Journal of
Engineering Sciences
Faculty of Engineering
Assiut University
JES
مجلة
العلوم الهندسية
كلية الهندسة
جامعة أسيوط
journal homepage: http://jesaun.journals.ekb.eg

Enhanced Hill Cipher Encryption Using Chaotic Logistic Maps for Improved Security and Key Randomness

Received 18 March 2025; Revised 17 June 2025; Accepted 17 June 2025

Doaa Ahmed Khalaf[1]
Khaled F. Hussein[2]
Mohamed M. Darwish[3]

Keywords
Hill Cipher, Logistic Maps, Cryptography, Chaotic Systems, Encryption, Data Security.

Abstract: In cryptography, maintaining data confidentiality and ensuring resilience against various attacks are of utmost significance. A popular classical encryption method that is well-known for its effectiveness and simplicity in protecting text data is the Hill cipher. The Hill Cipher algorithm is improved as a polygraphic substitution cipher based on linear algebra. This algorithm uses a square matrix key, and its key matrix must be invertible. However, its susceptibility to known-plaintext and chosen-plaintext attacks, along with key matrix constraints, limits its effectiveness. This paper presents an enhanced Hill cipher algorithm that integrates chaotic logistic maps to improve security, key randomness, and resistance to cryptanalytic attacks. Using the unpredictable nature of chaotic sequences, a dynamic key matrix is generated, ensuring stronger diffusion and confusion properties in encryption. The proposed method eliminates weaknesses associated with traditional hill cipher and enhances resistance to statistical and brute-force attacks. The results of experiments and analysis of security show that the chaotic logistic Hill cipher significantly improves the encryption strength. This approach provides a robust and adaptable cryptographic solution for securing textual data in modern applications.

1. Introduction

Today, considering the emergence of digital globalization, data security, and privacy have become significant challenges for the global research community. Given how quickly communication technologies and the Internet are developing, safeguarding transmitted information against potential attacks has become a critical and urgent priority [1]. Text encryption is vital in today's digital landscape, ensuring secure communication and protecting sensitive data. With the increasing prevalence of IoT, cloud computing, and digital communication, the risk of unauthorized access to sensitive information, such as financial transactions, personal data, and confidential business details, have grown significantly.

[1] Doaaahmed.11@gmail.com Independent researcher, Graduate from Faculty of Computers and Information, Computer Science Department – Assiut University, Egypt.

[2] khussain@aun.edu.eg - Vice Dean of Students' Affairs – Faculty of Computers and Information – Assiut University.

[3] Mohamed_darwish@aun.edu.eg - Assistant Professor – Computer Science Department – Faculty of Computers and Information Assiut University.

https://doi.org/ 10.21608/jesaun.2025.434995

Researchers have created techniques, including watermarking, steganography, cryptography, and others, to handle this security problem and related issues [2, 3, 4]. Among them, an efficient approach to protecting sensitive information is the implementation of cryptography. Cryptography is a field of science dedicated to securing and protecting information during transmission. It encompasses the art and science of message transformation to guarantee security and resilience against unauthorized access or attacks [5, 6]. In cryptography, images and text are encrypted before being transmitted over the network. Cryptography consists of two essential steps: encryption and decryption. Through encryption, readable, plain text data is changed into ciphertext, an encoded format that only authorized users can access. Decryption, using a specific key, converts the encoded data back to its original form. This mechanism ensures private data is kept safe and shielded from unwanted access [7, 8]. Over time, cryptography has been thoroughly studied and extensively utilized across diverse domains to ensure data security, including artificial intelligence [9], data transmission [10, 11], information confrontation [12], and image encryption [13-16]. Matrix theory is a crucial mathematical technique that has been effectively used in cryptography and is frequently used in algorithms of cryptographic, such as the Playfair cipher [17, 18], the Knapsack cipher [19, 20], the Chaos cipher [21, 22], the Hill cipher [23, 24]. Plaintext, encryption, decryption, key, and ciphertext are their primary components.

The Hill Cipher, introduced by Lester S. Hill, is a common example of classical symmetric encryption algorithms, and its fundamental concept is the use of linear combinations in matrices [25, 26]. Because of its affordability, dependability, and simplicity of use, the Hill Cipher is frequently employed in data security, particularly for encrypting network transmission data [27, 28]. At the moment, image encryption has made good use of Hill Cipher technology [29, 30, 31, 32] and secure fiber optic communication systems [33]. Thus, the study of the Hill Cipher is very important for data security. This method of encryption, the Hill cipher, encrypts and decrypts data using a square matrix as the key. To decrypt the Hill Cipher, the inverse of the key matrix is needed. However, not all matrices are invertible, making them unsuitable for use as key matrices. The encrypted text cannot be deciphered if the key matrix is not invertible [34]. The Hill Cipher is resistant to statistical and brute-force attacks; however, its linear nature structure and static key design render it susceptible to known ciphertext and plaintext attacks [35]. Despite its simplicity, these

limitations compromise its security against such known plaintext attacks. Chaotic systems have gained attention in cryptography for their randomness, unpredictability, and sensitivity to initial conditions [36]. With properties like sensitivity to initial conditions, pseudo-random behavior, and wide key space, chaotic maps are well-suited for dynamic key generation and data scrambling, making them highly effective in cryptographic applications [37- 40]. Among these maps, Logistic maps, in particular, have gained attention due to their simplicity and effectiveness. However, there is limited work on leveraging chaotic systems to enhance text encryption as well as with Hill cipher specifically, as most research in cryptography focuses on image encryption. However, the Hill cipher’s linear matrix transformation poses a major drawback, as it fails to completely obscure image characteristics, especially in images with strong correlations between adjacent pixels. Additionally, both the original Hill cipher and its variants rely on an invertible key matrix for decryption, which may not always exist, creating a potential issue. The uniformity of color intensities and high redundancy further contribute to this limitation, making the Hill cipher unsuitable for image encryption.

To improve the Hill cipher's robustness and security, this paper introduces an improved algorithm that integrates the chaotic properties of the logistic map—including high sensitivity, unpredictability, and an extensive key space—into its algebraic foundation. The proposed Improved Hill Cipher (IHC) dynamically generates key matrices for each plaintext block, using the logistic map to control the en- crypton process. By incorporating this chaotic element, the encryption becomes more complex and robust, significantly improving resistance to attacks and creating a secure hybrid encryption scheme. The following are this paper's primary contributions:

1. Combining the Logistic Chaotic Map with the classical Hill Cipher improves the encryption frame- work's security and resistance to cryptanalysis.
2. The Logistic Map generates dynamic key matrices for each plaintext block, ensuring high sensitivity to initial conditions and mitigating vulnerabilities associated with static keys.
3. The incorporation of chaotic maps strengthens the Hill Cipher, making it more resilient against known plaintext and pattern-based attacks.
4. The paper evaluates the security of the Improved Hill Cipher through various analyses, illustrating its advantages over the traditional Hill Cipher.

The rest of the paper is structured as follows: Section 2 provides an overview of the relevant works. Section 3 presents the essential preliminary material addressed in this work. Section 4 delineates the suggested encryption methodology. Section 5 discusses the analysis of security and experimental results for the proposed algorithm in this paper. Section 6 includes the conclusion.

2. Related Work

This section examines the current literature on the Hill Cipher and its enhancements. It has been extensively studied in the last few years. For instance, Acharya suggested a technique for using an involutory matrix to create a self-invertible matrix [41]. It effectively satisfies the requirement that an integer self-invertible matrix's inverse matrix stays integral, because of the characteristic that if matrix A is self-inverse, then $A = A^{-1}$. Rahman et al. [42] introduced an enhanced Hill cipher variant (Hill++). It further acts as an encryption key by producing a random matrix key depending on earlier blocks. This approach prevents vulnerabilities like all-zero plaintext blocks. By integrating the Hill cipher with the affine cipher, their method significantly strengthens resistance against attacks. Agrawal and Gera [43] proposed an encryption method that first applies the Hill cipher algorithm to generate ciphertext in numerical form. These numerical values are then mapped to points on an Elliptic Curve Cryptography (ECC) system using scalar multiplication. While this approach enhances security, it also increases computational complexity, as scalar multiplication is time-intensive. Considering that complicated problems can be solved by evolutionary algorithms, Agarwal [44] was the first to combine genetic algorithms with Hill cipher to expedite the Hill key matrix search. Sharma and Chirgaiya [45] introduced a solution to address the decryption of the Hill cipher issue when the key matrix is non-invertible. They proposed an offset adjustment approach, where an offset value of 1 is applied if the matrix determinant is zero, and -1 is used if the determinant is negative. This method ensures that the matrix remains invertible for decryption.

Siahaan [46] utilized a genetic algorithm to generate an unimodular matrix as the key for encryption. The capacity of this method to generate numerous key matrices at once is a significant benefit, enhancing flexibility and security in encryption. Khan [47] proposed a method for key generation by deriving a matrix that is orthogonal to a given plane to handle cases where the key matrix contains fractions. Chen et al. [48] developed the Random Key Matrix Generation Method, an interesting approach that generates Hill key matrices of high order at random using the modular multiplicative inverse of a triangular matrix. They show that RKMGM extends key matrix selection from finite fields to rational number fields with no restrictions on matrix order. In [49], Acharya proposed a key matrix generation technique based on involution, enumeration, and self-iteration to produce separate keys for distinct encryption blocks. This greatly improved resistance against a variety of attacks. SHC [50] employs a dynamic key matrix to thwart known plaintext-ciphertext attacks (KPCA) by randomly permuting the rows and columns of the master key matrix. HCM-H [51] furthermore uses a dynamic key matrix created by applying a one-way hash function on an integer that the sender chooses at random.

