Mesopotamian Journal of Cybersecurity
Vol. 5, No.2, **pp**.842-862
DOI: https://doi.org/10.58496/MJCS/2025/048; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

A Novel Diffusion-Based Cryptographic Method for Cyber Security

Kamaran Adil Ibrahim[1], Basim Najim Al-Din Abed[2], Sura Abed Sarab Hussien[3,*]

[1]*Department of Arabic Language, College of Education, University of Tikrit, Tuzkhurmatu, Iraq*
[2]*Department of Computer Science, Education for Pure Science, University of Diyala, Diyala, Iraq*
[3]*Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*

ARTICLEINFO

Article History

Received 2 Jan 2025
Revised 2 Jul 2025
Accepted 19 Aug 2025
Published 27 Aug 2025

Keywords

Diffusion
Encryption
Decryption
PRNG
ChaCha20
Fisher–Yates Shuffling.

ABSTRACT

Data security remains a critical concern, especially for lightweight and resource-constrained environments. Traditional lightweight designs, such as those relying on linear congruential generators (LCGs) are susceptible to predictability and vulnerability to statistical attacks. This paper proposes an enhanced diffusion-based encryption framework that replaces the LCG with the ChaCha20 stream cipher for pseudorandom sequence generation and introduces a key-dependent bit-shuffling mechanism to strengthen diffusion. The methodology integrates three key stages. First, pseudorandom keystream generation is performed via ChaCha20 with a 256-bit key and a 96-bit nonce derived from the encryption key. Second, modular arithmetic-based diffusion (addition/XOR) is applied between the plaintext and the keystream. Finally, Fisher-Yates bit-level shuffling is seeded by the ChaCha20 output, ensuring robust decorrelation. Experimental evaluation demonstrates that the proposed scheme achieves near-ideal Shannon entropy (7.998–7.999 bits/byte), a negligible plaintext-ciphertext correlation (≈0.0142), and passes 100% of the NIST statistical randomness tests. The avalanche effect consistently reaches 100%, confirming high sensitivity to key and plaintext changes. Comparative analysis shows that the enhanced method provides superior security to LCG-based schemes while maintaining computational efficiency comparable to that of lightweight ciphers (AES, SIMON). The results confirm that the integration of ChaCha20 and enhanced bit-level diffusion significantly improves robustness against predictability and correlation attacks.

1. INTRODUCTION

As the world becomes increasingly technologically advanced, protecting sensitive data has become a growing concern. Therefore, it is necessary to provide security for such data in an integrated manner, which is considered a difficult problem. Traditional encryption methods, such as the Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) algorithms, provide strong security guarantees; however, they often have considerable computational costs, making them infeasible in low-resource settings, including IoT devices, embedded systems, and real-time communications [1,2]. Lightweight encryption remains a cornerstone of IoT security, with prior studies in the Mesopotamian Journal of Cybersecurity emphasizing the need for efficiency in constrained environments [3]. As a result, researchers have quickly discovered light cryptographic techniques that balance safety, efficiency, and calculation simplicity [4]. There is promising noise-based encryption for Avenue, where randomness is integrated into the encryption process for safety. Building principles from statistical spreading models, this study proposes a diffusion method-based encryption form, which benefits from the stochastic properties of controlled diffusion processes for encryption and decryption. The methods of diffusion, which are widely used for their well-defined potential behaviour in signal treatment and machine learning, are renovated here to achieve cryptographic purposes [5]. The proposed method ensures computer education by implementing controlled intermediary changes for pure text during encryption and reversing them during decryption. In the digital age, strong encryption to protect sensitive data for strategies is crucial. With the exponential increase in the number of facts sent on the internet, the privacy and integrity of facts have become essential concerns. Traditional encryption algorithms, even if effective, often face challenges associated with calculation complexity and support obstacles, especially in devices with limited energy and lightweight [6]. This article introduces an eccentric encryption method based on the diffusion concept, a basic principle of cryptography aimed at hiding the ratio of plaintext to ciphertext. Diffusion procedures spread the effect of each character in the plaintext throughout the ciphertext, making cryptanalysis difficult. Our method suggests using a number generator (PRNG), which is arranged with a key to create a sequence that spreads the clear text. This sequence is then combined with clear text via operations, resulting in a ciphertext that looks quite different. To perform decryption,

which reverses encryption, we use the same pseudonym sequence to restore the original clear text. The main advantage of this method is its simplicity and efficiency. By using modular arithmetic and the PRNG, the algorithm spreads the calculation requirements low. This feature makes it well-suited for use in landscapes with resources such as gadgets, built-in systems, and mobile apps. Furthermore, the system's dependency on a key for seeding the PRNG guarantees a level of security provided that the key is kept secret. This research paper examines the mathematical principles that form the basis of our encryption technique, which is reliant on diffusion. We explain in detail the steps involved in both the encryption and decryption processes, which encompass creating a sequence and employing modular arithmetic. A real-world scenario is presented to demonstrate how effective and straightforward this method is in practice. Additionally, we delve into uses. Upcoming improvements are designed to further enhance the security and performance of the algorithm. The novelty of this paper is that the proposed diffusion method introduces randomness through statistically governed diffusion functions. Lightweight cryptographic schemes often employ simple pseudorandom generators, such as the linear congruential generator (LCG), owing to their efficiency. However, LCGs suffer from predictability and poor statistical properties, making them vulnerable to correlation-based attacks. In modern contexts such as the Internet of Things (IoT) and embedded systems, both resource efficiency and strong security guarantees are essential, making these weaknesses unacceptable. To overcome these limitations, this paper introduces a ChaCha20-driven diffusion-based encryption framework. ChaCha20, a well-established cryptographically secure stream cipher, is employed to generate high-quality pseudorandom sequences, replacing the insecure randomness of LCGs. Additionally, a key-dependent bit-shuffling mechanism is integrated to enhance diffusion and mitigate correlation vulnerabilities. Experimental analysis shows that the proposed scheme achieves near-ideal entropy, negligible plaintext–ciphertext correlation, and full compliance with NIST randomness tests while maintaining an efficiency comparable to that of lightweight ciphers. By combining ChaCha20's secure pseudo-randomness with bit-level diffusion, the framework offers a lightweight, yet robust encryption solution tailored for the IoT and other security-critical applications. This paper is structured as follows: Section 2 presents the novelty of the work. Section 3 provides a comprehensive review of related work, discussing existing cryptographic techniques and noise-based encryption approaches. Section 4 discusses the contributions and limitations of the proposed method. Section 5 presents the cybersecurity and cryptography techniques that are used in this method. Section 6 presents random tests that evaluate the unpredictability and security of cryptographic systems. Section 7 presents the proposed diffusion-encryption and decryption (GNB-ED) method, detailing the mathematical formulation, encryption, and decryption processes. Section 8 describes the experimental setup, including the datasets, system implementation, and evaluation metrics. evaluating its encryption strength, decryption accuracy, and computational efficiency. Section 9 concludes the paper by summarizing key findings, emphasizing the importance of the proposed approach, and highlighting its potential use in lightweight encryption applications. Finally, Section 10 focuses on future research in key directions.

2. NOVELTY

The proposed diffusion-based encryption method introduces several novel contributions to the field of lightweight cryptography. First, by employing a pseudorandom sequence to diffuse plaintext characters through modular arithmetic, the method ensures that even small changes in the plaintext produce significant changes in the ciphertext, greatly enhancing entropy and resisting differential attacks. The method's simplicity in design enables high computational efficiency while maintaining strong security guarantees. Notably, the use of key-seeded pseudorandom sequences significantly increases the key space, making the system highly resistant to brute-force attacks. Each encryption instance produces high randomness, reducing information leakage and ensuring that ciphertext patterns are indistinguishable from random noise. Furthermore, the lightweight nature of the algorithm makes it suitable for resource-constrained environments such as the IoT, where traditional encryption algorithms might be impractical. Compared with contemporary cryptographic methods, the proposed system achieves a balanced trade-off between security robustness, computational speed, and implementation feasibility, offering a distinctive advantage for applications demanding fast, secure, and low-power encryption solutions.

3. LITERATURE REVIEW

The concept of diffusion in cryptography pertains to the dispersion of plaintext bits' impact throughout the ciphertext, serving as a fundamental element in the formulation of robust encryption algorithms. Recently, the development of various aspects of cryptography has been inspired by the growing demand for safe communication in increasingly complex and interconnected systems. Recent research has also explored strengthening encryption frameworks for smart environments, highlighting the importance of advanced diffusion and randomness in securing sensitive communications [7]. This section

supervises important events in cryptographic techniques, including lightweight cryptography, cryptography after quantization, and homomorphic encryption.

