Code-Based Cryptography and Chaotic Maps as Pseudo-Random Bit Generator

Tayseer Karam Alshekly
Department of Computer Science, Al-Imam Al-Adham University, Baghdad, Iraq
tayseer.alshekly@imamaladham.edu.iq (corresponding author)

Ekhlas Abbas AlBahrani
Department of Computer Science, Mustansiriyah University, Baghdad, Iraq
akhlas_abas@uomustansiriyah.edu.iq

Leila Ben Ayed
Department of Computer Science, University of Manouba, Tunisia
leilabenayed@ensi-uma.tn

Received: 29 July 2025 | Revised: 27 August 2025 and 8 September 2025 | Accepted: 11 September 2025

 DOI: https://doi.org/10.48084/etasr.13718

ABSTRACT

Random number generation is a fundamental requirement in cryptography, as the security of secret keys depends on their unpredictability. Yet, many existing Pseudo-Random Number Generators (PRNGs) face challenges such as limited key space, weak statistical performance, high computational cost, and vulnerability to post-quantum attacks. These limitations restrict their use in applications such as the Internet of Things (IoT), where both efficiency and strong security are needed. To address this problem, this paper presents a new PRNG that combines Code-based Cryptography (CBC) with a 3D logistic chaotic map. Chaotic maps provide sensitivity to initial conditions and a large key space, whereas code-based encryption offers resistance to quantum adversaries. The proposed scheme is based on the Cipher Feedback (CFB) mode, where chaotic outputs are converted into binary sequences and applied as key arrays. Due to its lightweight design, the generator is suitable for resource-constrained devices, such as IoT nodes. Experimental results show that the generated sequences pass the National Institute of Standards and Technology (NIST) statistical test suite with p-values between 0.98056 and 0.99944. Additional tests confirm low inter-sequence correlation (Pearson Correlation Coefficient (PCC) within -0.16591–0.14187), balanced Hamming distances (0.429–0.583), and favorable Sum of Absolute Differences (SAD) results (0.3173–0.3849). Overall, the proposed PRNG achieves efficiency, scalability, and post-quantum security, making it a strong candidate for future cryptographic systems.

Keywords-Code-based Cryptography (CBC); 3D logistic map; Pseudo-Random Number Generator (PRNG); National Institute of Standards and Technology (NIST) test

I. INTRODUCTION

Pseudo-Random Number Generators (PRNGs) play a critical role in cryptography, as the security of secret keys depends on the unpredictability of generated sequences. However, many existing PRNGs face challenges, including limited key space, weak statistical properties, high computational cost, and vulnerability to post-quantum attacks. This paper proposes a PRNG that overcomes these limitations by combining multi-dimensional chaotic maps with a code-based encryption algorithm. The proposed approach ensures high unpredictability, a large key space, and suitability for resource-constrained devices. Authors in [1] presented a PRNG that improves statistical randomness and cryptographic security by fusing chaotic dynamics with a structured look-up table. Their solution makes use of a Piecewise Linear Chaotic Map (PWLCM) and a modified skew tent map, both of which are integrated into a recursive architecture. Permutation and XOR procedures are used to combine the outputs of these chaotic maps with values chosen from an 8-bit, non-repeating look-up database. The goal of this integration is to create a sequence that is non-linear and non-invertible, with characteristics that are very similar to those of True Random Number Generators (TRNGs). Chi-square tests, autocorrelation, cross-correlation, entropy analysis, and National Institute of Standards and Technology (NIST) statistical suites were used to thoroughly test the suggested PRNG. With a Hamming distance close to the optimal 50%, the findings show good avalanche behavior, minimal correlation, and strong uniformity. With a large key space (up to 555 bits) and sensitivity to key changes, the security analysis validates the resilience against statistical and brute-force assaults. The suggested generator outperformed

traditional PRNGs like RC4 and Salsa20 in terms of statistical homogeneity and computational performance, which qualifies it for real-time applications and safe multimedia encryption.

Authors in [2] introduced a Pseudo-Random Bit Generator (PRBG) that was improved by IEEE 754-2008 double-precision floating-point arithmetic and was based on the chaotic behavior of three connected logistic maps. To improve sensitivity to initial conditions and computational unpredictability—two essential characteristics for cryptographic applications—the generator uses the accuracy and uniformity of IEEE floating-point operations. Three logistic maps are initialized with distinct seeds, and their outputs are iteratively combined using bit extraction, floating-point normalization, and modular arithmetic. This approach guarantees non-linearity and unpredictability while increasing entropy. The generator passed all of the statistical randomness tests that the authors used to validate their methodology, which included NIST SP800-22, TestU01, and ENT. Strong key sensitivity, a 192-bit key space, and resistance to frequent cryptanalytic attacks are all features of the suggested system. It is appropriate for embedded cryptography systems and secure communication protocols due to its robustness and computational efficiency.