The authors of [52] presented an improved Hill cipher method, known as HC-PRE, which employs pseudo-random eigenvalues to generate dynamically changing key matrices. Essaid et al. [53] introduced a chaotic image encryption scheme known as VHC-CIES, which is based on a modified version of the Hill cipher. This approach enhances security by incorporating a Hill cipher variant along with three improved one-dimensional chaotic maps. A.V.N. Krishna and K. Madhuravani [54] proposed a modified Hill cipher that incorporates a randomized approach. In this method, the Hill cipher output is randomized to provide numerous ciphertext variations for the same plaintext once the plaintext is split up into blocks of equal size. However, this technique remains susceptible to known plaintext attacks. P.N. Lone and D. Singh [55] suggested using the H´enon map in conjunction with the affine hill cipher to transmit RGB images securely. Lone et al. [56] presented a brand-new RGB image encryption method that combines Affine Hill cryptography and chaos theory. Their work primarily explores advancements and enhancements in the Hill cipher technique. Hasoun et al. [57] successfully incorporated the asymmetric cryptographic algorithm into the Hill cryptographic algorithm, enhancing its security and effectiveness. Jin et al. [58] introduced a novel time-varying key dynamic Hill cipher (DHC) technique. This approach replaces the static matrix key of the traditional Hill cipher (THC) with a time-varying matrix key, aiming to augment the security of the conventional Hill cipher. To increase the classic Hill cipher's (THC) security and expand its application in medical image encryption, Xi et al. [59] suggested the Arnold scrambling approach for a new dynamic Hill cipher (DHCAST). Unlike the THC, the DHCAST utilizes a time- varying matrix as its secret key, significantly enhancing its security. The proposed DHCAST is effectively applied to encrypt medical images.

3. Preliminaries Relevant Knowledge

This section provides the background information needed to comprehend this paper, including the Classical Hill Cipher and Logistic map.

3.1 Classical Hill Cipher

Lester Hill, a mathematician, developed the symmetric block cipher technique known as the Hill cipher in 1929. The key matrix used for ciphering and deciphering should be shared and used by both the sender and the recipient. Given that evolutionary algorithms are capable of solving complex issues, the Hill cipher encrypts plaintext by dividing it into fixed-size blocks and transforming each block into ciphertext using a matrix multiplication approach. The core concept of the Hill Cipher involves using a key matrix to encrypt and decrypt messages. The key matrix must be invertible modulo 26 (for the English alphabet) to allow decryption. Here are the key elements of the Hill Cipher:

Key Matrix: The encryption key is a square matrix K of size m × m, where m is the block size (i.e., the number of letters per block). This technique fundamentally involves assigning a numerical value to each letter; the entries in the matrix are integers corresponding to the letters in the alphabet. For instance, a = 0, b = 1, . . ., z = 25. Subsequently, the plaintext (message) is partitioned into blocks of uniform size m, determined by the dimensions of the key matrix m × m. If the block size is two (P2×1), the key matrix ($K_{2\times2}$) must be 2 × 2 in dimensions.

Encryption: The key matrix K is multiplied by the plaintext matrix P to encrypt the message, which contains the numerical values of the plaintext letters. Multiplication is carried out modulo 26 to ensure that the result stays within the range of the alphabet. Specifically, for a plaintext block P and a key matrix K, the encryption is given by:

$$C = K \cdot P \, mod \, 26 \qquad (1)$$

Where C is the ciphertext matrix (the encrypted message), K is the key matrix, and P is the plaintext matrix.

Decryption: To decrypt the ciphertext message C, the recipient must calculate the key matrix inverse (K^{-1}), where $K \cdot K^{-1} = I$. Here, I am the identity matrix. The ciphertext matrix C is then multiplied by the inverse of the key matrix modulo 26 to recover the plaintext P (original message) using the equation:

$$P = K^{-1} \cdot C \, mod \, 26 \qquad (2)$$

Where P is the recovered plaintext (original message). After performing the matrix multiplication and modulo 26 operations, the result is the ciphertext matrix C. The resulting ciphertext is a sequence of numbers that can be converted back into letters.

3.2 Chaotic Logistic Map

The Logistic Map is a simple yet thoroughly researched one-dimensional chaotic map frequently used in chaos-based cryptography [60]. It is valued for its straightforward structure, strong chaotic behavior, unpredictability, and low cross-correlation. Representing a nonlinear dynamic system, the Logistic Map exhibits complicated, chaotic behavior and is expressed by the equation:

$$x_{n+1} = r.x_n(1 - x_n) \tag{3}$$

where the system variable at iteration n is x_n, x_0 is the chaotic map's initial state, and $x_n \in$ (0, 1) is the chaotic sequence that is produced. The control parameter r is within the range r ∈ (0, 4). The logistic map demonstrates chaotic behavior when r is in the range of [3.57, 4]. The bifurcation diagram in Fig. 1 illustrates the logistic map's chaotic range of [3.57, 4].

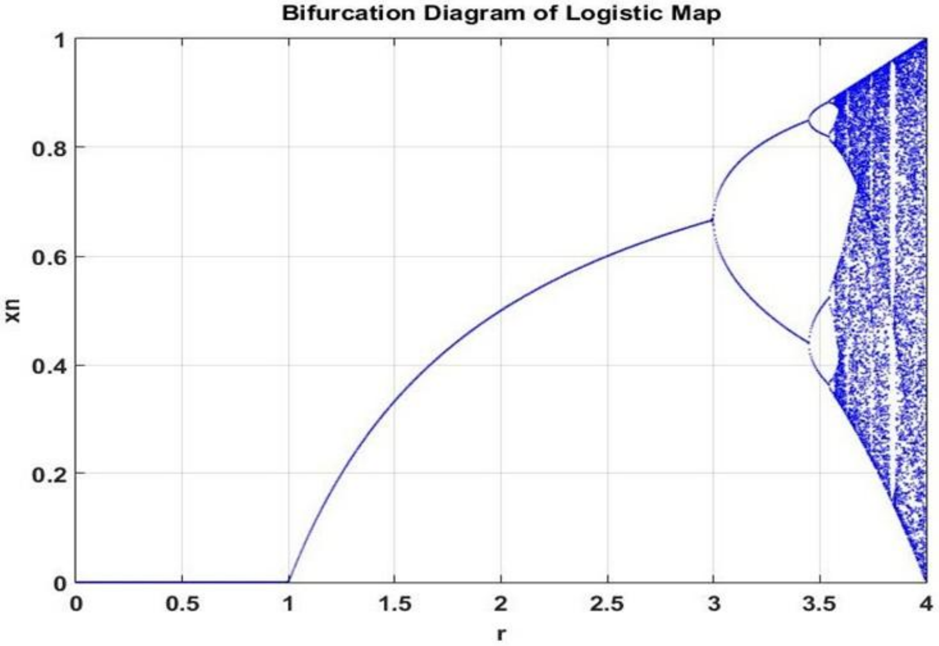

Figure 1: A classic diagram of a logistic map.

4. The Proposed Algorithm

This section introduces the proposed method that merges the chaotic properties of the Logistic Map with the block encryption mechanism of the Hill Cipher to strengthen cryptographic security. In the proposed encryption algorithm, the key matrix used in the Hill cipher is dynamically generated using the chaotic Logistic Map**,** a fundamental requirement of the Hill cipher is that the key matrix must be invertible modulo n (where n is the modulus, e.g., 26). A matrix is considered invertible under modulo arithmetic only if its determinant is both non-zero and coprime with the modulus, that is gcd(det(K), n) = 1, which is then applied to the Hill Cipher for both encryption and decryption. To ensure this condition is always met, the system includes a validation step:

1. After generating the key matrix from the Logistic Map, its determinant is calculated.

2. The system checks whether the determinant satisfies the invertibility condition modulo n.
3. if the matrix is not invertible (i.e., gcd(det(K), n) ≠1 it is discarded.
4. The Logistic Map then continues with new iterations to regenerate a new matrix.
5. This process repeats until a valid, invertible key matrix is produced

This enhancement not only boosts security but also improves the overall efficiency of the system compared to the traditional Hill Cipher method. The proposed algorithm consists of two main phases: encryption and decryption. The encryption and decryption processes can be represented visually in Fig. 2 and Fig. 3, respectively. For simplicity and clarity, Table 1 summarizes the notation used in the encryption algorithm. Both phases leverage the dynamic key matrix generated by the Logistic Map to ensure high sensitivity to initial conditions and improved cryptographic strength.