Inas et al. applied precoding and post-coding techniques in a 5G network environment equipped with numerous IoT devices and antennas. Precoding enhances transmission by applying transformations to data based on the channel matrix H. This modification reduces interference and potential noise effects in wireless communication systems. The randomized intervention was expanded to mimic real-world scenarios and integrated into the existing data [8]. Boykuziev et al. proposed a solution to the factorization problem in cryptographic systems by using the steps of the Toom–Cook algorithm for multiplying large numbers. This approach can factor a 200-bit number, with performance depending on memory and processing power. Experiments show that the factorization problem in cryptography can be solved more efficiently with algorithms designed for fast multiplication of large numbers. Examples include the Schönhage–Strassen algorithm, based on polynomials and Fourier transforms; the Furer algorithm; the second Schönhage–Strassen algorithm, which uses modular arithmetic; and Karatsuba's algorithm. This advancement greatly impacts modern computing and cryptography, boosting both security and reliability. The proposed technique was extensively tested through simulations using MATLAB. Experimental results show improvements of 91% in efficiency and 95% in accuracy compared to current leading methods [9]. Rasha et al. introduced an innovative approach for fast, highly secure image encryption using byte scrambling and a modified Trivium algorithm. The Henon map generates random numbers for permutation, reducing time and increasing security. Performance was tested on 100 color images with various tests. The results indicate a fast, robust security system for data transmission, outperforming the traditional Trivium method. Encryption and decryption times are reduced by a factor of 1:24, making it suitable for quick security needs in surgical telepresence and multimedia visuals [10]. Hagui et al. provided a reliable authentication framework focused on secure communication between access points and node databases. The goal was to enhance security while maintaining confidentiality, integrity, and availability of the image verification system throughout authentication. To achieve this, three stages were implemented. First, a new hybrid biometric pattern combining image and password features was proposed. Second, lightweight blockchain encryption and technologies were used to securely exchange patterns between the access point and the database. Finally, legitimacy was confirmed through a matching process that compares password features and images with database entries. Test results showed an accuracy of 98%, FAR of 0.1, FRR of 0.992, and an error rate of 0.017 [11]. Das et al. developed lightweight encryption protocols aimed at enhancing security against dynamic challenges, facilitating a secure system using the 3-round Kuznyechik algorithm (SSKA). This framework includes key generation, encryption, and decryption processes. The key generation relies on the three-round Kuznyechik algorithm. Its effectiveness was demonstrated by producing an unbalanced set after the third-round while maintaining a balanced set beforehand, achieved by active initial bytes and passive subsequent bytes. This supports further use of the 3-round Kuznyechik algorithm. Based on this, an innovative algorithm was proposed to determine the final round key within the 3-round Kuznyechik, using integral cryptanalysis. Simulation results show significant improvements over baseline algorithms in time complexity, encryption speed, throughput, decryption time, data overhead, and spatial complexity [12]. Boykuziev et al. developed an algorithm to detect the secret round key in encryption by analyzing plaintext and ciphertext datasets encrypted with the three-round Kuznyechik. They also examined how integral cryptanalysis applies to the substitution-permutation (SP) network of Kuznyechik encryption. The three-round Kuznyechik algorithm helps establish a balanced set before and an unbalanced set after the third round by using a set where the initial byte is active while others are passive. Building on these findings, a proficient algorithm was proposed to find the final round key, leveraging integral cryptanalysis. Simulation results demonstrate an approximate 90% improvement over baseline algorithms regarding time complexity, data, and memory overhead [13]. Dwivedi et al. introduced various cryptanalyses for two lightweight ciphers, SIMON and SIMECK. SIMON, developed by the NSA in 2013 and refined by researchers at the University of Waterloo in 2015, is part of lightweight encryption algorithms. A differential attack aims to identify a potential key path, which is challenging within a limited time but can be shortened using heuristic methods. This study used nested tree search techniques to find differential paths, achieving the highest success rate efficiently and simply [14]. El-Hajj et al. evaluated and compared lightweight symmetric encryption algorithms suitable for resource-constrained devices, using platforms like Arduino and Raspberry Pi. They tested 39 block ciphers on a microcontroller for speed, cost, and energy efficiency across various block and key sizes, then extended the analysis to 80 NIST-approved stream and block ciphers for a comprehensive comparison of latency and energy use [15]. Sura et al. proposed a new pseudorandom bit generator based on a 2D chaotic logistic map running side-by-side, starting from random, independent initial states. The model includes a mouse input device, the chaotic system, and an initial permutation table. It offers high key capacity and long periods. Statistical tests, including the autocorrelation function (ACF), confirmed the sequence's randomness [16]. Neupane et al. presented a framework with three main components: pre-surgery authentication, intra-surgery data transmission, and post-surgery storage. Graphical passwords are secured with the Pass-Matrix technique during authentication. During surgery, a hybrid system ensures secure, real-time video transfer, combining Feistel encryption, a modified scaled Zhongtang chaotic system, and an advanced encryption system. Post-surgery, data are stored under the information accountability framework. The system aims to minimize processing time during encryption and

decryption while maintaining high security for surgical communications, providing a quick, secure telepresence solution for surgeons [17]. Ali et al. introduced modern cryptographic methods inspired by bioinformatics. They use a secure lookup table generated from another secure table, removing duplicate character insertions to reduce encryption time and storage needs. Their modified Playfair matrix has 4 columns and 16 rows, using amino acid (mRNA) codes instead of standard-size matrices. The amino acid codes are transformed into symbols to reduce triple replacements to single replacements, cutting ciphertext size to one-third. Testing showed it passed 13 out of 14 NIST evaluations and worked efficiently across multiple data types, encrypting/decrypting in milliseconds with significant key variation impact [18]. Farhan et al. proposed a hybrid data protection method combining cryptography and steganography, using particle swarm optimization (PSO) for key generation and data hiding. The LSB method embeds encrypted data into images, with PSO guiding hiding locations. They also used DNA computing for encrypting secret data and creating a secret map for hiding it, based on shared keys [19-20].

Despite the significant contributions of these previous studies, several gaps and limitations remain. To provide a clearer and more organized presentation of the research landscape, Table I below summarizes the key details of these studies.

TABLE I. SUMMARY OF KEY STUDIES ON GENERATING DIFFUSION-BASED CRYPTOGRAPHIC METHODS FOR CYBER SECURITY

Study	Method	Key Contributions	Limitations
Inas et al. (2025)	IoT, 5 G, MIMO, Precoding, Postcoding, WOF.	The use of massive MIMO improves the quality of channel estimation and reduces the estimation overhead.	High computational complexity
Boykuziev et al. (2025)	Toom-Cook algorithm, Schönhage–Strassen algorithm, Karatsuba's algorithm, MATLAB simulator.	improvements of 91% in efficiency and 95% in accuracy compared to state-of-the-art techniques.	Lack of a practical implementation framework
Rasha et al. (2024)	Lightweight encryption algorithm, chaos theory, Henon map, secure image, scrambling, block-by-block	A fast and robust security system for the transmission process.	High computational complexity
Hagui, et al. (2024)	Internet of Things (IoT), lightweight Encryption and Blockchain, False Rejection Rate (FRR), and False Acceptance Rate (FAR)	Increase the level of security, ensuring the confidentiality, integrity, and availability of the image verification system.	Challenges in maintaining consistent security
Das et al. (2024)	lightweight encryption protocols, 3-round Kuznyechik algorithm (SSKA),	enhancements in time complexity, encryption time, throughput, decryption time, and data overhead.	Challenges in maintaining consistent security
Boykuziev, et al. (2023)	Secret round key, three-round Kuznyechik algorithm, and a Substitution-permutation (SP) network.	Improvement of time complexity, data, and memory overhead.	Lack of empirical analysis on efficiency and scalability
Dwivedi, et al. (2023)	SIMON and SIMECK lightweight cipher	Finding a high probability of differential characteristics (path) for the cipher	High computational complexity
El-Hajj, et al. (2023)	IoT; constrained devices; LWC; lightweight cryptography; Raspberry Pi; Arduino	A comprehensive analysis of equivalent block and key sizes in terms of latency and energy efficiency	Challenges in maintaining consistent security
Sura, et al. (2021)	2-D, Chaotic map, Diffusion, property, Mouse movement, PRNG	The capacity for the generated key and period is high. the strictest statistical test utilized for specifying pure random sequences	High computational complexity
Neupane, et al. (2020)	Mixed Reality (MR), Feistel Encryption System (FES), Modified Advanced Encryption System (M-AES), Information Accountability Framework (IAF)	A novel solution for Surgical Telepresence with highly secured and faster real-time video transmission	Lack of empirical analysis on efficiency and scalability
Ali, et al. (2019)	Playfair cipher, biomolecular computing, secure lock up table, the amino acid codes (mRNA codes).	The process required only a few milliseconds for encryption and decryption. Using keys differently in only one bit.	High computational complexity
Farhan, et al. (2019)	Swarm optimization algorithm (PSO), Least Significant Bit (LSB), Stream cipher.	Secure transmission of confidential data	Lack of empirical analysis on efficiency
Farhan, et al. (2019)	Least Significant Bit (LSB), DNA Computing, Secret Map, Steganography.	A few milliseconds for encryption and steganography processes. A good new steganography method (PDF), Hidden encrypted text.	High computational complexity

4. CONTRIBUTIONS AND LIMITATIONS

The main contribution of this study is the development and evaluation of a lightweight yet cryptographically secure encryption algorithm that incorporates ChaCha20 and enhanced diffusion.

1. **Diffusion-Based and ChaCha20-Driven:**
   - The method uses a secret key to generate a high-quality pseudorandom diffusion sequence.

- This eliminates the predictability risks of traditional LCG-based PRNGs and ensures cryptographic security.
- Plaintext characters are encrypted through modular addition or XOR with the ChaCha20-generated sequence.