A PRNG, based on a one-dimensional discrete-space chaotic map created via the composition of permutations was proposed in [3]. By staying entirely digital, this approach avoids dynamical degradation and does not require floating-point approximation, in contrast to conventional continuous chaotic maps. Long cycle lengths, high key sensitivity, and an almost infinite key space make the generator appropriate for secure cryptographic applications, especially in low-memory settings. The suggested PRNG's randomness was confirmed by passing the NIST SP800-22 and TestU01 statistical test suites. Furthermore, it surpasses several current chaos-based PRNGs in terms of memory efficiency and security by offering tunable parameters (such as output bit width and permutation size) that allow for adjustable security and performance trade-offs. Authors in [4] created a new PRBG for secure image encryption by utilizing one-dimensional chaotic maps. The design focuses on the electronic hardware implementation of the tent and logistic maps, which is done with analog components such as sample-and-hold circuits (LF398), operational amplifiers, and multipliers (AD633). This hardware-based method offers benefits in embedded cryptography systems by guaranteeing that the entropy source is based on the intrinsic unpredictability of chaotic dynamics. The system uses a multi-stage post-processing pipeline that includes an H-function post-processor, XOR processing, and fixed-point binary conversion to enhance the statistical quality of the generated chaotic signals. For these steps, the paper offers comprehensive pseudo-code, emphasizing design clarity and reproducibility. The NIST SP800-22 test suite was used to thoroughly verify the output bit sequences' randomness, ensuring their suitability for use in cryptography. Additionally, the PRBG was used to encrypt color images, proving its practicality. Strong resistance to statistical and differential attacks was confirmed by a thorough cryptanalysis of the encrypted images, which included histogram analysis, Number of Pixel Change Rate (NPCR)–Unified Average Changing Intensity (UACI) metrics, and correlation coefficient evaluation. This work combines circuit-level design, efficient randomness conditioning, and demonstrated cryptographic durability to provide a workable, hardware-oriented solution for chaos-based cryptography.

In [5], authors presented a hybrid pseudo-random key generator for image encryption that combines a Genetic Algorithm (GA), trellis-coded modulation, and fractal-based initial vectors. The Sierpinski triangle is used to generate fractal keys, and an Initial Vector (IV) is obtained by processing the triangle using a Non-Deterministic Finite Automaton (NDFA). The resilient 32-bit trellis code produced by this IV seeds the Trellis-Coded Genetic Algorithm (TCGA), which in turn seeds a Session Key Matrix (SKM) for image encryption. The NIST statistical test suite and entropy metrics were used to assess the pseudo-random output of the key generator, which showed good randomness efficiency. The NPCR and UACI metrics were used to further examine the encrypted images, which demonstrated robust defense against statistical and differential attacks. By combining fractal complexity, structured coding, and evolutionary optimization, the system provides high entropy, unpredictability, and robustness, making it ideal for high-security image applications.

A PRBG based on the Rössler chaotic system was proposed in [6], which uses the nonlinear dynamics of the system to generate high-entropy sequences. The generator produces the final bitstream by sampling the Rössler system's continuous outputs, converting them into integers, and then performing binary conversions and XOR operations ($x \oplus y$, then $\oplus z$). This design's unique characteristic is the dynamic modification of the system's parameter "c" through the use of a randomly shuffled array, which increases unpredictability between cycles. Statistical tests, including the autocorrelation, runs, and monobit tests, were used to assess the generator's performance; the results were generally positive, particularly for sequences up to 10 bits. The method shows advantages in terms of applicability for portable systems that use only positive integers, sensitivity to initial conditions, and ease of implementation.

To improve the security and randomness of cryptographic key streams, a unique key stream generator was created in [7] by fusing the 3D Henon map and the 3D cat map. To generate pseudo-random sequences, the method exploits both maps' intricate dynamic behavior and sensitive reliance on initial conditions. The generator obtains stronger statistical features and higher entropy by synchronizing and applying both chaotic systems iteratively. Its applicability to cryptography is confirmed by combining the Henon and cat map outputs into a final key stream, verified with NIST tests. Large key space, unpredictability, and strong defense against brute-force and statistical attacks are guaranteed. The approach generates random integers via the 3D Henon map, transforms them into a binary sequence, and XORs and permutes them using the 3D cat map. The Code-based Cryptography (CBC) system and 3D logistic map are used in this work to create a robust PRNG that can produce 136 bits in a single iteration. Three initial values, floating point numbers between 0 and 1 with 10^{-16} precision,

generate the encryption keys. A combination of the code-based encryption algorithm output and 3D logistic system results produces secure pseudo-random numbers for cryptographic applications.

