









New Approach to Image Encryption Based on Large Invertible Pseudo-Random Matrices

Hicham Rrghout^{1*}, Mourad Kattass¹, Younes Qobbi², Naima Benazzi³, Abdellatif JarJar¹,
Abdelhamid Benazzi¹

¹ MATSI Laboratory, High School of Technology, Mohamed First University, Oujda 60000, Morocco

² MASI Laboratory, ENS, Sidi Mohamed Ben Abdellah University, Fès 30000, Morocco

³ EEM Laboratory, High School of Technology, Mohamed First University, Oujda 60000, Morocco

Corresponding Author Email: h.rrghout@ump.ac.ma

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijse.150603>

ABSTRACT

Received: 12 May 2025

Revised: 15 June 2025

Accepted: 25 June 2025

Available online: 30 June 2025

Keywords:

hill cipher, image encryption, chaotic maps, z/256z ring, modular arithmetic, cryptographic matrices

In the digital age, where secure image transmission is essential, we present an improved image encryption scheme based on large scalable Hill matrices defined over the $Z/256Z$ ring. The encryption matrix is constructed by multiplying two triangular matrices generated from chaotic maps, providing a high degree of randomness and unpredictability. Each block incorporates arbitrary square submatrices, enhancing the structural complexity of the encryption. Experiments conducted on a diverse set of images validate the robustness of our approach: the correlation between the clear and encrypted images is close to zero, the entropy reaches 7.99 bits per pixel, and the performance achieves an NPCR of 99.64%, a UACI of 33.45%, and an avalanche effect of 50.33%. These results significantly outperform those of traditional variants of the Hill cipher, highlighting the effectiveness of the combination of evolving matrices and chaotic sequences for reliable and efficient image encryption.

1. INTRODUCTION

The concern of securely sharing information over a computer network securely is delicate, requiring the use of robust security systems in order to transfer data securely, and cryptography. The studies [1-4] is among the solutions, which plays a major role in this situation. The main objective of digital image encryption is to convert an original image into an encrypted, unintelligible and secure version protecting it from unauthorized access. Chaos [5-8] plays an important role in modern cryptography, especially in the field of image encryption. Chaotic systems are particularly appreciated for their sensitivity to initial conditions and their complex and unpredictable behavior, crucial characteristics for strengthening the security of encryption algorithms. Several researches are based on chaos which allows to improve certain classical techniques, such as Hill, Vigenère [9-13], Feistel [14-16]. The Hill cipher [17-20] is a mathematical concept used in cryptography and provides an important solution for securely transferring data. It is based on the use of matrices to encrypt messages. A square matrix is used as the key to encrypt blocks of plaintext by converting them into numerical vectors. The encryption is performed by multiplying these vectors by the key matrix. To decrypt, the inverse of this matrix is used.

Much research has focused on improving the Hill number by incorporating chaotic systems. Gietaneh and Akele [21], proposed a secure algorithm to encrypt Tele-Birr information using an improved version of the Hill cipher with a symmetric key. The algorithm encrypts all Tele-Birr information using two keys, k_1 and k_2 , in the form of a character matrix, thereby

generating a unique key for one-time encryption. Using this symmetric key cryptography technique, all Tele-Birr information can be transformed into unreadable messages using a combination of 131 additional characters and symbols. The character set used in this improved cipher comprises 141 characters, which makes the ciphertext extremely difficult to decipher for potential attackers, due to the exponential increase in the key space. Billore and Patel [22] proposed an extended generalized Fibonacci matrix, related to the extended generalized Fibonacci sequences, and established some properties in addition to classical matrix algebra. They also introduced a modified public-key cryptography using these matrices as keys in the Hill cipher in an affine transformation, as well as a key agreement protocol for encryption and decryption, based on the combination of terms of the extended generalized Fibonacci sequences under prime modulo. Naim and Pacha [18] proposed a new image encryption algorithm combining the advanced Hill cipher and a 6D hyperchaotic system. The method uses the prime number 257 as a modulo, replacing zero pixels with pixels of value 256. The image is first divided into four equal parts, and then each part into blocks of four pixels. Four variables of the hyperchaotic system are used to permute the blocks, while the remaining two variables generate the Hill matrices. Finally, each block is encrypted with the Hill cipher to obtain the final encrypted image. Mfungo et al. [20] proposed a cryptosystem in which the encryption process begins by shifting each row of the state matrix to the left. The modified matrix is then encrypted using the Hill cipher. Next, the top value of each column—regardless of whether it is even or odd—is used to perform an

XOR operation with all other elements in the same column (excluding the top value). The intermediate image is then diffused using a sigmoid logistic map, followed by a Kronecker XOR product operation applied between pixels to enhance security. Finally, an additional diffusion stage is performed using other keys derived from the sigmoid logistic map, producing the final encrypted image. Zheng et al. [23] proposed a new cryptosystem by introducing a two-dimensional chaotic map, called iterative Gaussian sinusoidal map (2D-IGSCM), which provides better ergodicity and higher unpredictability. Then, to overcome the limitations of Hill encryption, they presented an improved version of the three-dimensional Hill encryption model (3D-HC). This model uses a dynamic column vector generated from chaotic sequences derived from 2D-IGSCM to enhance the encryption efficiency. Finally, by combining the 2D-IGSCM and 3D-HC models, they proposed a new image encryption method.