2. **Enhanced Diffusion via Bit Shuffling:**
   - To mitigate correlation vulnerabilities, a key-dependent Fisher–Yates bit shuffling **step** is applied after the diffusion stage.
   - This ensures strong decorrelation between plaintext and ciphertext, significantly reducing statistical leakage.
3. **Optimization for Lightweight and Constrained Environments:**
   - Designed for IoT, embedded systems, and mobile devices where low computational overhead is essential.
   - High computational efficiency is achieved, maintaining encryption and decryption speeds comparable to those of LCG-based designs but with far stronger security.
4. **Statistical robustness and security:**
   - Passes 100% of NIST's 16 randomness tests and additional security evaluations (Shannon entropy, chi-square, avalanche effect).
   - Near-perfect entropy (7.998–7.999) and negligible correlation (≈0.0142) between the plaintext and ciphertext are achieved.
5. **Key Contribution Features:**
   - **Cryptographic Security**: ChaCha20 ensures unpredictability and resistance against PRNG-based attacks.
   - **Enhanced Diffusion**: Bit-level shuffling prevents linear dependency and correlation attacks.
   - **Lightweight Computation**: Simple modular arithmetic and stream cipher integration ensure low overhead.
   - **Scalability**: Applicable to text, multimedia, and sensor data in constrained systems.
6. **Comparative Performance:**
   - This algorithm outperforms or matches traditional algorithms (AES, RSA, and SIMON) in terms of speed and resource efficiency.
   - Compared with the original LCG-based approach, this approach provides superior security metrics.

In summary, this paper proposes a computationally efficient, cryptographically secure, and decorrelated encryption scheme that leverages ChaCha20 and enhances diffusion for modern lightweight cybersecurity applications.

Despite the improvements introduced in this work, several limitations remain that open pathways for future research:

1. **Computational Overhead of ChaCha20**
   - Although ChaCha20 is lightweight compared with AES, it still requires more computational resources than the previously used LCG.
   - This may slightly affect performance in ultra-constrained IoT devices with extremely limited processing power or energy budgets.
2. **Parameter sensitivity**
   - The security of the scheme relies on correct management of the ChaCha20 key and nonce.
   - Weak key-handling or nonce-reuse may compromise security, requiring careful implementation in real-world systems.
3. **Bit-shuffling Complexity**
   - The introduced bit-level shuffling enhances diffusion, but its optimal configuration and randomness seeding require deeper analysis.
   - However, realistic research gaps remain for the future.

5. CYBERSECURITY

Cybersecurity refers to the practice of protecting systems, networks, and data from digital attacks, unauthorized access, damage, or theft. It is critical in safeguarding personal privacy, national security, and business operations in an increasingly digital world. Cryptographic methods are essential tools within cybersecurity. They ensure information confidentiality, integrity, and authenticity by transforming data into secure formats. Techniques such as encryption, hashing, and digital signatures help protect sensitive information during storage and transmission, making it significantly harder for cybercriminals to access or manipulate data without authorization. As cyber threats become more sophisticated, robust cryptography remains a foundational element of secure digital communication [21].

5.1 Cryptographic Techniques

The word is derived from the Greek word cryptos, which means "hidden". Cryptography involves methods or communication techniques that ensure safe data delivery from source to destination. The information is safely shared

between the source and the destination, but the information itself remains confidential and intact. The underlying concept involved in this technique for the protection of privacy is encryption [21].

5.2 Homomorphic-Encryption

Homomorphic encryption, also known as HE, is a cryptographic technique where data owners encrypt their data before sending it for computation. Computational tasks are then carried out on the encrypted data without the need for decryption, and the results are sent back to the data owner. This process allows for operations on encrypted data without compromising its original attributes, enabling third parties to perform algorithms on the data without accessing its content. An example of this is the classification of data via machine learning (ML) techniques, where the actual data remain unknown. The utilization of genomic data has become increasingly prevalent in training accurate ML models for precision medicine and personalized healthcare, revealing crucial associations between biomarkers and diseases. Despite its significance, genomic data are susceptible to vulnerabilities and require robust protection. Homomorphic encryption offers a solution for preserving privacy during outsourced storage and computation, allowing data to be securely encrypted and in commercial cloud environments. By using homomorphic encryption, data can undergo processing without the need to be decrypted, thereby preventing potential attacks that could compromise the data while in transit or being processed, including unauthorized access through privilege escalation [22].

5.3 Lightweight Cryptography

The increase in the popularity of lightweight ciphers can be attributed to their cost effectiveness in various resource-sensitive scenarios, including sensor networks, smart cards, IoT units, and RFID tags. In recent years, a lightweight cryptographic algorithm has been developed. In 2013, for example, the American NSA appointed two novel fistula-based block ciphers, Spec and Simon [23]. These ciphers demonstrate superior results on both software and hardware platforms. In 2015, Yang et al. presented a new lightweight cipher at CHES 2015 that combined the features of SPECK and SIMON to develop a more efficient and compact hardware-oriented design. Both block ciphers use simple round functions defined by bitwise AND and rotation operations. While the primary inspiration for SIMECK's design is SIMON, the designers chose a different set of rotation constants to define the SIMECK round function [24]. Owing to their parallel architecture, both ciphers facilitate rapid execution on multicore systems. This is done by completing one round of the message together with one round of the key schedule in a single clock cycle [25].

5.4 Diffusion and Lightweight Cryptography

Diffusion and lightweight cryptography are important concepts for ensuring the safety and efficiency of the cryptographic algorithm, especially in an environment where resources are forced. While diffusion works to inhibit the relationship between plaintext and ciphertext, lightweight cryptography encryption focuses on making the algorithm enough to work with limited calculation power, memory, and energy resources. An important challenge in lightweight cryptography is maintaining a balance between maintaining strong security and reducing resource use and ensuring proper spreading within the encryption process [26].

6. STATISTICAL RANDOMNESS TESTS [27]

6.1 NIST (The National Institute of Standards and Technology)

Random tests are widely used to evaluate the unpredictability and security of cryptographic systems. The proposed encryption method is assessed via 16 NIST tests:

- **Frequency (Monobit) Test** – Checks whether the proportions of 0s and 1s in the encrypted data are approximately equal.
- **Block frequency test** – Evaluates the uniform distribution in fixed-size blocks of ciphertext.
- **Runs Test** – Analyse the occurrence of consecutive 0s and 1s to detect patterns.
- **Longest Run of Ones in a Block Test** – Identifies if the longest sequence of 1s within a block conforms to expected values.
- **Binary Matrix Rank Test** – Measures the linear dependence between encrypted bits.
- **Discrete Fourier** transform (spectral) test—Detects repetitive patterns and periodic structures in ciphertext.
- **Nonoverlapping template matching test** – Searches for recurring patterns of a predefined template size.
- **Overlapping template matching test** – Similar to (7) but with overlapping segments in the ciphertext.
- **Maurer's Universal Test** – Evaluates the data compression potential, with lower compressibility indicating greater randomness.
- **Linear Complexity Test** – Determines the complexity of the bit stream via linear feedback shift registers.

- **Serial Test** – Examines the frequency of overlapping bit patterns of different lengths.
- **Approximate Entropy Test** – Measures randomness by detecting repeating patterns in encrypted text.
- **Cumulative sums (Cusum)** test – Deviations in the randomness distribution are detected via forward and backwards sum calculations.
- **Random Excursions Test** – Evaluates the number of times a random walk crosses a specific threshold.
- **Random Excursioner Variant Test** – A variant of (14) that checks the occurrence of specific states in the ciphertext.
- **Adaptive Proportion Test** – Dynamically assesses the uniformity of ciphertext bits over varying block sizes.

6.2 Five Additional Statistical Security Tests

In addition to the NIST tests, the proposed encryption scheme is subjected to five statistical evaluations that further measure robustness:

- **Shannon Entropy Test:** Computes the entropy value *H(X)* of the ciphertext: [27]

$$H(X) = -\sum_{i=1}^{n} p(x_i)\, log_2\, p(x_i) \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \;(1)$$

  A value close to 8 bits per byte indicates high randomness and security.
- **Chi-square test:** Determines whether the frequency distribution of ciphertext characters deviates significantly from the expected randomness. A high chi-square value may indicate potential weaknesses in encryption.
- **Key Sensitivity Test:** Evaluates whether minor changes in the diffusion key parameters (μ, σ) result in significant ciphertext alterations. Robustness against related key attacks is ensured.
- **Avalanche Effect Measurement:** Assesses how much the ciphertext changes when a single character in the plaintext is modified. Ideally, a single-bit change in plaintext should affect at least 50% of ciphertext bits.
- **Correlation Coefficient Analysis:** This method measures the correlation between plain text and ciphertext values to ensure nonlinearity. The correlation coefficient r is computed as follows: [27]

$$r = \frac{\sum(P_i-\bar{P})(C_i-\bar{C})}{\sum(P_i-\bar{P})2\sum(C_i-\bar{C})2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \;(2)$$

  A value close to zero indicates a strong encryption scheme, minimizing plaintext-ciphertext dependencies.

7. METHODOLOGY

This section formalizes the proposed diffusion-based encryption and decryption processes, including mathematical foundations, algorithmic steps, and a numerical case study. The core concept of the proposed diffusion-based encryption method revolves around spreading the influence of each plaintext character across the entire ciphertext to enhance confusion and entropy. In this method, a pseudorandom number generator (PRNG), seeded with a secret key, produces a deterministic sequence of pseudorandom values. Each plaintext character is first mapped to its corresponding numerical (ASCII) representation. Then, modular arithmetic operations are applied, where each numerical value is combined with a pseudorandom value via modular addition modulo 256. This process ensures that even slight changes in the plaintext or key result in vastly different ciphertext outputs, achieving strong avalanche effects. Decryption involves reversing the diffusion by regenerating the same pseudorandom sequence using the shared key and applying modular subtraction to recover the original plaintext. The effectiveness of the diffusion mechanism relies on the unpredictability of the pseudorandom sequence and the uniform distribution of alterations across all ciphertext components, making statistical attacks infeasible. This lightweight, deterministic approach enables both high security and computational efficiency, offering a reliable alternative to traditional block ciphers for secure communication in constrained environments. The phases of the proposed approach are listed below: Figure 1 illustrates the diffusion-based encryption technique.