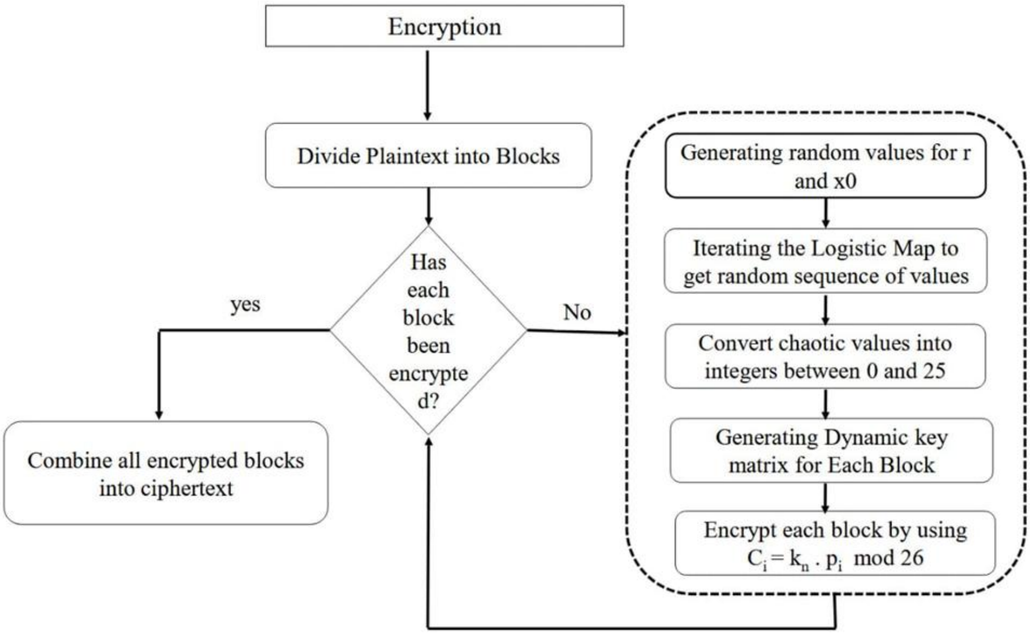
Encryption
Divide Plaintext into Blocks
Has each block been encrypted?
yes
No
Combine all encrypted blocks into ciphertext
Generating random values for r and x0
Iterating the Logistic Map to get random sequence of values
Convert chaotic values into integers between 0 and 25
Generating Dynamic key matrix for Each Block
Encrypt each block by using $C_i = k_n \cdot p_i \mod 26$

Figure 2: A diagram of the Encryption process

4.1 The Encryption Process

1. Input the Plaintext: Convert the plaintext into its numerical representation.

2. Preprocess the Plaintext: Divide the plaintext into fixed-size blocks that match the dimensions of the key matrix.

3. Key Generation:

Initialize the Logistic Map: Choose an initial value $\boldsymbol{x_0}$ (seed) and a control parameter r for the Logistic Map. Ensure x0 is in the range (0, 1) and r is within the chaotic range (e.g., $\mathbf{3.57 \leq r \leq 4}$)

Generate the Dynamic Key Matrix: Iterate the Logistic Map to obtain a sequence of chaotic numbers. Scale and discretize the chaotic values to generate the elements of an invertible key matrix. For example:

$$K = \lfloor x_i \times 100 \rfloor \, mod \, n, \qquad (4)$$

Where n is the modular base (e.g., 26 for alphabets), and ⌊ D ⌋ uses the floor function to find the largest integer that is less than or equal to D, where D is a real number.

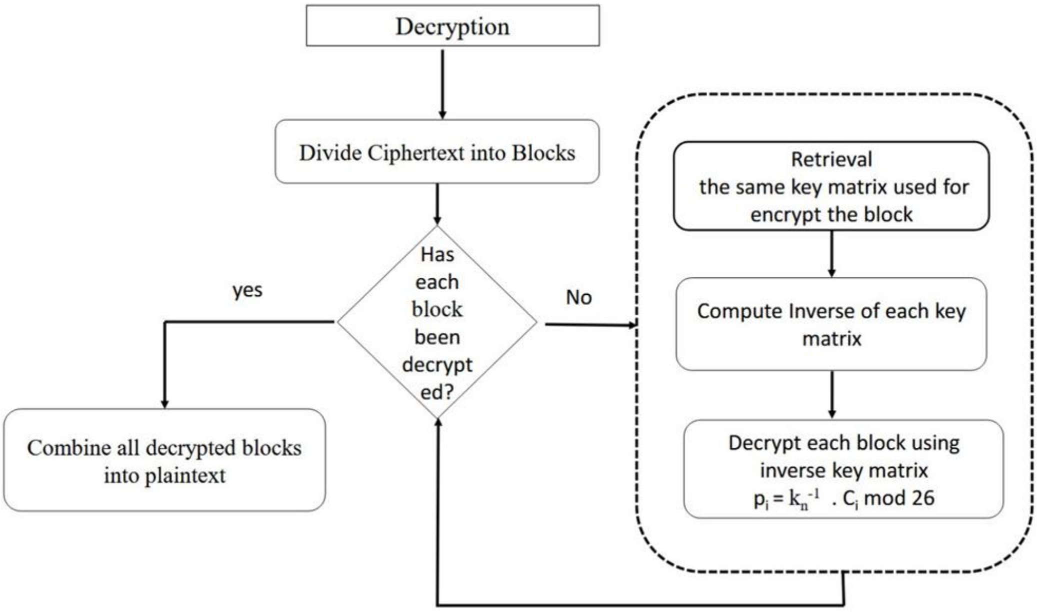
Decryption
Divide Ciphertext into Blocks
Has each block been decrypted?
yes
No
Retrieval the same key matrix used for encrypt the block
Compute Inverse of each key matrix
Decrypt each block using inverse key matrix
$p_i = k_n^{-1} \cdot C_i \bmod 26$
Combine all decrypted blocks into plaintext

Figure 3: A diagram of the Decryption process

Table 1: Notation for encryption Algorithm

Symbol	Description	Notes
x0	Initial value for Logistic Map	0< x0 <1
r	Control parameter for Logistic Map	3.57 ≤ r ≤ 4
x_n	Logistic sequence value at step n	Used to generate matrix elements
K	Key matrix used in Hill Cipher	Must be invertible modulo 26
k^{-1}	The inverse Key Matrix	Used to decrypt ciphertext
P	Plaintext vector	Divided into blocks
C	Ciphertext vector	Encrypted output
det(k)	Determinant of key matrix K	Used to check invertibility

4. **Encrypt Each Block**: Multiply each plaintext block with the dynamic key matrix:

$$C = (K \cdot P) \, mod \, n, \qquad (5)$$

where *P* is the plaintext block, *K* is the key matrix, *C* is the resulting ciphertext block, and *n* = 26 for alphabets.

5. **Generate the Ciphertext**: Combine all ciphertext blocks and convert them back to characters.

4.2 The Decryption Process

1. **Input the Ciphertext**: Convert the received ciphertext into its numerical representation.
2. **Reinitialize the Logistic Map**: Use the same initial value x_0 and control parameter r as in the encryption process to regenerate the dynamic key matrix.
3. **Generate the Dynamic Key Matrix**: Repeat the key matrix generation process using the Logistic Map to ensure synchronization with the encryption phase.
4. **Calculate the Inverse Key Matrix**: Compute the modular inverse of the key matrix K to be used for decryption:

$$K^{-1} \cdot K \equiv I \, mod \, n, \tag{6}$$

where I am the identity matrix.

5. **Decrypt Each Block**: Multiply each ciphertext block with the inverse key matrix:

$$P = (C \cdot K^{-1}) \, mod \, n \tag{7}$$

Where P is the plaintext block, C is the ciphertext block, and $K^{-1}$is the inverse key matrix.

6. **Reconstruct the Plaintext:** Combine all decrypted blocks and convert them back to characters.

4.3 Illustrative Example

Let the plaintext be**:" HOPE".**

Step 1: Substitute Each Letter with Its Corresponding Number

- H = 7, O = 14, P = 15, E = 4.
- Plaintext block (P1): [7, 14].
- Plaintext block (P2): [15, 4].

Step 2: Initialize the Logistic Map and Generate the Dynamic Key Matrix

➢ **To Encrypt the First Block (P1):**

- Initial value (x_0): 0.7820
- Control parameter (r): 3.5029
- Generate 4 chaotic numbers by iterating the Logistic Map using Equation (3):

$$x_0 = 0.7820,$$
$$x_1 = 3.5029 \cdot 0.7820 \cdot (1 - 0.7820) = 0.5971,$$
$$x_2 = 3.5029 \cdot 0.5971 \cdot (1 - 0.5971) = 0.8427,$$
$$x_3 = 3.5029 \cdot 0.8427 \cdot (1 - 0.8427) = 0.4643.$$

- Scale and discretize the chaotic values into integers between 0 and 25:

$$K_{11} = \lceil 0.7820 \cdot 25 \rceil = 19, K_{12} = \lceil 0.8427 \cdot 25 \rceil = 21,$$
$$K_{21} = \lceil 0.5971 \cdot 25 \rceil = 14, K_{22} = \lceil 0.4643 \cdot 25 \rceil = 11.$$

- Dynamic key matrix (K_1):

$$K_1 = \begin{pmatrix} 19 & 21 \\ 14 & 11 \end{pmatrix}$$

➢ **To Encrypt the Second Block (P2):**

- Initial value (x_0): 0.7735
- Control parameter (r): 3.9853
- Following the same steps as above, the dynamic key matrix (K_2) is:

$$K_2 = \begin{pmatrix} 19 & 20 \\ 17 & 13 \end{pmatrix}$$

Step 3: Encryption and Generate the Ciphertext

The encryption formula is:

$$C = (K \cdot P) \, mod \, 26.$$

➢ Encrypting **the First Block (P1):**

- Substitute values:

$$P_1 = \begin{bmatrix} 7 & 14 \end{bmatrix} \, , \, K_1 = \begin{pmatrix} 19 & 21 \\ 14 & 11 \end{pmatrix}$$

- Perform matrix multiplication:

$$C_1 = \, . \begin{bmatrix} 7 & 14 \end{bmatrix} \; mod \, 26 = \begin{bmatrix} 427 & 252 \end{bmatrix} \; mod \, 26 = \begin{bmatrix} 11 & 18 \end{bmatrix}$$

- Ciphertext vector (C_1): [11, 18].
- Convert numbers back to letters: 11 = L, 18 = S.
- Ciphertext:” **LS”**.