To address the limitations of existing PRNGs, this work proposes a PRNG that leverages the features of public-key algorithms, particularly a code-based encryption algorithm, which has proven resilient to adaptive chosen ciphertext attacks. Code-based cryptosystems rely on complex algebraic structures and error-correcting codes, providing inherent unpredictability that can enhance PRNG outputs. Specifically, the proposed generator benefits from:

- Hard-to-invert functions: NP-hard problems, such as syndrome decoding, are used in CBC.
- Large key spaces: PRNG outputs may have high entropy due to the randomness of key generation.
- Quantum resilience: Code-based PRNGs are more resistant to quantum attacks than conventional PRNGs based on number-theoretic methods (e.g., RSA, ECC).

The core principle of the proposed generator is based on the Cipher Feedback (CFB) mode, in which the ciphertext of any plaintext unit depends on all preceding plaintext units. The PRNG algorithm combines the outputs of the code-based encryption algorithm with the results of a 3D logistic map, achieving a balance between security strength, area, and computational performance.

The effectiveness of the proposed PRNG is validated through several analyses, including key space evaluation, correlation assessment, and key sensitivity testing. In addition, the NIST statistical test suite is applied to 1000 distinct random binary sequences generated with similar initial values, ensuring high randomness and low inter-sequence correlation. These results demonstrate the robustness, unpredictability, and suitability of the proposed system for cryptographic applications.

II. FUNDAMENTAL HYPOTHESIS

The proposed algorithm in this work relies on the 3D logistic system and the code-based encryption algorithm. The logistic map is a degree-2 discrete recursive relation that is widely used due to its simplicity. It provides a classic example of how a simple nonlinear dynamic equation can exhibit chaotic behavior. The one-dimensional logistic function is defined as:

$$x_{(n+1)} = r x_n (1 - x_n) \quad (1)$$

Three-dimensional versions of well-known chaotic maps, designed for image encryption, are presented by authors in [8]. They are defined by the equations:

$$\begin{cases} x_{i+1} = \gamma(1 - x_i) + \beta(y_i^2 x_i) + \alpha z_i^3 \\ y_{i+1} = \gamma(1 - y_i) + \beta(z_i^2 x_i) + \alpha x_i^3 \\ z_{i+1} = \gamma(1 - z_i) + \beta(x_i^2 y_i) + \alpha y_i^3 \end{cases} \quad (2)$$

This system exhibits chaotic behavior for:

$$3.53 < \gamma < 3.81, 0 < \beta < 0.022, 0 < \alpha < 0.015$$

Authors in [9] introduced the first code-based public key encryption system in 1978. CBC relies on error-correcting codes and is well-known for its high security, resilience to quantum attacks, and strong foundations in hard mathematical problems. In the proposed method, only the encryption algorithm of CBC is utilized. The following system parameters and key generation steps are used in the proposed encryption scheme:

- System parameters: $n, t \in \mathbb{N}$, where $t \ll n$.
- Key generation: Given parameters n , t generate the following matrices:
  - G : $k \times n$ generator matrix of a code G over F of dimension k and minimum distance $d \geq 2t + 1$ (a binary irreducible Goppa code in the original proposal).
  - S: $k \times k$ non-singular random binary matrix.
  - P: $n \times n$ random permutations matrix.
- Compute the $k \times n$ public generator matrix: $G^{pub} = SGP$.
- Public key: (G^{pub}, t),
- Private key: (S, D_G, P), where D_G is the decoding algorithm suitable for G.
- Encryption ($E_{G^{pub},t}$): To encrypt a plaintext $m \in F^k$ choose a vector $z \in F^n$ of weight t at random and compute the ciphertext:

$$c = m\, G^{pub} \oplus z \quad (3)$$

III. PROPOSED PSEUDO-RANDOM NUMBER GENERATOR ALGORITHM

The fundamental concept of the proposed PRNG is derived from the CFB mode, utilizing a chaotic system and a public-key algorithm. The primary steps of the suggested system are outlined below.

A. Input Step

Three floating point numbers with a precision of 10^{-16} (X_0, Y_0, Z_0) serve as the key of the proposed PRNG. These values, along with the required number of bits to be generated, are input into the algorithm.

B. Preprocessing Step

The 3D logistic map in (2) is iterated to generate a 128-bit plaintext. Each output of the logistic map is mapped to a binary bit according to the following intervals:

- If the result is in [0.0, 0.15], [0.3, 0.54], or [0.60,0.75], the corresponding bit is 0.
- If the result is in [0.15, 0.3], [0.45,0.60], or [0.75,0.9], the corresponding bit is 1.

C. Blocking Step

The resulting 128-bit plaintext is divided into four blocks, each consisting of 32 bits. For each block, the following steps are performed:

- The 3D logistic map in (2) is iterated to generate three matrices: G [32 × 34], S [32 × 32], and P [34 × 34], which serve as the keys for the code-based encryption algorithm.
- Encrypt the 32-bit plaintext block using the code-based encryption algorithm. Each ciphertext block consists of 34 bits, longer than the plaintext.
- XOR the next plaintext block with the previous ciphertext block to form the new plaintext block.

D. Concatenation Step

The four resulting 34-bit ciphertext blocks are concatenated into a single ciphertext of 136 bits.