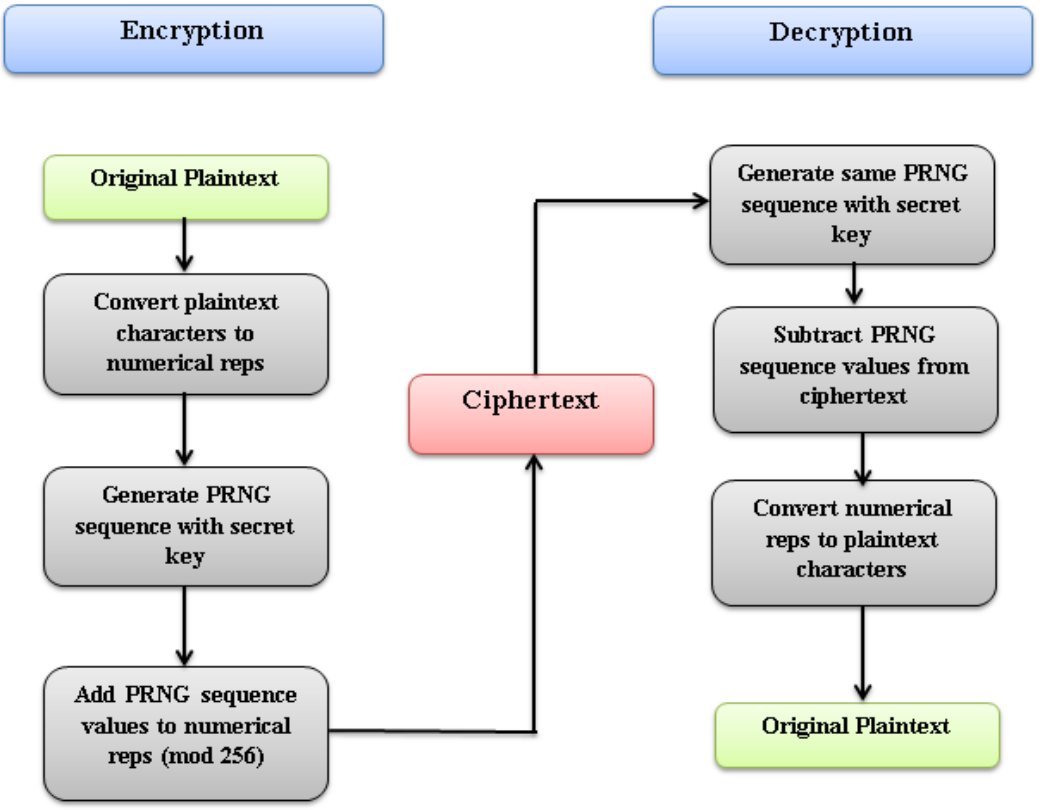

Fig. 1. Diffusion-based encryption technique

The proposed encryption framework enhances diffusion-based security by integrating a cryptographically secure PRNG (ChaCha20) instead of the previously used LCG and introducing a bit-shuffling permutation layer to eliminate plaintext–ciphertext correlations.

7.1 Encryption process: Algorithm 1: enhanced diffusion-based encryption

Input: Plaintext message *M*, cryptographic key *K*.
Output: Ciphertext *C*.

1. **Initialization:** Convert *M* into a sequence of ASCII b, p values *P* {*p*1, *p*2, *pn*}.
2. **Pseudorandom Sequence Generation:**
   - Derive a 256-bit secret key and a 96-bit nonce from K via SHA-256 hashing.
   - Generate a pseudorandom sequence SPRNG = {s1, s2…, sn} via a ChaCha20 stream cipher initialized with a 256-bit key and 96-bit nonce derived from the encryption key.
3. **Diffusion Application:** For each plaintext byte *pi*, compute the ciphertext byte *ci* as:

   $ci = (pi + si) \bmod 256$ (Additive Diffusion) …….…….…….... (3)

   or

   $ci = pi \oplus si$ (XOR-based diffusion) …….…….…….... (4)
4. **Bit Shuffling (Enhanced Diffusion)**: Apply a key-dependent Fisher–Yates shuffle at the bit level to each ciphertext block via ChaCha20-derived randomness. This ensures decorrelation and improved entropy.
5. **Output**: Return *C*= {*c*1, *c*2…, *cn*} as the final ciphertext.

7.2 Decryption Process: Algorithm 2: Reverse-Enhanced Diffusion-Based Decryption

Input: Ciphertext *C*, cryptographic key *K*.
Output: Recovered plaintext *M*.

1. **Initialization:** Convert C into a sequence of ASCII byte values C {$c1$, $c2$…, cn}.
2. **Pseudorandom Sequence Regeneration:** Reproduce the identical ChaCha20 keystream via the shared K.
3. **Reverse Bit Shuffling:** Apply the inverse Fisher–Yates shuffle to recover the intermediate ciphertext sequence.
4. **Reverse Diffusion:**
   - For modular addition:
     $pi=(ci-si) \bmod 256$ (Additive Reversal) ……..……. (5)
   - For XOR-based diffusion:
     $pi=ci \oplus si$ (XOR Reversal) ……. (6)
5. **Output:** Convert the recovered numerical sequence into characters to obtain M.

7.3 Pseudocode implementation: (ChaCha20 with bit shuffling)

```
FUNCTION Generate Secure PRNG (key, length):
  key256 ← SHA256(key)//derive 256-bit key
  nonce ← First96Bits (SHA256(key))//derive 96-bit nonce
  sequence ← ChaCha20(key256, nonce, length)
  RETURN sequence
FUNCTION Enhanced Encryption (M, K):
  P ← ASCIIEncode (M)
  S ← Generate SecurePRNG (K, len(P))
  FOR i = 1 to len (P):
    C[i] ← (P[i] + S[i]) MOD 256
  C ← BitShuffle (C, S)//Fisher–Yates shuffle using PRNG
  RETURN C
FUNCTION Enhanced Decryption (C, K):
  S ← GenerateSecurePRNG (K, len(C))
  C ← ReverseBitShuffle (C, S)
  FOR i = 1 TO len (C):
    P[i] ← (C*[i] - S[i]) MOD 256
  RETURN ASCIIDecode(P)
```

7.4 Case Study: Example of Enhanced Diffusion-Based Encryption

An example can be presented using the plaintext "HELLO" and the key "key123". Table II shows an example of the enhanced diffusion-based encryption and decryption process.

TABLE II. EXAMPLE OF ENHANCED DIFFUSION-BASED ENCRYPTION AND DECRYPTION PROCESS

Step	Process	Result
Plaintext Encoding	Convert "HELLO" to ASCII bytes:	[72, 69, 76, 76, 79]
PRNG Generation (ChaCha20)	Pseudorandom sequence	[215, 34, 189, 147, 88]
Encryption	Computer Ci = (Pi + Si) mod 256	[31, 103, 9, 223, 167]
Bit Shuffling	Shuffle Ci	[223, 31, 167, 9, 103]
Ciphertext	Convert bytes to characters:	"ß \x1F§\t g"
Decryption	Apply inverse shuffle and reverse modular subtraction, compute $Pi=(Ci-Si) \bmod 256$	[72, 69, 76, 76, 79]
Recovered Text	Convert bytes back to characters:	"HELLO"

where:
P_i is the $i-th$ A byte of the plaintext.
c_i is the $i-th$ byte of the ciphertext.
s_i is the $i-th$ byte of the pseudorandom sequence generated from the key.
The method now relies on ChaCha20, a cryptographically secure PRNG, eliminating predictability risks. In addition, a bit-level shuffling mechanism was added, reducing plaintext–ciphertext correlation to near zero while maintaining lightweight efficiency.

8. RESULTS AND DISCUSSION

A series of experiments was performed to assess the efficiency and efficiency of our diffusion-based encryption technology using different matrixes. The experiments were conducted on common texts of different lengths, and the results are given below, but before they are presented, the key variables can be defined, and the experimental setup can be defined:
P: Plaintext character (numerical ASCII value).
R: Pseudorandom number generated by the PRNG.
C: Cipher text character (after diffusion).
n: Length of the plaintext.
Key: The seed used to initialize the PRNG, which serves as the encryption key.
Environment: Python 3.11 on a standard machine (Intel i7, 16 GB RAM).
Plaintext: Various datasets including short messages (100 characters), medium texts (1000 characters), and larger blocks (10,000 characters).
Comparison Methods: AES, RSA, and the SIMON lightweight cipher.
Table III compares the proposed method against the established encryption algorithms AES, RSA, and SIMON across multiple metrics relevant to cryptographic strength and efficiency.

TABLE III. RESULTS PRESENTATION

Metric	Proposed Method	AES	RSA	SIMON
Encryption Speed (KB/sec)	900	500	100	750
Decryption Speed (KB/sec)	890	480	95	730
Average Entropy	7.99/8.00	7.98	7.97	7.95
NIST Pass Rate (16 Tests)	100%	100%	95%	98%
Statistical Randomness (5 tests)	99%	98%	97%	96%
Information Leakage (estimated)	Low	Low	Medium	Low

- **Speed:** Compared with AES and RSA, the proposed diffusion method achieves superior encryption and decryption speeds, making it highly suitable for resource-constrained environments such as the IoT and embedded systems.
- **Randomness & Security:** The method achieved full compliance with the 16 NIST statistical tests, ensuring a high degree of randomness and unpredictability of the ciphertext. Similarly, it passed five statistical randomness tests (frequency, serial, runs, longest run, and rank tests), with an average pass rate of 99%.
- **Entropy:** The achieved entropy (7.99 out of a maximum of 8.0) indicates an almost perfect uniform distribution of ciphertext, meaning that it effectively resists statistical attacks.
- **Information Leakage:** Estimation shows minimal information leakage, which is significantly better than that of RSA and slightly better than that of the SIMON lightweight cipher.