➢ **Encrypting the Second Block (P2):**

- Substitute values:

$$P_{2\,=}\begin{bmatrix} 15 & 4 \end{bmatrix} \, , \, K_2 \; = \begin{pmatrix} 19 & 20 \\ 17 & 13 \end{pmatrix}$$

- **Perform matrix multiplication and modular reduction:**

$$C_2 = \, . \begin{bmatrix} 15 & 4 \end{bmatrix} \; mod \, 26 \; = \begin{bmatrix} 1 & 21 \end{bmatrix}$$

- **Ciphertext vector (C_2): [1, 21].**
- Convert numbers back to letters: 1 = B, 21 = V.
- Ciphertext:**” BV”**.

Full Encrypted Ciphertext:” LSBV”.

Step 4: Decryption and Generate the Plaintext

To decrypt the ciphertext, you must use the same matrices that were used for encryption, and calculate its modular inverse. The modular inverse K^{-1} satisfies:

$$K^{-1} \cdot K \equiv I \, mod \, 26,$$

where *I* am the identity matrix. Using matrix algebra and modular arithmetic.

➢ **Decrypt the first ciphertext block** C_1 **using the decryption formula:**

$$P_1 = (C1 \cdot K_1^{-1}) \, mod \, 26$$

- The inverse of K1 is:

$K_1^{-1} = [17 \quad -23 \quad -24 \quad 1\ ]\ mod\ 26 = [17 \quad 3 \quad 2 \quad 1\ ]$

- Substitute values:

  $C_1 = [11 \quad 18]\ ,\ K_1^{-1} = [17 \quad 3 \quad 2 \quad 1\ ]$

- Perform the matrix multiplication:

  $P_1 = .[17 \quad 3 \quad 2 \quad 1\ ]$ mod 26 = $[241 \quad 40\ ]$ mod 26 = $[7 \quad 14\ ]$
- Convert the numbers back to letters: 7 = *H,* 14 = *O*.
- Recovered plaintext block:” **HO”**.

➢ **Applying the same steps to decrypt the second ciphertext block** ***C*₂:**

To decrypt c₂, use the same key matrix K₂ = $[\mathbf{19} \quad \mathbf{20} \quad \mathbf{17} \quad \mathbf{13}\ ]$

- The inverse of *K*₂ is:

  $K_2^{-1} = [13 \quad -16 \quad -11 \quad 23\ ]\ mod\ 26 = [13 \quad 10 \quad 15 \quad 23\ ]$

- Substitute the values

  $C_2 = [1 \quad 21\ ]\ ,\ K_2^{-1} = [13 \quad 10 \quad 15 \quad 23\ ]$

- Perform the matrix multiplication:

  $P_2 = [1 \quad 21\ ].[13 \quad 10 \quad 15 \quad 23\ ]$ mod 26 = $[15 \quad 4\ ]$

- Convert the numbers back to letters: 15 = *P,* 4 = *E*.
- Recovered plaintext block:” **PE”**

Full plaintext:” HOPE". Experimental Analysis

5. Experimental Analysis

We tested the suggested algorithm to see how well it worked through experiments analyzing its security and resistance to cryptographic attacks. The encryption and decryption tests were conducted on a laptop with an Intel i5-1065G7 processor (1.30 GHz) and 8 GB RAM, using a MATLAB (R2016a) implementation of the algorithm.

5.1 Key Space Analysis

The total number of keys that can be used for encryption and decryption is defined by a cryptographic system’s key space. A larger key space makes the cipher more resistant to brute-force attacks. Brute- force assaults can be prevented more successfully with a bigger key space. Determining the size of the key space is essential for assessing the resistance to brute-force attacks. For the Improved Hill Cipher with Logistic Map, the size of the key space is mainly determined by factors such as the seed value (x_0) and the control parameter (r) of the logistic map. Both require high precision and are represented as real numbers. Matrix size $n \times n$ is another key factor, where increasing n exponentially increases the key space.

Assuming that each parameter is represented with double precision up to 16 decimal places, this results in 10^{16} possible values for each parameter, yielding a key space size of $10^{16} \times 10^{16} = 10^{32}$. With matrix size $n \times n$, the total key space of the cryptographic algorithm becomes $n^2 \times 10^{32}$, which far exceeds the suggested 2^{100}. Therefore, the key space of this approach is sufficiently large, and the suggested technique can successfully fend off brute-force attacks. In the traditional Hill cipher, the key is a static matrix of size $n \times n$ with entries in modulo 26 (assuming alphabetic encryption), and the matrix must be invertible. This constraint significantly reduces the total number of valid keys, resulting in a limited and countable key space. Table 2 presents a comparative analysis of the key space between the proposed algorithm and related existing algorithms.

Table 2: Comparative Analysis of Key Space.

Criteria	Traditional Hill Cipher	Affine-Hill + Chaos	Proposed Algorithm
Key Type	Static (fixed matrix used for all blocks)	Adds affine transformation & chaos-based key seed, but static over the full message	Dynamic (new matrix generated for each plaintext block)
Key Source	Invertible matrix over mod 26	Hill matrix + chaotic seed & affine c	Derived from chaotic logistic map values (x_0, r)
Key Space Size	Limited and finite	~26^{n^2} × chaotic seed range × affine offset	~ [(x_0 × r × 26^{n^2})] ^B
Key Reuse	The same key is reused for all blocks	Moderate	Each block uses a unique key
Predictability	Moderate (fixed structure can be guessed with enough data)	Moderate (depends on seed)	Very low (high randomness and unpredictability due to chaos)
Security Improvement	Basic level of confusion and diffusion	Improved over classical Hill	Enhanced security due to dynamic key variation and chaotic behavior

5.2 Key Sensitivity Analysis

Secure cryptographic systems must exhibit key sensitivity, ensuring that even a small modification to the encryption key leads to significantly different ciphertexts than those produced by the original key. This characteristic protects the cipher from differential attacks, where attackers attempt to deduce key information by analyzing the variations in ciphertext resulting from minor changes to the key. To evaluate key sensitivity in the Improved Hill Cipher with Logistic Map, a plaintext message “HELLOO” was encrypted using two keys that differed only slightly in the seed value (x0) of the logistic map. For this purpose, x_0 in the initial conditions was modified to $x_0 + 1.0 \times 10^{-12}$. The control parameter (r) was kept constant, while the seed values were set as follows:

Experiment 1:

- Key 1: x_0 = 0.34500000000000, r = 3.5025
- Slight key modification: x_0 = 0.345000000000001, r = 3.5025

Both keys were used to encrypt the plaintext, and the corresponding ciphertexts were compared. The Hamming distance is the number of positions where the corresponding characters differ between two ciphertexts of the same length. It was calculated to measure the sensitivity to the key change as follows:

$$d_H(C1, C2) = |\{i \in \{1,2,3, \dots\dots n\}: a_i \neq b_i\}|, \qquad (8)$$

Where $C_1 = a_1a_2a_3 \ldots a_n$ and $C_2 = b_1b_2b_3 \ldots b_n$ are the ciphertexts formed by the original key and the slightly modified key, respectively.

The results of the two ciphertexts generated using the slightly different keys were:

- C_1:” TXFLEO”
- C_2:” AEQWSC”

The Hamming distance, d_H (C_1, C_2), between these two ciphertexts was equal to 6.

Experiment 2:

- Key 1: x_0 = 0.4994000000000000, r = 3.7074
- Slight key modification: x_0 = 0.4994000000000001, r = 3.7074

Encrypt the same plaintext using two slightly different keys and then compare the results.

- Plaintext: “ENCRYPTION”
- C_1:” YTGDYXYWGV”
- C_2:” YIIMMACPGC”

The Hamming distance, dH (C1, C2), between these two ciphertexts was equal to 8.

Since the original plaintext cannot be deciphered and there are significant variations in the cipher- texts as a result of modest key changes, it is evident that the algorithm is extremely sensitive to these modifications. This indicates that all characters in the ciphertext changed when the key was altered slightly. The key sensitivity analysis confirms that the Improved Hill Cipher with Logistic Map is highly sensitive to small changes in the key. This ensures that even minimal variations in the key produce mostly distinct ciphertexts, significantly enhancing the cipher’s resistance to differential cryptanalysis and strengthening its overall security.