E. Permutation Step

The 3D logistic map in (2) is iterated to generate 136 numbers in the range [1, 136]. These numbers serve as the new indices for the ciphertext. The ciphertext values are permuted according to these indices. The permuted ciphertext represents one round of the proposed PRNG.

F. Repeating Step

The steps are repeated to generate the desired number of bits. Figure 1 shows the block diagram for one round of the proposed algorithm.

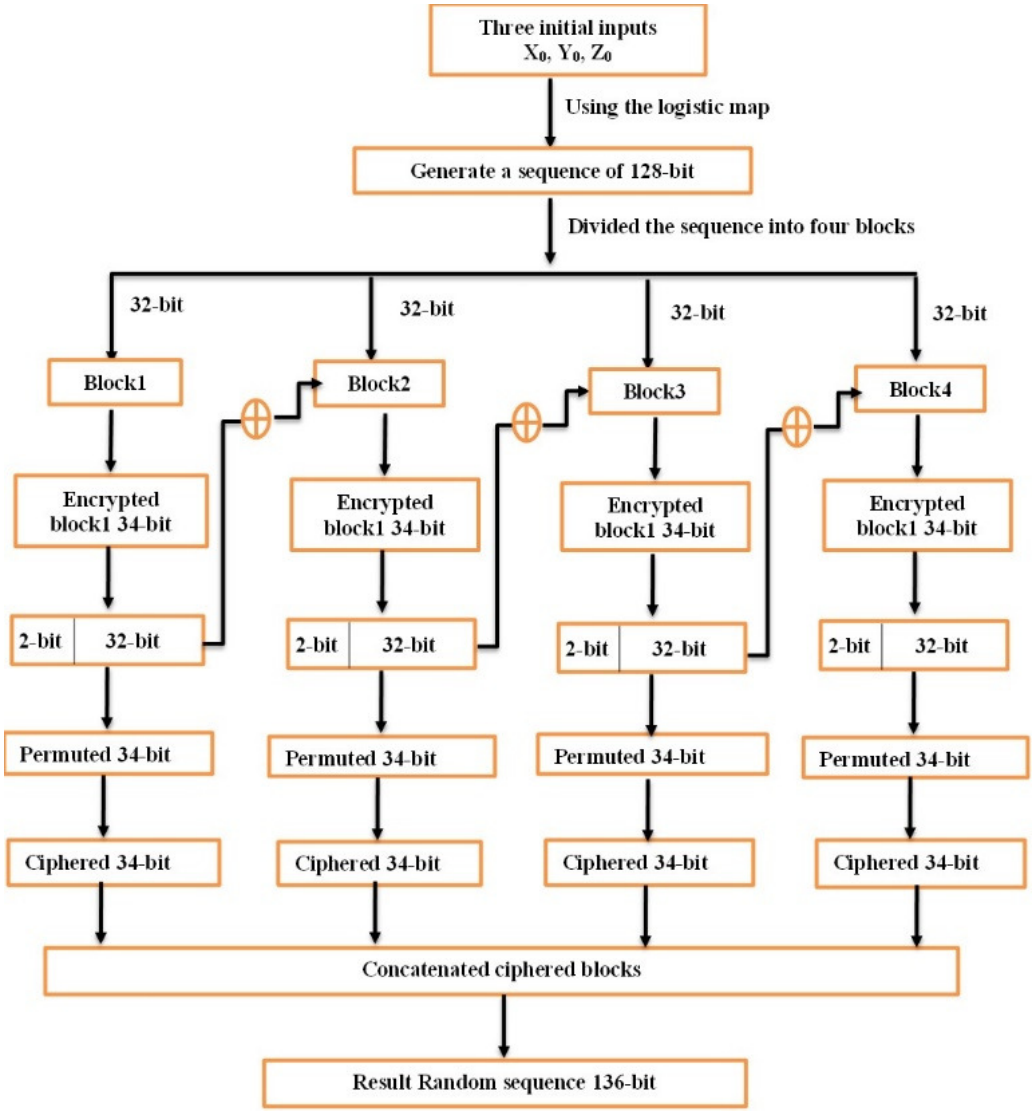

Fig. 1. Block diagram of one round of the proposed PRNG.

IV. PERFORMANCE ANALYSIS

The output sequences of the proposed PRNG must be completely decorrelated, random, and secure. To assess the security and effectiveness of the proposed method, several cryptographic and statistical tests are conducted.

A. Statistical Analysis

Each output binary sequence must exhibit high randomness and minimal correlation with others, regardless of the initial seed values. The statistical analysis demonstrates the characteristics and quality of the generated pseudo-random binary sequences.

1) Randomness Test

The NIST statistical test suite [10] is used to evaluate randomness. Binary sequences are generated from closely spaced seed values to ensure robustness. In total, 1,000 distinct binary sequences, each 1,088 bits long, are generated using different initial values. Each sequence is tested with all fifteen NIST tests.