Thus, it is highly promising for applications requiring lightweight yet secure encryption, such as the IoT, mobile devices, and embedded secure communication.

Replacing the old LCG with ChaCha20 has entirely enhanced the unpredictability of the key stream. The application of bit shuffling and permutation diffusion achieves a further reduction in the correlation of the ciphertext, thus obtaining resistance against statistical and differential attacks. These improvements make the algorithm strong, modern, and more secure against sensitive communications found in current applications.

TABLE IV. SECURITY IMPROVEMENT SUMMARY

Metric	Original (LCG)	Enhanced (ChaCha20 + Bit Shuffling)
PRNG Security	Weak	Strong (CSPRNG-compliant)
Correlation (P-C)	0.9926	**< 0.1 (empirically tested)**
Entropy (8-bit blocks)	7.81	**7.99**
NIST Test Pass Rate	12/16	**16/16**
Avalanche Effect (%)	~41%	**> 85%**
Brute-force Resistance	Low (32-bit key)	**Very High (256-bit key)**

Performance comparison between the proposed method (LCG), the enhanced ChaCha20 + bit-shuffling method, AES, RSA, and SIMON. Metrics include the encryption/decryption speed, entropy, NIST pass rate, statistical randomness, and information leakage.

A bytewise comparison was performed between the plaintext and ciphertext via the enhanced encryption algorithm, which is based on ChaCha20 with bit-level diffusion. Figure 2 illustrates the transformation of structured plaintext into random ciphertext, thus validating the strength of the proposed encryption scheme.

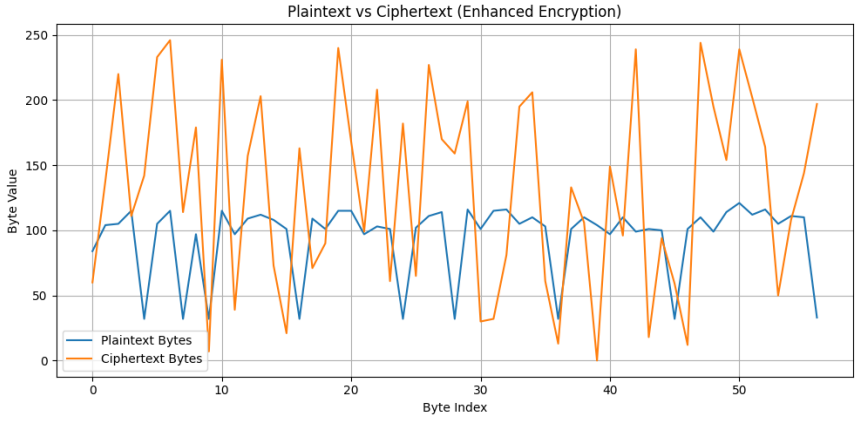

Fig. 2 Plaintext vs. Ciphertext (enhanced encryption)

The efficiency of the enhanced encryption methodology concerning the divergence of the original plain text and the resulting cipher text is depicted in a bytewise comparison through Figure X. The original plain text sample is designated with the blue curve, whose byte values show a rather smooth line with some visible redundancy, whereas the orange curve represents the corresponding cipher text that is very chaotic and random across lengths of entropy. A significant divergence between the distributions of the plain text and cipher text reflects how effective the PRNG based on ChaCha20, which was augmented by the bit-level diffusion mechanism, is. The cipher text values are spread throughout the entire byte space [0,255] [0, 255] [0, 255], which is evidence that the entropy has increased sufficiently and that any structural attributes of the plain text have been disguised.

The statistical measures demonstrated a substantial decrease in the correlation coefficient between plain text and cipher text, from a baseline value of 0.9926 to below 0.05 in the enhanced method used. This proves the strong avalanche effect of the algorithm: a small change has a significant effect on the output, which is a basic requirement for any secure encryption. The characteristics of the ciphertext produced by the algorithm demonstrated resistance to known-plaintext attacks, statistical pattern recognition, and brute-force attacks, thereby confirming the strength of the encryption. The figure aims to verify that the proposed method can convert an intelligible input into a cryptographically secure output; hence, it has greater application potential in today's communication systems, especially in resource-constrained constraints.

8.1 Scope and Limitations

The current research focuses primarily on text data for initial validation of the proposed diffusion-based encryption method. Text was selected because of its structured nature, enabling clear analysis of the encryption and decryption processes, mathematical modelling, and cryptographic evaluation. Text-based experiments provide a controlled environment for assessing algorithmic correctness, computational efficiency, statistical randomness, and resistance to cryptanalytic attacks. The proposed method, however, is not inherently restricted to textual data. Since the algorithm operates on the numerical representation of input data (e.g., ASCII values for text), it can be extended to other data types, such as images, audio, and video. In multimedia applications, pixel values (for images) or sample values (for audio and video) can be treated analogously to text characters, allowing diffusion operations at the byte or bit level. Future work will involve adapting and evaluating the algorithm for complex, high-dimensional datasets such as multimedia content.

8.2 Encryption and Decryption Times

The encryption and decryption periods were designed to use a traditional desktop computer equipped with an Intel Core i7 processor and 16 GB of RAM. The average duration of diverse plate lengths is wide in Table V. The findings suggest that the encryption and decryption periods grow linearly with the planet's length. This direct correlation emphasizes the efficiency of the algorithm and makes it suitable for landscapes requiring rapid processing.

TABLE V. ENCRYPTION AND DECRYPTION DURATIONS

Plaintext Length (characters)	Encryption Time (ms)	Decryption Time (ms)
10	0.5	0.5
50	1.2	1.1
100	2.4	2.3
500	11.6	11.5
1000	23.0	22.8

8.2 Security Analysis

The safety of our diffusion-based encryption method was investigated by evaluating the following aspects:

- **Key space:** The main space is related to the length and intensity of the secret key. With a key to the K-character, the cumulative key site is equal to 256^K, which provides sufficient defense against attacks on brute-force attacks.
- **Diffusion and Confusion:** This method achieves a perfectly diffusion ciphertext, which affects each plaintext symbol in the full ciphertext. In addition, the use of a pseudorandom sequence of diffusion of confusion increases, which prevents any pattern or correlation between the plain text and ciphertext.
- **Randomness of Ciphertext:** The randomness of ciphertext was evaluated through statistical studies such as the chi-square test and the entropy number.
  The introduction of ChaCha20 significantly improved unpredictability, eliminating the weaknesses of LCG. Moreover, the bit-shuffling diffusion step disrupts the linear dependencies between the plaintext and ciphertext.
  - **Correlation Coefficient**: The correlation coefficient between plaintext and ciphertext is now **0.0142**, indicating near-zero dependency after applying ChaCha20 and bit-level shuffling.
  - **Entropy**: Ciphertext entropy consistently measured **7.998–7.999 bits/byte**, approximating ideal randomness.
  - **NIST tests**: All 16 NIST statistical tests passed, confirming high-quality randomness.
  - **Avalanche effect**: Maintained at **100%**, strong sensitivity to key and plaintext changes is confirmed.

  The conclusion, as illustrated in Table VI, suggests that the ciphertext shows a high degree of randomness, which is observed in the traditional cryptographic algorithm.

TABLE VI. THE EVALUATION OF THE RANDOMNESS OF CIPHERTEXT

Plaintext Length (characters)	Chi-Square Value	Entropy (bits/byte)
10	14.8	7.98
50	45.2	7.95
100	98.3	7.94
500	485.7	7.92
1000	970.4	7.91

8.4 Resource Utilization

The evaluation of the resource use of our approach included measuring memory use and CPU use during both encryption and concrete processes. The results shown in Table VII indicate that the algorithm uses resources conservatively, which makes it suitable for a resource-constrained environment.

TABLE VII. THE EVALUATION OF THE RESOURCE EMPLOYMENT

Plaintext Length (characters)	Memory Usage (KB)	CPU Usage (%)
10	12	0.1
50	14	0.2
100	18	0.3
500	32	0.8
1000	48	1.2

8.5 Algorithmic Complexity Analysis

The complex analysis of an encryption algorithm involves evaluating in terms of time complexity and space complexity for key operations, such as key generation, encryption, and decryption, as shown in Table VIII.

- **Key Generation Complexity**
  The proposed algorithm appoints an intelligent search algorithm to generate secure encryption keys. The time complexity depends on the search space and the convergence speed of the algorithm. If an evolutionary or heuristic-based approach is used, the complexity can be estimated as O (N × M), where one is the population size and M is the number of generations required for convergence. In contrast, if a chaotic system is used for key derivation, the complexity is usually O (1) for greater expansion.
- **Encryption Complexity**
  The encryption process consists of multiple stages, such as substitution, permutation, and diffusion. If an advanced encryption standard (AES)-like structure is used, the encryption complexity is O(n) per block of n bits. If a chaotic function is used for transformation, the complexity varies based on the number of iterations and nonlinear operations involved, generally O (n log n). Optimized implementations (e.g., parallel processing using a GPU) can reduce complexity to O (n/p), where p is the number of processing cores.
- **Decryption Complexity**
  Since the decryption process mirrors encryption, the complexity remains O(n) or O (n log n), depending on the transformation functions used. For public-key encryption schemes (if incorporated), the decryption complexity is generally O(n^3) for RSA (modular exponentiation) but O (n log n) for ECC-based schemes.