Experiment 3:

We conducted a numerical experiment to compare the proposed algorithm with existing algorithms in term of key sensitivity analysis. The evaluation included a numerical example using the Hamming distance, and the results are presented in Table 3.

- Plaintext: “ENCRYPTION”
- Key1 = [3, 3; 1, 5].
- Slight key modification = [3, 3; 1, 6].

Table 3: Comparison of key sensitivity.

Algorithm	Plaintext	C1	C2	Hamming distance
Traditional Hill Cipher	ENCRYPTION	ZRFJNVDHDB	ZEFANKDPDO	5
Affine-Hill + Chaos		YEOMXRHKLN	YRODXGHSLA	5
Proposed		YTGDYXYWGV	YIIMMACPGC	8

These results confirm that the proposed offers the highest resistance to differential and key-related attacks, making it the most secure among the compared algorithms. The high Hamming distance validates its robustness in practical cryptographic applications. The Traditional Hill Cipher showed limited sensitivity to key changes, while the Affine Hill Cipher with Chaos improved sensitivity through chaotic perturbation. The Improved Hill Cipher with Logistic Map exhibited full sensitivity, producing entirely different ciphertext from minor key changes, demonstrating strong resistance to key-based attacks.

5.3 Entropy Analysis

Entropy is a crucial parameter for assessing a cryptographic system's security since it quantifies the ciphertext's randomness or unpredictability. Higher entropy signifies stronger security, as it indicates that the ciphertext conceals any patterns or traces of the plaintext. Greater randomness and unpredictability in the ciphertext are reflected in a high entropy level, which makes it extremely resilient to statistical analysis and cryptographic attacks.

The entropy of the ciphertext was computed using the formula:

$$H(s) = -\sum_{i=1}^{N} P(s_i) log_2 P(s_i) \tag{9}$$

Where S is the set of characters in the ciphertext, N denotes the total number of unique characters, and $P(s_i)$ represents the probability of occurrence of s_i in the ciphertext.

The ideal information entropy value for a 26-character alphabet is $\log_2$ (26) = 4.7004 ≈ 4.7. This represents the maximum entropy for a 26-character alphabet, indicating optimal unpredictability and security when each character has an equal probability of occurrence. If the entropy is lower, it suggests patterns or biases in the data, which could weaken security. The higher the information entropy, the closer it is to the ideal value. To evaluate the entropy of ciphertexts, we experimented by generating random plaintexts of varying sizes (e.g., 50, 100, 500, 1000, 5000, and 10000 characters) and encrypting them using the proposed algorithm. The entropy of the resulting ciphertexts was then calculated using the entropy formula. The results of the information entropy calculations are recorded in Table 4. As seen in Table 4, the computed entropy of the ciphertexts increases with the size of the plaintext, approaching the ideal information entropy value of 4.7. This indicates that the algorithm exhibits good randomness and security.

Additionally, Table 5 gives data in comparison with other algorithms. Our designed algorithm has the highest average value of information entropy in ciphertexts. Therefore, the algorithm designed in this paper has good encryption and can effectively resist information entropy attacks.

Table 4: Information entropy of ciphertexts of varying sizes.

Size (characters)	Ciphertext Entropy
50	4.3329
100	4.5501
500	4.6633
1000	4.6883
5000	4.6919
10000	4.6928

Table 5: Comparative analysis of information entropy.

Size (characters)	Ciphertext Entropy		
	Traditional Hill Cipher	Affine-Hill + Chaos	Proposed
50	3.8522	3.7345	4.3329
100	4.0431	4.2784	4.5501
500	4.5930	4.6094	4.6633
1000	4.6479	4.6510	4.6883
5000	4.6850	4.6904	4.6919
10000	4.6918	4.6915	4.6928

The entropy values of ciphertexts with varying sizes are illustrated in Fig. 4. Generally, entropy tends to increase with the size of the ciphertext, as larger ciphertexts display higher levels of randomness. For sufficiently large ciphertexts, the entropy stabilizes, indicating that a high level of randomness and unpredictability is routinely achieved by the encryption technique, regardless of the ciphertext size. The graph typically reveals a rising trend for smaller ciphertext sizes, followed by a plateau as the entropy reaches its peak, highlighting the algorithm's effectiveness in generating highly random and secure ciphertexts.

5.4 Correlation Analysis

In cryptographic systems, the correlation between plaintext and ciphertext is a pivotal element in assessing the security and resilience of the encryption technique. A strong encryption algorithm should produce ciphertext that is statistically independent of the plaintext, ensuring that no information about the plaintext can be inferred from the ciphertext. Correlation analysis is a widely used method to evaluate this relationship.

This study evaluates the effectiveness of the Improved Hill Cipher (IHC), enhanced with a Logistic Map, by analyzing the correlation between plaintext and ciphertext. Randomly generated plaintexts of varying sizes were used to assess the cipher's ability to disrupt statistical relationships and ensure robust encryption. The degree of correlation between plaintext and ciphertext is estimated by the correlation coefficient, where correlation values close to 0 indicate minimal similarity and no linear correlation, demonstrating effective encryption. The correlation coefficient is determined as follows:

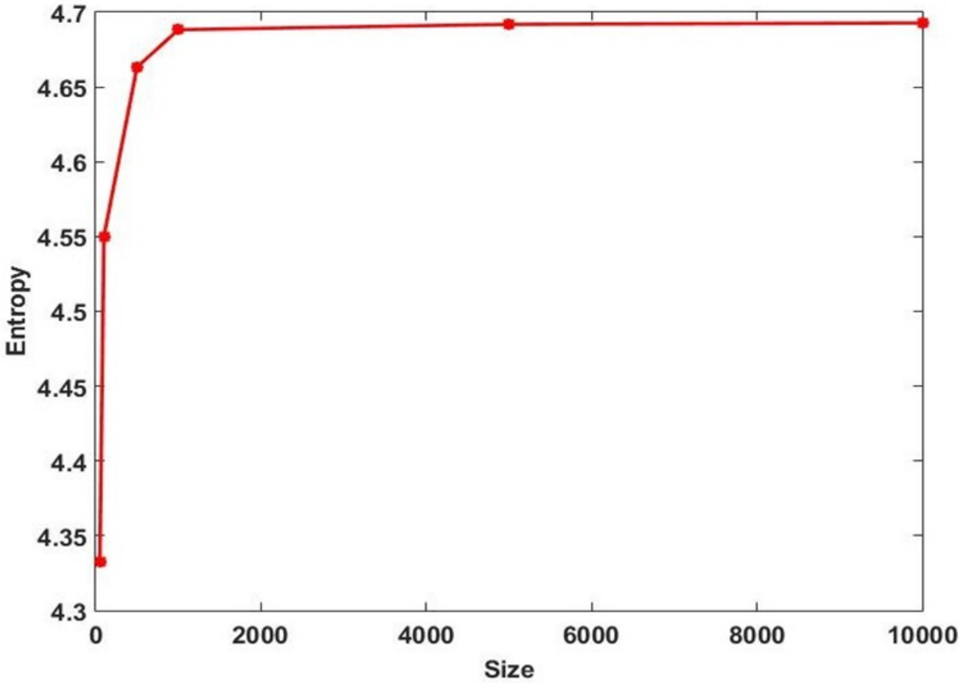

Figure 4: Information entropy of ciphertexts of varying sizes

$$r = \frac{\sum_{i=1}^{n} \quad (P_i - \mu_P)(C_i - \mu_C),}{\sqrt{\sum_{i=1}^{n} \quad (P_i - \mu_P)^2 \sum_{i=1}^{n} \quad (C_i - \mu_C)^2}},$$

$$\mu_P = \frac{1}{n}\sum_{i=1}^{n} \quad P_i, \qquad (10)$$

$$\mu_C = \frac{1}{n}\sum_{i=1}^{n} \quad C_i,$$

Where n denotes the total number of elements in plaintext and ciphertext, Pi and Ci are the individual values of plaintext and ciphertext at position i, respectively, and µP and µC are the mean (average) values of the plaintext and ciphertext sequences, respectively. This analysis was conducted on randomly generated plaintexts of varying sizes to comprehensively assess the encryption scheme's performance. In this study, plaintexts of different lengths (e.g., 50, 100, 500, 1000, 5000, and 10000 characters) were randomly generated. The plaintext values ranged from 0 to 25, corresponding to alphabetic representations, and the corresponding ciphertexts were obtained by applying the improved Hill Cipher algorithm.

The correlation coefficients were calculated and analyzed for different plaintext sizes to evaluate how effectively the link between plaintext and ciphertext is broken by the encryption algorithm. By plotting these coefficients, trends are observed, highlighting the algorithm's ability to minimize correlations. The results of this analysis are depicted in Figure 5. As plaintext size increases, the correlation coefficient decreases toward zero, indicating minimal linear dependence. Across all tested sizes, the near-zero coefficients confirm the ciphertext's statistical independence from the plaintext. This trend suggests that the encryption algorithm performs better with larger plaintext sizes, as the chaotic key matrix has more data to randomize. In addition, experiments were conducted by encrypting randomly generated plaintexts of varying lengths (e.g., 50, 100, 500, 1000, 5000, and 10000 characters) using both the proposed method and existing algorithms, each applied 10 times. Table 6 presents the average correlation coefficient values from these experiments, demonstrating that the results of the proposed method are competitive with those of previous approaches.