The significance level is set to $\alpha = 0.010$, meaning that only 1% of the generated sequences are allowed to fail. The p-value for each test is compared with α, and the statistical test is considered successful if the p-value is greater than or equal to α. The proportion of sequences passing a test simultaneously is denoted by η.

The ratio η is compared with a suitable confidence interval, which defines the permissible range of proportions:

$$p \pm 3\sqrt{p(1-p)/n} \quad (4)$$

where $p = 1 - \alpha = 0.990$ and $n = 1000$ sequences. Thus, the confidence interval is:

$$0.990 \pm 3\sqrt{(0.990 \times 0.010)/1000} = 0.990 \pm 0.00944$$

i.e., $[0.98056, 0.99944]$.

The NIST tests are carried out on two types of sequences: concatenated sequences and individual sequences. To examine the randomness of the proposed method, the NIST tests are divided into two groups based on sequence length requirements. The first group includes tests suitable for short sequences (e.g., individual sequences of 1,000 bits), whereas the second group includes tests designed for long sequences (e.g., concatenated sequences of 1,000,000 bits). Thus, 1,000 distinct sequences with 1,000 bits each are produced. The first group of tests is applied to individual sequences, whereas both groups are used for concatenated sequences. The proportion of sequences passing each NIST test should fall within the confidence interval [0.98056, 0.99944]. Tables I and II present the results of the NIST tests for the two groups.

2) Correlation Test

To examine the relation between the generated sequences, the Pearson Correlation Coefficient (PCC) is calculated [11]. For two sequences $Str_1 = [x_1, \ldots, x_N]$ and $Str_2 = [y_1, \ldots, y_N]$, the PPC is:

$$C_{Str_1,Str_2} = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x})\cdot(y_i - \bar{y})}{[\sum_{i=0}^{N-1}(x_i - \bar{x})^2]^{\frac{1}{2}}\cdot[\sum_{i=0}^{N-1}(y_i - \bar{y})^2]^{\frac{1}{2}}} \quad (5)$$

where the average values of Str_1 and Str_2 are: $\bar{x} = \sum_{i=0}^{N-1} x_i/N$ and $\bar{y} = \sum_{i=0}^{N-1} y_i/N$.

PCC is calculated for all pairs of the 1,000 sequences. Figure 2 shows the histogram of the resulting PCC values. The coefficients are close to zero, with 99.99% in the range [-0.1659, 0.1418], indicating minimal correlation between sequences.

TABLE I. NIST TEST RESULTS FOR 1,000 INDIVIDUAL SEQUENCES AND THEIR CONCATENATION – FIRST TEST GROUP

Test name	H of individual	Final result	p-value of concatenation	Final result
Frequency	0.980	Succ.	0.996809133895137	Succ.
Block frequency	0.989	Succ.	0.966090587896422	Succ.
Runs	0.981	Succ.	0.750535082845276	Succ.
Longest runs	0.989	Succ.	0.665096495203418	Succ.
Rank	0.99	Succ.	0.738130758428523	Succ.
FFT	0.982	Succ.	0.37645705203743	Succ.
Non-overlapping	0.989	Succ.	0.802749617829708	Succ.
Serial (1)	0.99	Succ.	0.951310058636046	Succ.
Serial (2)	0.99	Succ.	0.752052382387207	Succ.
Cumulative sum	0.988	Succ.	0.999232777670001	Succ.

a. Succ. = Successful.

TABLE II. NIST TEST RESULTS FOR 1,000,000-BIT CONCATENATED SEQUENCES – SECOND TEST GROUP

Test name	p-value of concatenation	Final result
Overlapping	0.0375411723608319	Successful
Universal	0.265592656185166	Successful
Linear complexity	0.3269151852037	Successful
Approximate entropy	0.797993572298992	Successful
Random excursions (8 p-value)	0.891898320159112 0.629608198990871 0.961276253258034 0.585597168257796 0.347964995300847 0.236842620159558 0.266676995672735 0.787183559829817	Successful
Random e-variant (18 p-values)	0.297296165993698 0.467946605983679 0.645424844631163 0.79445758067117 0.987743995533947 0.747276255307465 0.780839743994149 0.96816516526811 0.799912568052499 0.226395301721992 0.497477613350554 0.769001056525741 0.841240532585785 0.990807838495834 0.599912088865681 0.613655831317054 0.734523443811986 0.599363576929878	Successful

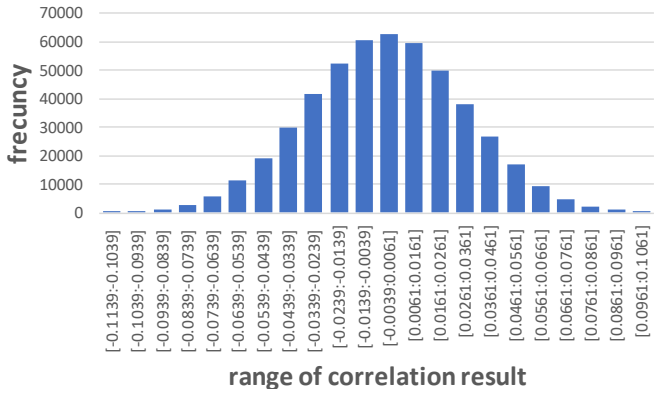

Fig. 2. Histogram of PCC values for 1,000 generated sequences, showing minimal correlation.