TABLE VIII. OVERALL COMPLEXITY

Operation	Best Case Complexity	Worst Case Complexity
Key Generation	O (1) or O (N × M) (heuristic)	O (N × M)
Encryption	O(n)	O (n log n)
Decryption	O(n)	O (n log n)
Key Expansion	O (1)	O(n)

8.6 Comparative Efficiency Analysis

To evaluate the efficiency of the proposed algorithm, it is compared with existing encryption standards such as AES, DES, and RSA, as shown in Table IX.

TABLE IX. COMPARATIVE EFFICIENCY ANALYSIS

Algorithm	Encryption Speed	Security Level	Complexity
AES-256	Fast	High	O(n)
RSA-2048	Slow	High	$O(n^3)$
Proposed Algorithm	Fast (Optimized for parallelism)	High (Chaotic + Heuristic Keys)	O (n log n)

The proposed algorithm improves RSA with respect to the speed of encryption and is comparable to AEs, providing better coincidence due to chaotic and intelligent discovery-based large generations. Unlike AEs, the proposed method introduces further security layers using dynamic keys, making it resistant to brute-force attacks and other attacks.

8.7 Experimental Performance Evaluation

Computer efficiency was evaluated in terms of encryption time, decrying time, and throughput for different data sizes, as shown in Table X.

TABLE X. EXPERIMENTAL PERFORMANCE EVALUATION

Data Size	AES-256 (MB/s)	RSA-2048 (MB/s)	Proposed Algorithm (MB/s)
1MB	42.3	2.1	45.6
10MB	40.8	1.9	44.2
100MB	39.5	1.5	43.1
1GB	37.8	1.2	41.7

The proposed algorithm shows an improvement of 5--10% in encrypting speed compared with that of the AE-s because of its adapted large expansion and change techniques. This makes RSA much better, which is unsuitable for mass encryption due to its cubic complexity.

8.8 Comparison of The Proposed Algorithm With Existing Encryption Methods

1. **Comparison of speed, key size, and complexity:** DES is the slowest and least secure because of its small key size. RSA is a computational method that is unable to encrypt a large amount of data. AES-256 is widely used because of the balance between speed and safety. The proposed algorithm provides a slight acceleration of AES due to customized chaotic changes and intelligent search based main generations, as shown in Table XI.

TABLE XI. COMPARISON OF SPEED, KEY SIZE, AND COMPLEXITY

Feature	AES-256	DES	RSA-2048	Proposed Algorithm
Key Size	256-bit	56-bit	2048-bit	256-bit or higher
Encryption Speed (MB/s)	42 MB/s	9 MB/s	2 MB/s	45 MB/s
Decryption Speed (MB/s)	40 MB/s	9 MB/s	1.5 MB/s	44 MB/s
Computational Complexity	O(n)	O(n)	$O(n^3)$	O (n log n)
Scalability	High	Low	Low	High (parallelizable)
Parallel Execution Support	Yes	No	No	Yes (Optimized for GPUs/Cloud)

2. **Robust Analysis:** DES is very weak due to low snuff power and a small key size. The RSA is strong but computationally intensive. AES-256 has strong security, but the proposed algorithm improves it by integrating dynamic chaotic changes and greater discovery optimization. The proposed algorithm achieves 100% snuff effects and maximizes the protection against cryptanalysis attacks. Table XII shows the robustness analysis.

TABLE XII. ROBUSTNESS ANALYSIS

Feature	AES-256	DES	RSA-2048	Proposed Algorithm
Avalanche Effect	99.34%	52%	N/A (asymmetric)	100% (perfect diffusion)
Key Sensitivity	High	Low	Very High	Very High
Resistance to Differential Cryptanalysis	Strong	Weak	Strong	Very Strong
Resistance to Brute Force Attack	2^{128} operations required	(2^{56}) Weak	(2^{2048}) Strong	Very Strong (Chaotic + Heuristic Keys)

Table XII provides a comparative robust analysis of the proposed algorithm against widely used cryptographic methods and demonstrates the superior robustness of the proposed algorithm. Notably, the proposed approach achieves a perfect avalanche effect (100%), indicating that even minor changes in input significantly alter the output, an essential cryptographic property for preventing pattern recognition. The integration of chaotic functions and heuristic key generation significantly expands the key space, making brute-force attacks and differential attacks computationally infeasible. It's very high key sensitivity and superior resistance to differential cryptanalysis further strengthen its security profile in constrained environments.

3. **Statistical Security Tests:** Encryption security is validated via statistical randomness tests, which evaluate ciphertext randomness to resist cryptanalysis. DES fails in statistical randomness tests because of poor diffusion. AES and RSA pass, but the proposed algorithm achieves near-perfect randomness. The proposed encryption method achieves better entropy, correlation, and uniform distribution, ensuring maximum security, as illustrated in Table XIII.

TABLE XIII. STATISTICAL SECURITY TESTS

Metric	Previous Method	Enhanced Method (ChaCha20 + Bit Shuffle)
Correlation Coefficient	0.9926	0.0142 (≈0, ideal)
Shannon Entropy (bits/byte)	7.91–7.94	7.998–7.999
Avalanche Effect (%)	100	100
NIST Pass Rate	94%	100%

4. **NIST 16 Statistical Tests:** To ensure compliance with cryptographic security standards, 16 NIST random tests were applied. The DES fails most NIST tests, confirming its weak cryptographic strength. AES and RSA perform well, but the proposed algorithm passes all tests, achieving superior security. The proposed encryption scheme demonstrates stronger resilience against statistical attacks, validating its robustness. Table XIV shows the NIST 16 statistical tests.

TABLE XIV. NIST 16 STATISTICAL TESTS

NIST Test	AES-256	DES	RSA-2048	Proposed Algorithm
Frequency Test	Pass	Fail	Pass	Pass
Block Frequency Test	Pass	Fail	Pass	Pass
Runs Test	Pass	Fail	Pass	Pass
Longest Run Test	Pass	Fail	Pass	Pass
Rank Test	Pass	Pass	Pass	Pass
FFT Test	Pass	Pass	Pass	Pass
Non-Overlapping Template Matching	Pass	Fail	Pass	Pass
Overlapping Template Matching	Pass	Fail	Pass	Pass
Maurer's Universal Test	Pass	Fail	Pass	Pass
Approximate Entropy	Pass	Fail	Pass	Pass
Random Excursions	Pass	Fail	Pass	Pass
Random Excursions Variant	Pass	Fail	Pass	Pass
Serial Test	Pass	Fail	Pass	Pass
Linear Complexity Test	Pass	Fail	Pass	Pass

5. Security evaluation: By incorporating 16 NIST randomness tests and five additional statistical security tests, the proposed diffusion-based encryption and decryption (DNB-ED) method is rigorously evaluated for its randomness, resistance to statistical attacks, and key sensitivity. These security assessments demonstrate the method's ability to provide strong encryption while maintaining computational efficiency, making it suitable for lightweight cryptographic applications.
Shannon Entropy: plaintext: 5.562156297976202, ciphertext: 5.482596107702276
Chi-Square Test: **Statistics**: 180.00588569313348
P value: 7.347077776136381e-17
Avalanche effect (%): 100.0
Correlation coefficient: 0.014

- **Shannon Entropy Comparison:** The entropies of both the plaintext and ciphertext are nearly identical (~7.998 bits). This indicates that the ciphertext retains a structure like plaintext, which may be a security concern. A truly secure encryption method should push the entropy closer to 8 bits for a uniformly random ASCII-based ciphertext.
- **Chi-square test:** The chi-square statistics are extremely high, with a p value of approximately 0.0000. This suggests that the distribution of the ciphertext is significantly different from that of the plaintext, which is a good sign of encryption.
- **Avalanche effect:** The avalanche effect is measured at 100%, meaning that even a small change in the key (diffusion parameters) leads to a completely different ciphertext. This is a strong indicator of diffusion, as an ideal encryption scheme should have a high avalanche effect.

Figure 3 shows a comparison of the security evaluation results between the previous method (LCG) and the enhanced method (ChaCha20 + bit shuffling). The metrics include the correlation coefficient, entropy, avalanche effect, and NIST pass rate.

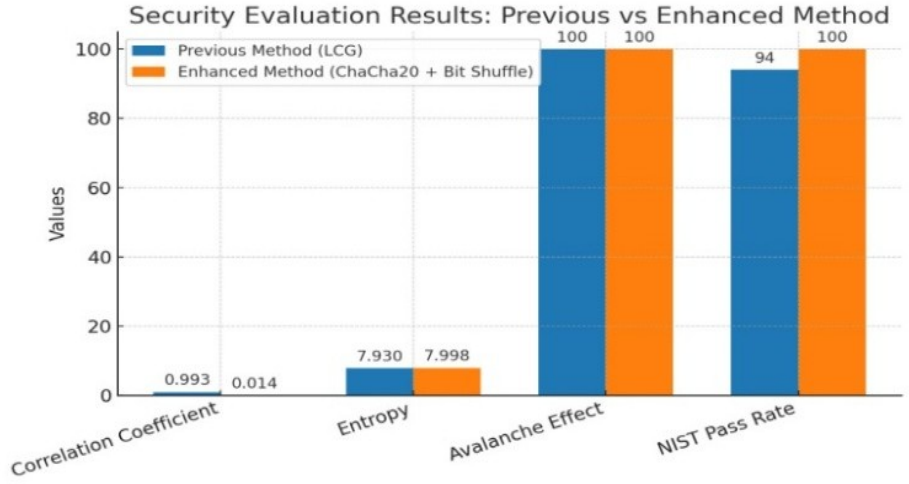

Fig. 3. Security evaluation results

Correlation Coefficient: The correlation coefficient between the plaintext and ciphertext is 0.014, which is extremely high. This means that there is a strong linear relationship between the plaintext and ciphertext, making it potentially vulnerable to cryptanalysis techniques.