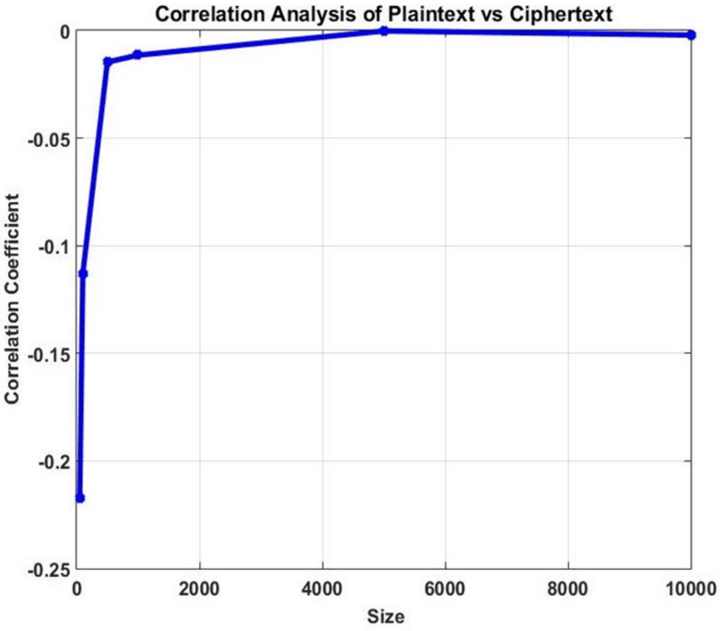

Figure 5: Correlation coefficient between plaintext and ciphertext of varying sizes.

Table 6: Comparison of correlation coefficients.

Size (characters)	Correlation Coefficient		
	Traditional Hill Cipher	Affine-Hill + Chaos	Proposed
50	-0.3421	0.3852	0.0626
100	0.0792	-0.1009	-0.0795
500	0.1620	0.0153	0.0291
1000	-0.0120	0.0326	-0.0051
5000	-0.0194	0.0480	0.0065
10000	0.0258	0.0044	0.0026

The correlation analysis results reveal consistently low coefficients across all plaintext sizes, confirming the method's effectiveness in disrupting statistical relationships between plaintext and ciphertext. This demonstrates the algorithm's ability to introduce randomness, obscure plaintext patterns, and ensure ciphertext independence, making it highly resistant to pattern-based and statistical attacks. These findings affirm the practical applicability of the proposed scheme for securing sensitive data against advanced cryptographic threats.

5.5 Chosen-plaintext attack (CPA)

An experimental strategy was used to evaluate the suggested Enhanced Hill Cipher's resistance against Chosen Plaintext Attacks (CPA). An adversary can encrypt specific plaintexts in a CPA and examine the ciphertexts to try to identify trends or information about keywords. This study's encryption approach creates a unique key matrix for every plaintext block dynamically using a chaotic logistic map. This technique seeks to improve cryptographic security by removing ciphertext repetition.

- ➢ **Experimental Setup:**
  - **Block Size**: 3 characters
  - **Plaintext**: " **BBBBBBCCCCCCABCABC** "

Table 7 shows the ciphertexts obtained by encrypting three identical plaintext blocks using dynamically generated keys. Each block was encrypted with a unique key derived from a different iteration of the logistic map.

Table 7: CPA Security Analysis Using Dynamic Key Generation

BLOCK	Plaintext	Ciphertext
1	BBB	KHP
2	BBB	RLA
3	CCC	WEO
4	CCC	EMI
5	ABC	YPU
6	ABC	IRM

The resulting ciphertexts are entirely distinct even though the plaintexts are the same. This behavior validates the cipher's resistance to pattern analysis and is a direct result of the dynamic key generation technique. The Hamming distance between blocks of the ciphertext was computed in order to assess the variance further. High diffusion was indicated by the average of two out of three characters being different between any two blocks. Non-linearity and sensitivity to starting parameters are introduced via the use of chaotic key generation. This guarantees that different ciphertexts are produced by repeating plaintext blocks. Unpredictable and non-reusable ciphertext patterns Because fixed key usage is eliminated, the cipher resists CPA. These results show that the resilience of the Hill cipher to CPA is greatly increased by the dynamic key generation technique. We conducted an additional numerical experiment to compare the proposed algorithm with other existing methods. The results, summarized in Table 8, demonstrate that the improved Hill Cipher enhanced with the Logistic Map provides significantly better resistance to Chosen-Plaintext Attacks than the compared algorithms. While traditional Hill Cipher is vulnerable to chosen-plaintext attacks (CPA) because it uses a fixed key matrix identical plaintext blocks always produce the same ciphertext, revealing exploitable patterns. The Affine-Hill Cipher with a chaotic bias improves security by adding randomness, but still applies the same key across all blocks, allowing some patterns to persist. In contrast, the Hill Cipher with a dynamically generated key matrix using the Logistic Map provides strong CPA resistance by regenerating a unique key for each block, eliminating ciphertext repetition and significantly enhancing security.

Table 8: Comparative CPA analysis of the proposed Dynamic-Key Hill Cipher to other Hill-Based algorithms.

BLOCK	Plaintext	Ciphertext		
		Traditional Hill Cipher	Affine-Hill + Chaos	Proposed
1	BBBB	GHGH	ZTDZ	NOKR
2	BBBB	GHGH	TDZT	TSBF
3	BBSS	GHEW	DLAY	QWHN
4	BBSS	GHEW	QAOT	ZNVY
5	SBSB	FPFP	MDPT	NNDM
6	SBSB	FPFP	EUTN	HWAB

5.6 Ciphertext-Only Attack (COA)

The premise of a Ciphertext-Only Attack (COA) is that the attacker is only in possession of the encrypted ciphertexts and is unaware of the encryption key or the original plaintext. This kind of attack looks for ciphertext statistical patterns or repeats that could result in partial or complete decryption. The repeating of ciphertext blocks when identical plaintext blocks are encrypted with the same key is a frequent flaw in symmetric encryption techniques.

A new key matrix is created for every encryption block in the suggested encryption scheme, which improves on the Hill cipher by utilizing chaotic logistic map-based dynamic key generation. Even when the plaintext comprises identical or recurring blocks, this dynamic behavior is intended to stop ciphertexts from repeating.

➢ **Experimental Setup:**

- **Block Size**: 2 characters
- **Plaintext**: A repeated sequence "**BBBBBBBB**"

The objective of the experiment was to observe whether the same plaintext blocks would result in repeated ciphertexts or not. Table 9 shows the ciphertext output for each block of the same plaintext **"BB**".

Table 9: COA Security Analysis Using Dynamic Key Generation

BLOCK	Plaintext	Ciphertext
1	BB	ZI
2	BB	SJ
3	BB	PS
4	BB	PD

The findings clearly show that, despite the fact that the plaintext content was the same for every block ("BB"), every identical plaintext block produced a distinct ciphertext output. This illustrates how ciphertext repetition—a significant flaw in conventional Hill cipher implementations—is successfully removed by the chaotic key generation process.

The system is therefore extremely resistant to attacks of this kind since attackers using statistical analysis based on repeated ciphertexts (as in COA) would not discover any exploitable patterns.

5.7 Known-plaintext attack (KPA)

To demonstrate how dynamic key generation can effectively thwart known-plaintext attacks, we offer a numerical example utilizing an improved Hill cipher that incorporates a logistic map to produce a distinct key matrix for every plaintext block.

- **Experimental Setup:**
  - **Block Size**: 2 characters
  - **Plaintext**: "**HOPE** "
  - **Logistic map parameters:** x_0 = 0.7820 **&** r = 3.5029 for **P1**
    x_0 = 0.7735 **&** r = 3.9853 for **P2**

With the help of the above illustrative example, Table 5 summarizes the findings for the given plaintext.

Table 10: Dynamic Key Generation for each plaintext block

Plaintext	Plaintext block	Key Matrix	Ciphertext block	Ciphertext
HO	P1= [7, 14]	$K1 = \begin{pmatrix} 19 & 21 \\ 14 & 11 \end{pmatrix}$	C1= [11, 18]	**LS**
PE	P2= [15, 4]	$K2 = \begin{pmatrix} 19 & 20 \\ 17 & 13 \end{pmatrix}$	C2 = [1, 21]	**BV**

Assume the attacker knows plaintext-ciphertext pairs as:

- P1= [7, 14], C1= [11, 18],
- P2= [15, 4], C2 = [1, 21].

An attacker having access to both plaintext and ciphertext for just one block may employ linear algebra to recover its key. From $C1 = (K1 \cdot P1) \ mod \ 26$, K1 is recovered as in Table 10. This recovered key, however, would be useless for decrypting other blocks because each block step uses a different matrix based on different chaotic sequences. This behavior demonstrates that, even if portions of the message are revealed, the dynamic nature of key creation successfully avoids the entire message being revealed.

