10.1038/s41598-025-96527-0.

[8] H. K. Zghair, S. A. Mehdi, and S. B. Sadkhan, "Bifurcation of Novel Seven-Dimension Hyper Chaotic System," J. Phys. Conf. Ser., vol. 1804, no. 012051, pp. 1–13, 2021, doi: 10.1088/1742-6596/1804/1/012051.

[9] Z. H. Thabit, S. A. Mehdi, and B. M. Nema, "Enhancing Color Image Security: Encryption with Dynamic Chaotic Three-Dimensional System and Robust Security Analysis," Al-Mustansiriyah J. Sci., vol. 34, no. 4, pp. 87–95, 2023, doi: https://doi.org/10.23851/mjs.v34i4.1411.

[10] M. Tawfik, N. M. Al-Zidi, B. Alsellami, A. M. Al-Hejri, and S. Nimbhore, "Internet of Things-Based Middleware Against Cyber-Attacks on Smart Homes using Software-Defined Networking and Deep Learning," in 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), IEEE, Dec. 2021, pp. 7–13. doi: 10.1109/ICCMST54943.2021.00014.

[11] K. A. Hussein and T. B. Kareem, "Proposed Parallel Algorithms to Encryption Image Based on Hybrid Enhancement RC5 and RSA," in Fifth International Engineering Conference on Developments in Civil & Computer Engineering Applications 2019 - (IEC2019) -, IEEE, 2019, pp. 101–106.

[12] A. M. Al-Hejri, N. M. Al-Zidi, M. Tawfik, A. Albakhrani, and A. H. Sable, "A Facilitation System for Arabic Foreigners in India Using the Web and Android System," in 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, Mar. 2022, pp. 239–245. doi: 10.1109/ICACCS54159.2022.9785022.

[13] Z. A. Mohammed and K. A. Hussein, "PRC6: Hybrid Lightweight Cipher for Enhanced Cloud Data Security in Parallel environment," Secur. Priv., vol. 7, no. 5, p. e413, 2024, doi: https://doi.org/10.1002/spy2.413.

[14] H. Delfs and H. Knebl, "Symmetric-Key Cryptography," in Information Security and Cryptography, Springer, 2015, pp. 11–47. doi: https://doi.org/10.1007/978-3-662-47974-2_2.

[15] A. Braeken, "Public key versus symmetric key cryptography in client – server authentication protocols," Int. J. Inf. Secur., vol. 21, no. 1, pp. 103–114, 2021, doi: 10.1007/s10207-021-00543-w.

[16] A. S. Pannwar, "ASYMMETRIC KEY CRYPTOGRAPHY," 2014. doi: http://dx.doi.org/10.2139/ssrn.2380622.

[17] N. Bansal and S. Singh, "RSA Encryption and Decryption System," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 6, no. 5, pp. 109–113, 2020, doi: https://doi.org/10.32628/CSEIT206520.

[18] R. Imam, Q. M. Areeb, A. Alturki, and F. Anwer, "Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status," IEEE Access, vol. 9, pp. 155949–155976, 2021, doi: 10.1109/ACCESS.2021.3129224.

[19] C. E. Shannon, "Communication Theory of Secrecy Systems*," Bell Syst. Tech. J., vol. 28, no. 4, pp. 656–715, 1948, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[20] A. Biryukov, "Substitution–Permutation (SP) Network," in ENCYCLOPEDIAOF CRYPTOGRAPHYANDSECURITY, H. C. A. van Tilborg, Ed., Springer, 2011, p. 602. doi: https://doi.org/10.1007/978-1-4419-5906-5_619.

[21] T. P. Berger, J. Francq, M. Minier, and G. Thomas, "Extended Generalized Feistel Networks using Matrix Representation to Propose a New Lightweight Block Cipher: L ILLIPUT," IEEE Trans. Comput., vol. 65, no. 7, pp. 1–15, 2015, doi: 10.1109/TC.2015.2468218.

[22] M. Togan and C. Plesca, "Comparison-Based Computations Over Fully Homomorphic Encrypted Data," in 10th international conference on communications (COMM). IEEE Access, 2014, pp. 1–6.

[23] O. U. Idhalama and J. O. Oredo, “Exploring the next generation Internet of Things (IoT) requirements and applications: A comprehensive overview,” Inf. Dev. Sage, pp. 1–16, 2024, doi: 10.1177/02666669241267852.

[24] J. Jebrane and S. Lazaar, “General Letters in Mathematics (GLM) A performance comparison of lightweight cryptographic algorithms suitable for IoT transmissions,” Gen. Lett. Math., vol. 10, no. 2, pp. 46–53, 2021, doi: 10.31559/glm2021.10.2.5.

[25] F. Thabit, S. Alhomdy, and S. Jagtap, “Security analysis and performance evaluation of a new lightweight cryptographic algorithm for cloud computing,” Glob. Transitions Proc., vol. 2, no. 1, pp. 100–110, 2021, doi: 10.1016/j.gltp.2021.01.014.

[26] Z. A. Mohammed, H. Q. Gheni, Z. J. Hussein, and A. K. M. Al-qurabat, “Advancing Cloud Image Security via AES Algorithm Enhancement Techniques,” Eng. Technol. Appl. Sci. Res. J., vol. 14, no. 1, pp. 12694–12701, 2024, doi: https://doi.org/10.48084/etasr.6601 ABSTRACT.