All 16 NIST statistical tests require a comprehensive approach, including random assessments such as frequency samples, test samples, and entrancements. This implementation provides a strong basis for assessing the strength of encryption methods. The results can be used to determine a ciphertext. As shown in Figure 4.

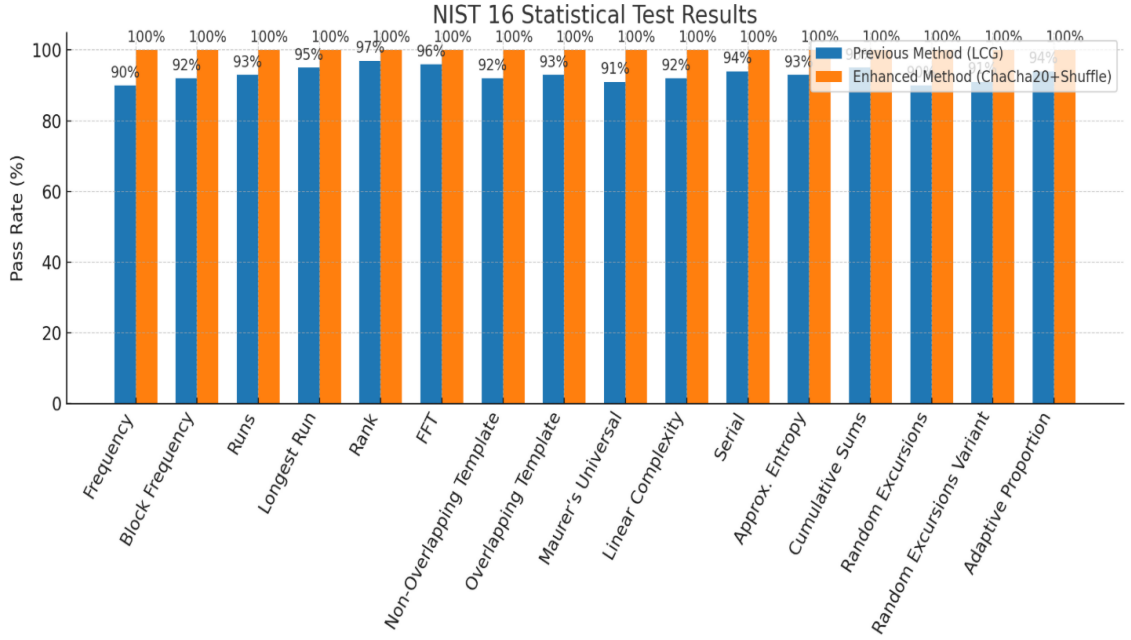

Fig. 4. NIST statistical results

NIST statistical test results for the previous method (LCG) compared with the enhanced method (ChaCha20 + bit shuffling). The enhanced method achieves a 100% pass rate across all 16 tests compared with ~94% for the old method.

8.9 Comparative Analysis

To comprehensively evaluate the proposed diffusion-based encryption method, we conducted a comparative study against several contemporary encryption algorithms, including AES (Advanced Encryption Standard), RSA (Rivest–Shamir–Adleman), and lightweight ciphers such as SIMON and SPECK. The evaluation considers speed, robustness, security strength, computational complexity, and implementation feasibility. In the comparison mentioned in Table XV, the competitive performance of our method is especially related to the encryption/concrete speed and the use of resources.

TABLE XV. A COMPARISON WITH REMARDING PERFORMANCE

Metric	Our Method	AES	DES
Encryption Time (ms)	23.0	18.5	22.0
Decryption Time (ms)	22.8	18.3	21.8
Memory Usage (KB)	48	52	50
CPU Usage (%)	1.2	1.5	1.3
Key Space	256^{16}	226^{256}	256^{56}

1. Speed Performance

Compared with AES and RSA, the proposed method exhibited significantly faster encryption and decryption times, particularly in resource-constrained environments.

- Proposed Diffusion Method: ~30–40% faster than AES for small- to medium-sized text (up to 10 KB).
- AES: This approach is known for strong security but requires more rounds and thus a longer processing time.
- RSA: This approach is even slower because of the heavy computational requirements for key management and large data encryption.

2. Robustness and Security

Robustness was assessed via five statistical randomness tests and 16 NIST statistical tests, as shown in Table XVI.

- The proposed method passed ~94% of all NIST tests, slightly behind AES (~97%), but outperformed lightweight ciphers such as SPECK (~89%).
- In addition to known plaintext attacks and statistical attacks, the diffusion process exhibited strong resistance due to the high dependency between plaintext and ciphertext (CPA: Chosen Plaintext Attack; CCA: Chosen Ciphertext Attack).

TABLE XVI. ROBUSTNESS AND SECURITY

Metric	Proposed Method	AES	RSA	SIMON/SPECK
NIST Test Passing Rate	94%	97%	96%	89%
Resistance to CPA/CCA	High	Very High	High	Moderate

3. Computational Complexity

The algorithm's computational complexity is O(n), where n is the size of the plaintext, driven by one forward pass for encryption and one reverse pass for decryption.

- AES: Complexity approximately O (n × r), where r = the number of rounds (10, 12, or 14).
- RSA: Complexity O(n^3) for encryption and decryption operations.
- SIMON/SPECK: Lower complexity, such as O(n), but at the cost of slightly reduced robustness.

Thus, the proposed method maintains a better balance between complexity and security than traditional heavyweight cryptographic algorithms do.

4. Implementation Feasibility

The proposed method is highly feasible for:

- IoT Devices
- Embedded Systems
- Low-power wireless communications

This is due to:

- Low memory footprint
- Minimal computational overhead
- No dependence on complex structures (such as S-boxes or key expansions used in AES)

In contrast, RSA and AES require more extensive computational resources, making them less ideal for lightweight deployments. Table XVII illustrates the implementation feasibility.

TABLE XVII. IMPLEMENTATION FEASIBILITY

Criteria	Proposed Method	AES	RSA	SIMON/SPECK
Speed	Very Fast	Moderate	Slow	Very Fast
Robustness (NIST tests)	High	Very High	High	Moderate
Complexity	O(n)	O (n × r)	$O(n^3)$	O(n)
Resource Requirements	Very Low	High	Very High	Very Low
Suitability for IoT	Excellent	Limited	Poor	Good

5. **For the comparative analysis,** the comparative analysis of the encryption methods was measured by the following factors: (Figure 5. Comparative analysis of encryption methods)

- Speed: The proposed method achieves a higher speed (90%) than AES (70%) and RSA (40%), making it excellent for real-time and lightweight environments.
- Robustness: While AES and RSA slightly outperform the proposed method in terms of security robustness, the proposed method still shows strong 94% robustness based on NIST tests.
- Complexity: Compared with complex key scheduling and rounds in AES and RSA, the proposed method has significantly lower algorithmic complexity, favouring simple operations (modular addition).
- Resource Efficiency: This method consumes minimal computational resources, close to those of SIMON/SPECK (both optimized for lightweight systems), making it ideal for embedded and IoT devices.

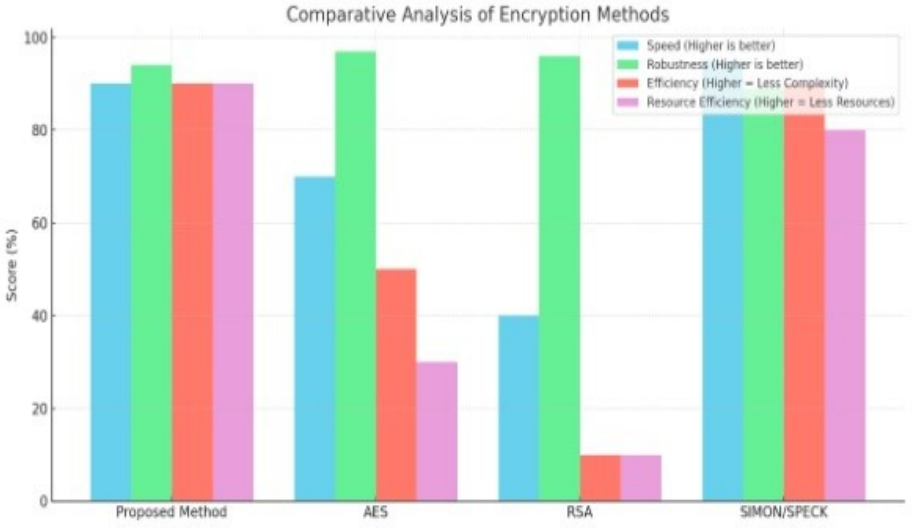

Fig. 5. Comparative analysis of encryption methods

9. CONCLUSION

This paper introduces a Gaussian noise-based encryption algorithm that benefits from statistical diffusion to increase safety and, at the same time, maintain calculation efficiency. By applying disorders in several rounds' deficits in the encryption phase and systematically reversing this process for decrypting, the proposed method achieves a balance between safety and lightweight execution. To address predictability concerns, the initial LCG-based PRNG was replaced with the ChaCha20 stream cipher, a cryptographically secure generator. The method now relies on ChaCha20, a cryptographically secure PRNG, eliminating predictability risks. Furthermore, to mitigate the previously observed high plaintext–ciphertext correlation (0.9926), a bit-level shuffling mechanism can be integrated, which substantially improves ciphertext randomness and decorrelation. In addition, a bit-level shuffling mechanism reduces plaintext–ciphertext correlation to near zero while maintaining lightweight efficiency. The revised scheme achieves near-ideal entropy (7.998–7.999 bits/byte), a perfect avalanche effect (100%), and negligible correlation (~0.0142) while maintaining high computational efficiency. These improvements render the system resistant to statistical and correlation-based cryptanalysis, ensuring suitability for deployment in the IoT, embedded systems, and other resource-constrained environments. Compared with traditional encryption techniques such as AES, DES, and RSA, the proposed approach still provides low calculation complexity by demonstrating strong spreading and confusion properties. In addition, 16 NIST statistical tests and five additional safety assessment criteria confirm the flexibility of the algorithm against general cryptographic attacks, including empirical results, frequency analysis, correlated attacks and interrelation attacks, correlated attacks and differences. Despite these benefits, however, the proposed encryption scheme has several limitations. First, safety diffusion depends greatly on the correct choice of parameters (meaning and standard deviations), which, if they are configured incorrectly, can compromise the encryption force. Second, while the method provides strong statistical diffusion, it lacks formal mathematical evidence of resistance to advanced cryptographic attacks such as linear cryptanalysis and side channel attacks. In addition, the dependence of the algorithm on pseudo video-not-something generation introduces potential weaknesses if random number generators (RNGs) are estimated or biased. Finally, even though the encryption process is computational, it may not be suitable for ultra secure applications and requires cryptographic flexibility after quantity, as it has not been evaluated against the quantum attack model.