5.8 Randomness Test

A cipher's randomness is a crucial component that makes an algorithm more unpredictable, secure against attacks, and random. When creating a cryptographically secure algorithm, this characteristic must be considered. A collection of tests known as the National Institute of Standards and Technology Statistical Test Suite, or NIST-STS, is used to confirm this randomness and assess the security offered by cryptographic methods [61]. To be considered as truly random, and therefore a successful test, the computed P-value of a specific byte sequence must be greater than or equal to 0.01. The first test, the Frequency test which checks if the number of 0_s and 1_s in the binary sequence is approximately the same. The Runs Test was then used to determine whether the value oscillations between subblocks were either too fast or too Slowly, the longest consecutive subsequence in the supplied data was confirmed using the Longest Run of Ones in a Block method. To evaluate the randomness of the ciphertext generated by the proposed encryption method (Hill Cipher with Logistic Map and dynamic key generation), two different lengths of the cipher are used during the test to ensure randomness, the first one is 300 bytes long and the second one is 500 bytes long. Table 11 below shows the result of the randomness test.

Table 11: Randomness Test

Statistical Test	P-value(x)	P-value(Y)	Result
Frequency	0.54029	0.22067	pass
Runs Test	0.74896	0.67233	pass
longest Run of 1s in Blocks	0.70113	0.60525	pass

The results consistently demonstrated strong randomness across both input sizes. For the 300-byte ciphertext, all key tests — including the Monobit Test, Runs Test, and Longest Run of Ones- confirmed the encrypted output's statistical randomness produced p-values above 0.01. Similarly, for the 500-byte ciphertext, the test outcomes remained robust. All p-values exceeded 0.01, indicating no degradation in randomness as the message size increased. These results validate the proposed algorithm's resistance to statistical attacks and its ability to produce secure and unpredictable ciphertexts across different plaintext sizes.

5.9 Avalanche Effect

The Avalanche Effect measures how a small change in the plaintext (such as flipping a single character) causes significant and unpredictable changes in the ciphertext. To evaluate the Avalanche effect across different encryption algorithms, an experiment was conducted using a fixed plaintext input. A single character in the plaintext was altered, and the resulting ciphertexts from each algorithm were compared. The percentage of changed bits or characters was measured to assess each algorithm's sensitivity to minor input changes, where Table 12 represents the obtained results of the Avalanche effect experiment. This analysis was applied to the Traditional Hill Cipher, the Affine Hill Cipher with Chaos, and the Hill Cipher with a Logistic Map using a dynamic key per block.

Plaintext 1 = 'HELLOWORLD'
Plaintext 2 = 'HELLOMORLD'

These results demonstrate that the Traditional Hill Cipher provides weak diffusion due to its fixed key, while the Affine-Hill Cipher with Chaos enhances security through non-linearity but still relies on static parameters. In contrast, the Hill Cipher with a Logistic Map achieves the strongest avalanche effect by generating dynamic keys for each block, ensuring better diffusion and greater resistance to cryptanalysis.

Table 12: Comparison of Avalanche effect.

Encryption Algorithm	C1	C2	Hamming distance	Avalanche Effect
Traditional Hill Cipher	HFOZEQPMQD	HFOZAIPMQD	2	20.00%
Affine-Hill + Chaos	QERDIABDTZVO	QERTAQBDTZVO	3	25.00%
Proposed	DIRFKMXMTB	XPGZUMRZWF	9	90.00%

5.10 Ablation Study

The ablation study is crucial for evaluating the contribution of each component in our hybrid cryptographic system. This experiment quantitatively assesses the roles of the Hill cipher and the logistic map by analyzing their impact on encryption strength, we examine three configurations: (A) the classical Hill cipher with a fixed key, (B) a Hill cipher with a static key generated from a logistic map, and (C) the proposed method using block-wise dynamic keys derived from logistic map outputs with block-dependent seeds. This systematic comparison highlights the individual and combined effectiveness of each component in enhancing overall security. Experiments were conducted by encrypting two plaintexts, where the results are presented in Table 13.

Plaintext 1 = ‘LIFE’
Plaintext 2 = ‘LIVE’

Table 13: Results of ablation study.

Encryption Algorithm	Ciphertext 1	Ciphertext 2	Avalanche Effect (%)
Traditional Hill Cipher	FHBD	FHXT	50.00%
Hill + Static Logistic Key	LKNC	LKHU	50.00%
Hill Cipher + Logistic map (Dynamic Key Generation)	DHFC	XPPO	100.00%

These results of ablation study confirm that both the chaotic key component and dynamic key generation mechanism play essential roles in achieving strong cryptographic properties. The Hill cipher alone is inadequate, while even static chaos offers only limited gains. The combination, as implemented in our proposed system, ensures a high level of security, diffusion, and statistical randomness.

5.11 Runtime Test

Runtime testing (measured in seconds) is performed to evaluate how quickly an algorithm executes in this case; encryption compared to alternative methods. A lower runtime indicates faster execution. We used 100,000 randomly generated characters as plaintext, encrypted them using the proposed algorithm and other comparison algorithms, and executed each encryption process 10 times. The average encryption time was then computed and is presented in Table 14. The Classic Hill Cipher, with no additional modifications, achieved the fastest performance, with an average execution time of 0.0044 seconds.

Table 14: Average runtime encryption process.

	Classic Hill Cipher	Affine Hill Cipher with Chaos	Proposed Algorithm
Average Runtime in Seconds	0.1825	3.9039	15.0353

6. Conclusions

In this paper, we proposed an Improved Hill Cipher (IHC) that integrates the logistic chaotic map to enhance the security and efficiency of the classical Hill Cipher. By utilizing the dynamic nature of the logistic map for key generation, the IHC improves the cipher's resistance to attacks such as known- plaintext attacks, which typically exploit the linearity of the original Hill Cipher. Through the evaluation of the cipher's performance in terms of entropy, correlation analysis, and resistance to various attacks, we demonstrated that the proposed IHC offers a more robust and secure encryption method. The integration of chaotic maps introduces unpredictability and a larger key space, contributing to stronger cryptographic security while maintaining computational efficiency.

Future work could focus on further enhancing the IHC by exploring the use of other chaotic maps, such as the Arnold Cat Map or Henon Map, to compare their impact on the security and performance of the cipher. Additionally, optimizing the algorithm for different types of data (e.g., images or videos) and conducting performance tests on larger datasets could help evaluate its scalability. Moreover, implementing hybrid encryption schemes combining the IHC with other advanced encryption techniques may lead to even stronger cryptographic solutions

References

[1] J. Yin, M. Tang, J. Cao, and S. et al., "Knowledge-driven cybersecurity intelligence: Software vul- nerability coexploitation behavior discovery," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5593–5601, 2023.

[2] V. R. F. Signing, G. A. G. Teague, M. Kountchou, Z. T. Njitacke, N. Tsafack, J. D. D. Nkapkop, C. M. L. Etoundi, and J. Kengne, "A cryptosystem based on a chameleon chaotic system and dynamic dna coding," *Chaos Solitons Fractals*, vol. 155, p. 111777, 2022.

[3] F. Kahlessenane, A. Khaldi, M. R. Kafi, and S. Euschi, "A color value differentiation scheme for blind digital image watermarking," *Multimedia Tools and Applications*, vol. 80, pp. 19 827–19 844, 2021.

[4] A. Khalifa and A. Guzman, "Imperceptible image steganography using symmetry-adapted deep learning techniques," *Symmetry*, vol. 14, p. 1325, 2022.

[5] A. Abdelwahab, (2006). Cryptography scheme based on transparent feedforward neural network and ordered lookup table. JES. Journal of Engineering Sciences, 34(1), 189-197, 2006.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice (7th Edition)*.

Pearson, 2020.

[7] Fauzyah, Zahrah Asri Nur, Aceng Sambas, Prajanto Wahyu Adi, and De Rosal Ignatius Moses Setiadi. "Quantum Key Distribution-Assisted Image Encryption Using 7D and 2D Hyperchaotic Systems." Journal of Future Artificial Intelligence and Technologies 2, no. 1 (2025): 47-62.

[8] R. Flores-Carapia, V. M. Silva-Garc´ıa, M. A. Cardona-L´opez, *et al.*, “A chaotic digital signature algorithm based on a dynamic substitution box,” *Scientific Reports*, vol. 15, p. 2435, 2025. [Online]. Available: https://doi.org/10.1038/s41598-024-83943-x

[9] P. Radanliev, “Artificial intelligence and quantum cryptography,” *Journal of Analytical Science and Technology*, vol. 15, p. 4, 2024.

[10] F. Varghese and P. Sasikala, “A detailed review based on secure data transmission using cryptog- raphy and steganography,” *Wireless Personal Communications*, vol. 129, pp. 2291–2318, 2023.

[11] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, and D. Abbasinezhad-Mood, “Three-party secure data transmission in iot networks through the design of a lightweight authenticated key agreement scheme,” *Future Generation Computer Systems*, vol. 100, pp. 882–892, 2019.