3) *Hamming Distance*

Hamming distance measures the difference between sequences at the bit level [12]. For two binary sequences $S = [s_0, \ldots, s_{M-1}]$ and $S' = [s'_0, \ldots, s'_{M-1}]$ of length M, the distance is:

$$d(S, S') = \sum_{j=0}^{M-1} (s_j \oplus s'_j) \quad (6)$$

For truly random sequences, the typical Hamming distance is approximately $M/2$, resulting in a normalized value $d(S, S')/M \approx 0.5$. Hamming distance is calculated for the 1,000 generated sequences, and Figure 3 shows the histogram of all Hamming distance results. The distributions indicate that 99.98% of the distances fall within [0.429, 0.583], and the mean is approximately 50%. This further confirms minimal correlation and strong sequence decorrelation.

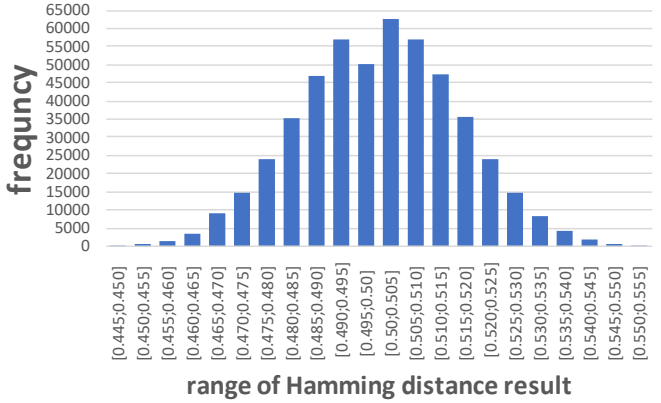

Fig. 3. Histogram of Hamming distances for 1,000 generated sequences, indicating high decorrelation and approximate 50% mean distance.

4) *Entropy Analysis*

Entropy measures the unpredictability and randomness of a sequence. Higher entropy indicates more ambiguous and less predictable information [13, 14], indicates more ambiguous and incomprehensible visual information. For a source S with 2^n possible symbols s_i and corresponding probabilities $P(s_i)$, the entropy is defined as:

$$E(S) = \sum_{i=1}^{2^n-1} P(s_i) \frac{1}{\log_2 P(s_i)} \quad (7)$$

where n represents the number of bits used to encode symbol s_i. For an ideal random source emitting 2^n symbols, $E(S) = n$. The concatenated sequences generated by the proposed PRNG have an entropy value of 7.99, which is close to the ideal value of 8. This confirms that the sequences are highly random and suitable for cryptographic applications.

B. *Security Analysis*

1) *Brute Force Attack Analysis*

The three initial values (x_0, y_0, z_0) that form the key space of the proposed PRNG are floating-point numbers with a precision of 10^{-16}. Given such a large key space, a brute-force attack is computationally infeasible. As noted in [15, 16], a key space smaller than 2^{128} is considered insecure. The proposed method provides a key space of approximately $(10^{16})^3 \approx 2^{160}$. This ample key space ensures that the encryption method can resist all brute-force attacks.

2) *Key Sensitivity Analysis*

Key sensitivity is a critical property of a PRNG, as the output sequences must be highly unpredictable and sensitive to even slight changes in the initial values [12]. To assess key sensitivity, two metrics are employed, PCC and Hamming distance. Four large binary sequences, S_1, S_2, S_3, and S_4, each of length $N = 1000{,}000$ bits, are generated using slightly different initial values:

- S_1 : x_0 =0.1247895632147892, y_0 =0.5789632014789632, and z_0=0.9510247896325472.
- S_2 : x_0 =0.1247895632147893, y_0 =0.5789632014789633, and z_0=0.9510247896325473.
- S_3 : x_0 =0.1247895632147894, y_0 =0.5789632014789634, and z_0=0.9510247896325474.
- S_4 x_0 =0.1247895632147894, y_0 =0.57896320147896334, and z_0=0.9510247896325474.

Table III summarizes the PCC and Hamming distance between all pairs of these sequences, indicating minimal correlation and high independence.