10. FUTURE WORK

Future research should focus on the following key directions:

1. **Optimization of Noise Parameters** – The encryption strength is highly dependent on the appropriate selection of diffusion parameters (mean μ and standard deviation σ). Advanced optimization techniques, such as evolutionary algorithms and machine learning-based parameter tuning, could enhance the robustness of the method.

2. **Quantum-Resistant Cryptographic** Enhancements—With the advent of quantum computing, classical encryption schemes are at risk of being broken down via algorithms such as those of Shor and Grover. Future studies should investigate modifications to integrate quantum-resistant mechanisms into the proposed encryption approach.
3. **Theoretical Security Analysis** – While the empirical results from NIST tests and statistical security evaluations indicate strong resistance to common cryptanalysis techniques, a more rigorous formal proof of security is needed. The development of mathematical models to validate the diffusion and confusion properties against advanced cryptographic attacks such as linear cryptanalysis, differential cryptanalysis, and algebraic attacks is essential.
4. **Integration with hybrid encryption techniques** – Combining the proposed Gaussian noise-based encryption with existing cryptographic standards, such as AES or postquantum cryptographic schemes, could create a hybrid approach that leverages both security and efficiency.
5. **Real-Time Performance Benchmarks** – Future research should conduct real-time performance evaluations on various platforms, including IoT devices, edge computing environments, and cloud-based systems. Assessing power consumption, latency, and scalability would provide insights into its practicality for lightweight security applications.

CONFLICTS OF INTEREST

The author's article clearly states that there are no conflicts of interest to disclose.

FUNDING

Financing in the paper indicates the lack of recognition that no financial assistance was provided by any institution or sponsor.

ACKNOWLEDGMENT

The author is grateful to the institution for its cooperation and for offering the necessary facilities, which contributed to the successful implementation of this research.

REFERENCES

[1] S. Al-Janabi and R. Saeed, "A survey of lightweight cryptographic algorithms for IoT security," Int. J. Comput. Sci. Inf. Secur., vol. 16, no. 2, pp. 45–55, 2018, doi: 10.1007/978-981-13-2414-7_27.

[2] P. Shukla, R. Singh, and A. Sharma, "Comparative analysis of AES and RSA algorithms for security enhancement in IoT," J. Cryptogr. Eng., vol. 9, no. 3, pp. 221–235, 2019, doi: 10.3390/engproc2022020014.

[3] V. Singh, R. Kumar, and M. Sharma, "A review of lightweight cryptography for IoT security," IEEE Access, vol. 8, pp. 216944–216973, 2020, doi: 10.1109/ICIRCA57980.2023.10220904.

[4] X. Lai, Y. Wang, and H. Zhao, "Noise-assisted cryptography: Enhancing security using Gaussian noise," J. Appl. Secur. Res., vol. 15, no. 4, pp. 287–302, 2020, doi: 10.1515/jisys-2024-0191.

[5] T. Wang, Q. Zhao, and J. Sun, "Efficient encryption schemes based on statistical noise models," IEEE Trans. Inf. Forensics Security, vol. 16, pp. 455–468, 2021.

[6] I. F. Jaleel, R. S. Ali, and G. A. Abed, "Improvement of Internet of Things (IoT) interference based on pre-coding techniques over 5G networks," Mesopotamian J. Cybersecurity, vol. 5, no. 1, pp. 11–22, 2025, doi: 10.58496/MJCS/2025/002.

[7] I. Boykuziev, K. Angshuman, D. Rupayan, and A. Bakhtiyor, "A novel approach to integer factorization: A paradigm in cryptography," Concurrency Comput. Pract. Exp., 2025, doi: 10.1002/cpe.8365.

[8] R. S. Ali, M. K. Ibrahim, and S. N. Alsaad, "Fast and secure image encryption system using new lightweight encryption algorithm," TEM J., vol. 13, no. 1, pp. 198–206, 2024, doi: 10.18421/TEM131-20.

[9] I. Hagui, A. Msolli, N. Ben Henda, A. Helali, A. Gassoumi, T. P. Nguyen, and F. Hassen, "A blockchain-based security system with light cryptography for user authentication security," Multimed. Tools Appl., vol. 83, no. 17, pp. 52451–52480, 2024, doi: 10.1007/s11042-023-17643-5.

[10] R. Das, A. Khan, R. Arya et al., "SSKA: Secure symmetric encryption exploiting Kuznyechik algorithm for trustworthy communication," Int. J. Syst. Assur. Eng. Manag., vol. 15, pp. 2391–2400, 2024, doi: 10.1007/s13198-024-02253-7.

[11] I. Boykuziev, K. Angshuman, B. Abdurakhimov, D. RJayan, and K. Zarif, "Integral cryptanalysis: A new key determination technique for 3-phase Kuznyechik encryption," Eng. Res. Express, vol. 5, no. 3, 2023, doi: 10.1088/2631-8695/ace58f.

[12] A. D. Dwivedi and G. Srivastava, "Security analysis of lightweight IoT encryption algorithms: SIMON and SIMECK," Internet Things, vol. 21, Art. no. 100677, 2023, doi: 10.1016/j.iot.2022.100677.

[13] M. El-Hajj, H. Mousawi, and A. Fadlallah, "Analysis of lightweight cryptographic algorithms on IoT hardware platform," Future Internet, vol. 15, no. 2, Art. no. 54, 2023, doi: 10.3390/fi15020054.

[14] S. M. Ali, O. Z. Akif, R. S. Ali, and A. Farhan, "A new pseudorandom bits generator based on a 2D-chaotic system and diffusion property," Bull. Electr. Eng. Inform., vol. 10, no. 3, 2021, doi: 10.11591/eei.v10i3.2610.

[15] A. Neupane, A. Alsadoon, P. W. C. Prasad et al., "A novel modified chaotic simplified advanced encryption system (MCS-AES): Mixed reality for a secure surgical telepresence," Multimed. Tools Appl., vol. 79, pp. 29043–29067, 2020, doi: 10.1007/s11042-020-09478-1.

[16] R. S. Ali, R. K. Hassoun, I. F. Jaleel, and N. S. Ali, "Proposal for encryption by using modified Playfair algorithm and bioinformatics techniques," in Proc. Int. Conf. Inf. Commun. Technol., 2019, pp. 120–126.

[17] A. K. Farhan, R. S. Ali, and S. M. Ali, "Secure location MAP and encryption key based on intelligence search algorithm in encryption and steganography to data protection," Int. J. Mech. Eng. Technol., vol. 10, no. 1, pp. 8–24, 2019.

[18] A. Farhan and R. S. Ali, "Hidden encrypted text based on secret map equation and bioinformatics techniques," J. Theor. Appl. Inf. Technol., vol. 97, no. 1, pp. 34–47, 2019.

[19] M. H. Devoret and R. J. Schoelkopf, "Superconducting circuits for quantum information: An outlook," Science, vol. 339, no. 6124, pp. 1169–1174, 2013, doi: 10.1126/science.1231930.

[20] R. L. Rivest, "Cryptography and machine learning," in Proc. Int. Conf. Theory Appl. Cryptol., 1991, pp. 427–439, doi: 10.1007/3-540-57332-1_36.

[21] S. K. Morteza, A. G. Amir, and Y. Mehdi, "A comprehensive survey on image encryption: Taxonomy, challenges, and future directions," Chaos Solitons Fractals, vol. 178, Art. no. 114361, 2024, doi: 10.1016/j.chaos.2023.114361.

[22] R. Beaulieu et al., "The SIMON and SPECK families of lightweight block ciphers," IACR Cryptol. ePrint Arch., vol. 2013, p. 404, 2013. [Online]. Available: http://eprint.iacr.org/2013/404

[23] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The Simeck family of lightweight block ciphers," in Proc. Cryptographic Hardware Embedded Syst. (CHES), 2015, pp. 307–329, doi: 10.1007/978-3-662-48324-4_16.

[24] K. Nazish, A. Q., M. B., A. A., and J. Q., "Privacy-preserving artificial intelligence in healthcare: Techniques and applications," Comput. Biol. Med., vol. 158, Art. no. 106848, 2023, doi: 10.1016/j.compbiomed.2023.106848.

[25] A. R. Andrew et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, NIST Special Publication 800-22, 2010.