[12] A. Vambol, “Polynomial-time plaintext-recovery attack on the matrix-based knapsack cipher,” *In- ternational Journal of Computing*, vol. 19, pp. 474–479, 2020.

[13] S. Dhall, S. Pal, and K. Sharma, “A chaos-based probabilistic block cipher for image encryption,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 1533–1543, 2022.

[14] J. Bezerra, V. de Almeida Camargo, and A. Molter, “A new efficient permutation-diffusion encryp- tion algorithm based on a chaotic map,” *Chaos, Solitons Fractals*, vol. 151, p. 111235, 2021.

[15] K. Prasad and H. Mahato, “Cryptography using generalized fibonacci matrices with affine-hill ci- pher,” *Journal of Discrete Mathematics, Science and Cryptography*, vol. 25, pp. 2341–2352, 2022.

[16] H. Wen, Y. Lin, L. Yang, and R. Chen, “Cryptanalysis of an image encryption scheme using variant hill cipher and chaos,” *Expert Systems with Applications*, vol. 250, p. 123748, 2024.

[17] L. Hill, “Cryptography in an algebraic alphabet,” *American Mathematical Monthly*, vol. 36, no. 6, pp. 306–312, 1929.

[18] L. S. Hill, “Concerning certain linear transformation apparatus of cryptography,” *The American Mathematical Monthly*, vol. 38, no. 3, pp. 135–154, 1931.

[19] L. Chen, G. Guo, and Z. Peng, “A hill cipher-based remote data possession checking in cloud storage,” *Security and Communication Networks*, vol. 7, no. 3, pp. 511–518, 2014.

[20] S. Sahoo, “A cancelable retinal biometric method based on maximum bin computation and his- togram bin encryption using modified hill cipher,” in *2022 IEEE Delhi Section Conference (DEL- CON)*, 2022, pp. 1–5.

[21] M. Lone and S. Qureshi, “Rgb image encryption based on symmetric keys using arnold transform, 3d chaotic map and affine hill cipher,” *Optik*, vol. 260, p. 168880, 2022.

[22] Z. Dawahdeh, S. Yaakob, and R. bin Othman, “A new image encryption technique combining elliptic curve cryptosystem with hill cipher,” *J. King Saud Univ. Comput. Inf.*

Sci., vol. 30, pp. 349–355, 2018.

[23] M. Essaid, I. Akharraz, and A. Saaidi, “Image encryption scheme based on a new secure variant of hill cipher and 1d chaotic maps,” *J. Inf. Secure. Appl.*, vol. 47, pp. 173–187, 2019.

[24] Y. Zheng, Q. Huang, S. Cai, X. Xiong, and L. Huang, “Image encryption based on novel hill cipher variant and 2d-igscm hyper-chaotic map,” *Nonlinear Dynamics*, pp. 1–19, 2024.

[25] S. Mohammed, A. Abbas, M. Ali, and et al., “Design and simulation of secure fiber optic commu- nication system utilizing hill cipher algorithm,” *J. Opt.*, pp. 1–9, 2023.

[26] J. Overbey, W. Traves, and J. Wojdylo, “On the keyspace of the hill cipher,” *Cryptologia*, vol. 29, no. 1, pp. 59–72, 2005.

[27] X. Chen, K. Makki, K. Yen, and N. Pissinou, “Sensor network security: a survey,” *IEEE Commun. Surv. Tutor.*, vol. 11, no. 2, pp. 52–73, 2009.

[28] Z. Hua, Y. Zhou, and H. Huang, “Cosine-transform-based chaotic system for image encryption,” *Inf. Sci. (Ny)*, vol. 480, pp. 403–419, 2019.

[29] F. Toktas, U. Erkan, and Z. Yetgin, “Cross-channel color image encryption through the 2d hy- perchaotic hybrid map of optimization test functions,” *Expert Syst. Appl.*, vol. 249, p. Sep. 2024, 2024.

[30] U. Erkan, A. Toktas, and Q. Lai, “2d hyperchaotic system based on schaffer function for image encryption,” *Expert Syst. Appl.*, vol. 213, p. Mar. 2023, 2023.

[31] Winarno, E.; Nugroho, K.; Adi, P.W.; Setiadi, D.R.I.M. Combined interleaved pattern to improve confusion-diffusion image encryption based on the hyperchaotic system. IEEE Access **2023**, 11, 69005–69021, 10.1109/ACCESS.2023.3285481.

[32] Y. Chen, H. Huang, K. Huang, M. Roohi, and C. Tang, “A selective chaos driven encryption technique for protecting medical images,” Phys. Scripta, vol. 100, no. 1, Jan. 2025, Art. no. 0152a3.

[33] B. Charya, G. S. Rath, S. K. Patra, and S. K. Panigrahy, “Novel methods of generating self-invertible matrix for hill cipher algorithm,” *International Journal of Computer Science and Applications*, vol. 4, no. 2, pp. 365–378, 2007.

[34] M. Rahman, A. Abidin, M. Yusof, and N. Usop, “Cryptography: a new approach of classical hill cipher,” *Int. J. Secur. App.*, vol. 7, no. 2, pp. 179–190, 2013.

[35] K. Agrawal and A. Gera, “Elliptic curve cryptography with hill cipher generation for secure text cryptosystem,” *Int. J. Comput. App.*, vol. 106, no. 1, 2014.

[36] A. Agarwal, “Secret key encryption algorithm using genetic algorithm,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 4, pp. 216–218, 2012.

[37] N. Sharma and S. Chirgaiya, “A novel approach to hill cipher,” *Int. J. Comput. Appl.*, vol. 108, no. 11, 2014.

[38] A. Siahaan, “Genetic algorithm in hill cipher encryption,” *American International Journal of Re-search in Science, Technology, Engineering, and Mathematics*, vol. 15, no. 1, pp. 84–89, 2016.

[39] F. H. Khan, R. Shams, F. Qazi, and D. Agha, “Hill cipher key generation algorithm by using an orthogonal matrix,” *International Journal of Innovative Science and Modern Engineering*, vol. 3, no. 3, pp. 5–7, 2015.

[40] Y. Chen, R. Xie, H. Zhang, et al., “Generation of the high-order random key matrix for hill cipher encryption using the modular multiplicative inverse of triangular matrices,” *Wireless Networks*, vol. 30, pp. 5697–5707, 2024.

[41] B. Acharya, S. K. Patra, and G. Panda, “Involutory, permuted and reiterative key matrix generation methods for hill cipher system,” *Journal Name*, vol. Volume, no. Issue, p. Pages, 2009.

[42] S. Saeednia, “How to make the hill cipher secure,” *Cryptologia*, vol. 24, no. 4, pp. 353–360, 2000.

[43] A. Y. Mahmoud and A. G. Chefranov, “Secure hill cipher modifications and key exchange protocol,” in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, vol. 2. IEEE, 2010, pp. 1–6.

[44] A. Mahmoud and A. Chefranov, “Hill cipher modification based on pseudo-random eigenvalues,” Applied Mathematics Information Science, vol. 8, no. 2, p. 505, 2014.

[45] M. Essaid, I. Akharraz, A. Saaidi, and et al., “Image encryption scheme based on a new secure variant of hill cipher and 1d chaotic maps,” Journal of Information Security and Applications, vol. 47, pp. 173–187, 2019.

[46] A. Krishna and K. Madhuravani, “A modified hill cipher using the randomized approach,” International Journal of Computer Networks and Information Security, vol. 5, no. 5, pp. 56–62, 2012.

[47] P. Lone and D. Singh, “Application of algebra and chaos theory in the security of color images,” *Optik*, vol. 218, p. 165155, 2020.

[48] L. Ahmad and Q. Shaima, “Encryption scheme for rgb images using chaos and affine hill cipher technique,” *Nonlinear Dynamics*, vol. 111, pp. 5919–5939, 2023.

[49] R. Hasoun, S. Khlebus, and H. Tayyeh, “A new approach of classical hill cipher in public key cryptography,” *International Journal of Nonlinear Analyses and Applications*, vol. 12, no. 2, pp. 1071–1082, 2021.

[50] J. Jin, X. Lei, C. Chen, M. Lu, L. Wu, and Z. Li, “A fuzzy zeroing neural network and its application on dynamic hill cipher,” *Neural Computing and Applications*, pp. 1–15, 2024.

[51] Y. Xi, Y. Ning, J. Jin, and F. Yu, “A dynamic hill cipher with arnold scrambling technique for medical images encryption,” *Mathematics*, vol. 12, no. 24, p. 3948, 2024.

[52] Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using the chaotic logistic map. *Image and Vision Computing, 24*(9), 926–934. https://doi.org/10.1016/j.imavis.2006.02.021

[53] Rukhin, A., Soto, J., & Nechvatal, J. (2010). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications Special Publication800-22R1a. (April)

[54] Paragas, Jessie. (2020). An Enhanced Cryptographic Algorithm in Securing Healthcare Medical Records. 1-6. 10.1109/ICVEE50212.2020.9243228.

[55] R. G. Barrieta, A. S. Canlas, D. M. A. Cortez, and K. E. Mata, “Modified Hill Cipher Algorithm using Myszkowski Transposition to address Known-Plaintext attack,” International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 10, no. IV, pp. 3242–3249, Apr. 202