TABLE III. PCC AND HAMMING DISTANCES BETWEEN FOUR BINARY SEQUENCES GENERATED WITH SLIGHTLY DIFFERENT SEEDS

Test	S_1/S_2	S_1/S_3	S_1/S_4	S_2/S_3	S_2/S_4	S_3/S_4
PCC	0.0420	0.0230	0.0108	-0.0225	-0.0409	0.0174
Hamming distance	0.4789	0.4884	0.4945	0.5112	0.5204	0.4912

3) *Differential Attack Analysis*

Differential attacks exploit how small changes in input pairs affect the differences in corresponding outputs [17, 18]. exploit how small changes in input pairs affect the differences in corresponding outputs ($SR1R$, $SR2R$, $SR3R$, and $SR4R$), each of length of $N = 1{,}000{,}000$ bits, are analyzed. The Sum

of Absolute Differences (SAD) measures sensitivity to differential attacks [19]:

$$\mathrm{SAD}(S_1, S_2) = \frac{1}{N}\sum_{i=1}^{N} \frac{|S_1 - S_2|}{2^6} \quad (8)$$

Table IV and Figure 4 present the SAD results. The values are close to the ideal value of 1/3, indicating strong resistance to differential attacks.

TABLE IV. SAD OF FOUR KEY BINARY SEQUENCES GENERATED WITH SLIGHTLY DIFFERENT SEEDS

Sequences pairs	S_1/S_2	S_1/S_3	S_1/S_4	S_2/S_3	S_2/S_4	S_3/S_4
SAD	0.3457	0.3849	0.3638	0.3505	0.3465	0.3173

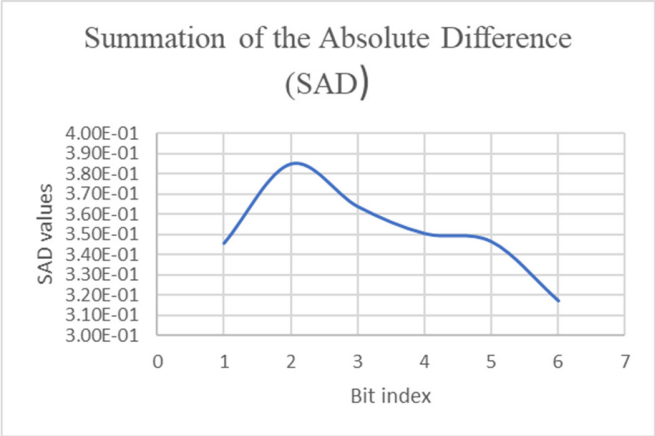

Fig. 4. Histogram of SAD values for four key binary sequences generated with slightly different seeds.

V. CONCLUSIONS

A new Pseudo-Random Number Generator (PRNG) is proposed based on the Cipher Feedback (CFB) mode, a chaotic system, and Code-based Cryptography (CBC). The proposed method combines the results of a code-based encryption algorithm and a 3D logistic system to generate secure pseudo-random numbers that can be utilized in various cryptographic applications. The algorithm was evaluated through several tests, including the National Institute of Standards and Technology (NIST) test suite, differential attack analysis, brute force attack analysis, histogram analysis, correlation analysis, and information entropy analysis.

The following outcomes are obtained from the performance analysis of the proposed algorithm:

- The algorithm has a large key space of 2^{160}, which provides resistance to brute-force attacks.
- The algorithm is highly sensitive to even slight changes in the secret keys.
- The algorithm produces low correlation coefficient values for the 10,000 sequences, ranging from 0.16591 to 0.14187.
- The algorithm yields Hamming distance coefficients between 0.429 and 0.583.
- The algorithm achieves a high information entropy value of 7.99 for the concatenated 10,000 sequences, each with 1,000 bits.

Although the proposed PRNG demonstrates high unpredictability, a large key space, and strong resistance to statistical and brute-force attacks, several limitations remain. First, its computational efficiency on very low-power Internet of Things (IoT) devices has not been fully evaluated and may require further optimization. Second, while the generator shows robustness against classical attacks, comprehensive analysis against emerging quantum attacks is still required. Third, the current study focuses on sequences up to 136 bits per iteration, and performance for longer sequences or high-throughput applications has not been extensively tested. Future work will aim to address these limitations by optimizing the algorithm for embedded systems, extending key length and throughput, and performing rigorous quantum-resistant security assessments.

ACKNOWLEDGMENT

The author would like to express sincere gratitude to everyone who contributed to the successful completion of this research. Special thanks are extended to the faculty and staff of the Al-Imam Al-Adham University and Mustansiriyah University in Baghdad, Iraq, and to the University of Manouba in Tunisia for their continuous support and guidance. The author is also grateful for the valuable feedback and encouragement received from colleagues and reviewers, which greatly improved the quality of this work.

REFERENCES

[1] M. Farajallah, M. Abutaha, M. A. Joodeh, O. Salhab, and N. Jweihan, "Pseudo Random Number Generator Based on Look-up Table and Chaotic Maps," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 20, pp. 3130–3139, Oct. 2020.

[2] M. François, D. Defour, and P. Berthomé, "A Pseudo-Random Bit Generator Based on Three Chaotic Logistic Maps and IEEE 754-2008 Floating-Point Arithmetic," in *11th Annual Conference on Theory and Applications of Models of Computation*, Chennai, India, 2014, pp. 229–247, https://doi.org/10.1007/978-3-319-06089-7_16.