[27] A. M. Al-Hejri, R. M. Al-Tam, A. H. Sable, B. Almuhaya, S. S. Alshamrani, and K. M. Alshmrany, “A hybrid vision transformer with ensemble CNN framework for cervical cancer diagnosis,” BMC Med. Inform. Decis. Mak., vol. 25, no. 1, p. 411, Nov. 2025, doi: 10.1186/s12911-025-03250-x.

[28] F. Thabit, O. Can, S. Alhomdy, G. H. Al-gaphari, and S. Jagtap, “A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing,” Int. J. Intell. Networks, vol. 3, pp. 16–30, 2022, doi: 10.1016/j.ijin.2022.04.001.

[29] L. M. Al-ramini, “Implementation of proposed lightweight cryptosystem for use in Cloud Computing Security,” Middle East University, 2018.

[30] S. Bao, Y. Lu, Y. Yang, C. Wang, M. Chen, and G. Yang, “A Data Partitioning and Scrambling Method to Secure Cloud Storage with Healthcare Applications,” in IEEE International Conference on Communications (ICC), 2015, pp. 478–482.

[31] S. Bao, M. Chen, and G. Yang, “A Method of Signal Scrambling to Secure Data Storage for Healthcare Applications,” IEEE J. Biomed. Heal. Informatics, vol. 21, no. 6, pp. 2168–2194, 2017, doi: 10.1109/JBHI.2017.2679979.

[32] A. Y. Alamari, A. Fanfakh, and E. Hadi, “Parallel Message Authentication Algorithm Implemented Over Multicore CPU,” Int. J. Intell. Eng. Syst., vol. 16, no. 4, pp. 642–654, 2023, doi: 10.22266/ijies2023.0831.52.

[33] M. R. Al-Maamari, R. Ramteke, A. M. Al-Hejri, and S. S. Alshamrani, “Integrating CNN and transformer architectures for superior Arabic printed and handwriting characters classification,” Sci. Rep., vol. 15, no. 1, p. 29936, Aug. 2025, doi: 10.1038/s41598-025-12045-z.

[34] Z. Mishra, P. K. Nath, and B. Acharya, “High Throughput Unified Architecture of LEA Algorithm for Image Encryption,” Microprocess. Microsystems J., vol. 78, no. 103214, pp. 2–16, 2020, doi: 10.1016/j.micpro.2020.103214.

[35] D. Lee, D. Kim, D. Kwon, and H. Kim, “Efficient Hardware Implementation of the Lightweight Block Encryption Algorithm LEA,” Sensors, vol. 14, no. 1, pp. 975–994, 2014, doi: 10.3390/s140100975.

[36] M. Hussam, G. H. Abdul-majeed, and H. K. Hooomod, “New Lightweight Hybrid Encryption Algorithm for Cloud Computing (LMGHA-128bit) by using new 5-D hyperchaos system,” Turkish J. Comput. Math. Educ., vol. 12, no. 10, pp. 2531–2540, 2021.

[37] A. Soltani and S. Sharifian, “An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA,” Microprocess. Microsyst., vol. 39, no. 7, pp. 480–493, 2015, doi: 10.1016/j.micpro.2015.07.005.

[38] J. Lu, Y. Liu, T. Ashur, B. Sun, and C. Li, “Improved rotational-XOR cryptanalysis of Simon-like block ciphers.pdf,” IET Inf. Secur., vol. 16, no. 4, pp. 282–300, 2022, doi: 10.1049/ise2.12061.

[39] R. Shahzadi, S. M. Anwar, F. Qamar, M. Ali, and J. J. P. C. Rodrigues, “Chaos Based Enhanced RC5 Algorithm for Security and Integrity of Clinical Images in Remote Health Monitoring,” IEEE Access, vol. 7, pp. 52858–52870, 2019, doi: 10.1109/ACCESS.2019.2909554.

[40] O. S. Faragallah et al., “Improved RC6 Block Cipher Based on Data Dependent Rotations,” Comput. Mater. Contin., vol. 70, no. 1, pp. 1921–1934, 2022, doi: 10.32604/cmc.2022.019798.

[41] J. Aguilar, S. Sierra, and E. Jacinto, “Implementation of ‘ HIGHT ’ Encryption Algorithm on Microcontroller,” in CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2015, pp. 937–942.

[42] A. A. Zakaria, A. H. Azni, F. Ridzuan, N. U. R. H. Zakaria, and M. Daud, “Extended RECTANGLE Algorithm Using 3D Bit Rotation to Propose a New Lightweight Block Cipher for IoT,” IEEE Access, vol. 8, pp. 198646–198658, 2020, doi: 10.1109/ACCESS.2020.3035375.

[43] Z. Yang, Y. Li, B. Wang, S. Ding, and P. Jiang, “A Lightweight Sea Surface Object Detection Network for Unmanned Surface Vehicles,” J. Mar. Sci. Eng. Artic., vol. 10, no. 965, pp. 1–15, 2022, doi: https:// doi.org/10.3390/jmse10070965.

[44] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, “The LED Block Cipher ∗,” in In International workshop on cryptographic hardware and embedded systems, Springer Berlin Heidelberg, 2011, pp. 326–341. doi: https://doi.org/10.1007/978-3-642-23951-9_22.

[45] N. K. Valmik, “BLOWFISH ALGORITHM,” IJESMR, Int. J. Eng. Sci. Manag. Res., vol. 2, no. 10, pp. 45–52, 2015.