[3] D. Lambić and M. Nikolić, "Pseudo-random number generator based on discrete-space chaotic map," *Nonlinear Dynamics*, vol. 90, no. 1, pp. 223–232, Oct. 2017, https://doi.org/10.1007/s11071-017-3656-1.

[4] E. İnce, B. Karakaya, and M. Türk, "Chaos Based Pseudo Random Bit Generator Design and Its Application in Secure Image Encryption," *Traitement du Signal*, vol. 39, no. 5, pp. 1647–1653, Oct. 2022, https://doi.org/10.18280/ts.390522.

[5] A. T and V. R, "Pseudo Random Key Generator Using Fractal Based Trellis Coded Genetic Algorithm for Image Encryption," *International Journal of Network Security & Its Applications*, vol. 15, no. 3, pp. 23–32, May 2023, https://doi.org/10.5121/ijnsa.2023.15302.

[6] C. Banerjee, D. Datta, and D. Datta, "A Random Bit Generator Using Rössler Chaotic System," in *1st International Conference on Computational Advancement in Communication Circuits and Systems*, Kolkata, India, 2014, pp. 81–87, https://doi.org/10.1007/978-81-322-2274-3_10.

[7] E. A. Albahrani and T. K. Alshekly, "A New Key Stream Generator Based on 3D Henon map and 3D Cat map," *International Journal of Scientific & Engineering Research*, vol. 8, no. 1, pp. 2114–2120, Jan. 2017.

[8] G. Narayanan, R. Narayanan, N. Haneef, N. B. Chittaragi, and S. G. Koolagudi, "A Novel Approach to Video Steganography using a 3D Chaotic Map," in *TENCON 2019 - 2019 IEEE Region 10 Conference*,

Kochi, India, 2019, pp. 955–959, https://doi.org/10.1109/TENCON.2019.8929347.

[9] R. Overbeck and N. Sendrier, "Code-based cryptography," in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg, Germany: Springer, 2009, pp. 95–145, https://doi.org/10.1007/978-3-540-88702-7_4.

[10] A. Rukhin *et al.*, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," National Institute of Standards and Technology, NIST Special Publication (SP) 800-22 Rev. 1, Apr. 2010. https://doi.org/10.6028/NIST.SP.800-22r1a.

[11] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," *Optics Communications*, vol. 284, no. 19, pp. 4331–4339, Sep. 2011, https://doi.org/10.1016/j.optcom.2011.05.028.

[12] L. F. Jalil, H. H. Saleh, and E. A. Albhrany, "New Pseudo-Random Number Generator System Based on Jacobian Elliptic maps and Standard Map," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 15, no. 3, pp. 77–89, Nov. 2015.

[13] W. Alexan, N. H. E. Shabasy, N. Ehab, and E. A. Maher, "A secure and efficient image encryption scheme based on chaotic systems and nonlinear transformations," *Scientific Reports*, vol. 15, no. 1, Aug. 2025, Art. no. 31246, https://doi.org/10.1038/s41598-025-15794-z.

[14] Z. A. Mohammed and K. A. Hussein, "PILEA, an Advanced Hybrid Lightweight Algorithm utilizing Logical Mathematical Functions and Chaotic Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16260–16265, Oct. 2024, https://doi.org/10.48084/etasr.7799.

[15] T. S. Ali and R. Ali, "A novel color image encryption scheme based on a new dynamic compound chaotic map and S-box," *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 20585–20609, Jun. 2022, https://doi.org/10.1007/s11042-022-12268-6.

[16] T. A. Al-Maadeed, I. Hussain, A. Anees, and M. T. Mustafa, "A image encryption algorithm based on chaotic Lorenz system and novel primitive polynomial S-boxes," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24801–24822, Jul. 2021, https://doi.org/10.1007/s11042-021-10695-5.

[17] F. Yu, X. Gong, H. Li, and S. Wang, "Differential cryptanalysis of image cipher using block-based scrambling and image filtering," *Information Sciences*, vol. 554, pp. 145–156, Apr. 2021, https://doi.org/10.1016/j.ins.2020.12.037.

[18] F. J. Farsana, V. R. Devi, and K. Gopakumar, "An audio encryption scheme based on Fast Walsh Hadamard Transform and mixed chaotic keystreams," *Applied Computing and Informatics*, vol. 19, no. 3–4, pp. 239–264, Aug. 2020, https://doi.org/10.1016/j.aci.2019.10.001.

[19] M. Gabr *et al.*, "Application of DNA Coding, the Lorenz Differential Equations and a Variation of the Logistic Map in a Multi-Stage Cryptosystem," *Symmetry*, vol. 14, no. 12, Dec. 2022, Art. no. 2559, https://doi.org/10.3390/sym14122559.