In the current context where securing digital images has become crucial, classical encryption methods such as the Hill cipher have significant limitations, including the small size of key matrices, insufficient diffusion, and vulnerability to differential and statistical attacks. These constraints limit the robustness and flexibility of encryption systems, particularly for processing high-resolution or high-detail images.

In this work, we suggest employing a large invertible matrix, derived from the block multiplication of two triangular matrices generated from chaotic systems. This innovative

construction significantly expands the key space and improves diffusion, while retaining the invertibility property essential for decryption. Thus, our approach overcomes the weaknesses of classical Hill cipher variants by offering a more robust, scalable system suited to modern digital image security requirements.

This work begins in Section 1 by introducing the security challenges in image transfer, along with existing solutions, and related work. Section 2 covers the theoretical background, while Section 3 details our proposed approach. Section 4 presents and compares the results with prior studies. Finally, we conclude the paper.

2. THEORETICAL FOUNDATIONS

2.1 The hill cipher

2.1.1 Classical hill cipher

The Hill cipher was published by Lester S. Hill in 1929. It is a polygraphic cipher [24, 25], that is, the letters are not (de)ciphered one after the other, but in packets, that is, the letters are grouped two by two. To encode a message using this method, the letters of the message are first grouped two by two, then each letter is replaced by a number, as shown in the following Table 1.

Table 1. Letter values in the Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example:

We code the message « J ADORE LES MATHS ».

We break it down into: JA-DO-RE-LE-SM-AT-HS then we replace with: (9;0) -(3;14) -(17;4) -(11;4) -(18;12) -(0;19) -(7;18). Then each pair of numbers (x;y) from the previous list is transformed into a new pair (x';y') of integers between 0 and 25, using a matrix:

$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ via the relationship $A \begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} x' \\ y' \end{pmatrix} [26]$. which means that:

$$\begin{cases} ax + by = x' [26] \\ cx + dy = y' [26] \end{cases} \text{ Matrix A is called the encryption key.}$$

Finally, these two numbers x' and y' are transformed into letters using the correspondence table. The recipient therefore receives the coded message, and from the pairs of numbers (x';y'), and he finds the pairs (x;y) in order to find the message in plain text.

The classic Hill symmetric encryption algorithm, applied to data encryption, has several drawbacks. In particular its vulnerability to plain text attacks. and the linearity problem that can be exploited by attacks to simplify the search for the key. In addition, the small size (2×2) of the matrix facilitates brute force attacks.

In order to overcome the limitations of classic Hill encryption, various improvements have been made, including:

2.1.2 Proposed improvement

Several improvements have been made, offering significant advances to the classical Hill cipher. However, despite these advances, the matrices used remain relatively small, which exposes the system to brute-force attacks. In this paper, we

propose a new improvement allowing the use of scalable-size matrices, thus enhancing the robustness of our cryptographic system. Consider the following two matrices M1 and M2:

$$M1 = \begin{pmatrix} I & A_1 & O \\ O & I & B_1 \\ O & O & I \end{pmatrix} \text{ et } M2 = \begin{pmatrix} I & O & O \\ A_2 & I & O \\ O & B_2 & I \end{pmatrix}$$

With:

A1, A2, B1 and B2 are any square matrices of the same size.

I and O denote respectively the identity matrix and the null matrix, all of the same dimension as the matrices A1, A2, B1, and B2.

The Hill matrix proposed in this improvement is of the form:

$$H = M_1 \times M_2$$

$$H = \begin{pmatrix} I + A_1A_2 & A_1 & O \\ A_2 & I + B_1B_2 & B_1 \\ O & B_2 & I \end{pmatrix}$$

The matrix proposed in this approach is characterized by a scalable size, such as:

If A1, A2, B1 and B2 are size 2×2 then our Hill matrix H is size 6×6.

If A1, A2, B1 and B2 are of size 3×3 then our Hill matrix H is of size 9×9.

In general,

If A1, A2, B1 and B2 are of size R×R then the Hill matrix H is of size 3R×3R.

To decrypt the encrypted data, we need the inverse matrix of H. This allows us to recover the original message.

Let's calculate the inverse of H.

We have $H^{-1} = M_2^{-1} \times M_1^{-1}$.

With:

$$M_1^{-1} = \begin{pmatrix} I & -A_1 & A_1B_1 \\ 0 & I & -B_1 \\ 0 & 0 & I \end{pmatrix} \text{ and } M_2^{-1} = \begin{pmatrix} I & 0 & 0 \\ -A_2 & I & 0 \\ B_2A_2 & -B_2 & I \end{pmatrix}$$

Let us show that M_1^{-1} et M_2^{-1} are respectively the inverse of the matrices M_1 et M_2 .

$$M_1 \times M_1^{-1} = \begin{pmatrix} I & A_1 & 0 \\ 0 & I & B_1 \\ 0 & 0 & I \end{pmatrix} \times \begin{pmatrix} I & -A_1 & A_1B_1 \\ 0 & I & -B_1 \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}$$

And:

$$\begin{pmatrix} I & 0 & 0 \\ A_2 & I & 0 \\ 0 & B_2 & I \end{pmatrix} \times \begin{pmatrix} I & 0 & 0 \\ -A_2 & I & 0 \\ B_2A_2 & -B_2 & I \end{pmatrix} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}$$

So:

$$H^{-1} = \begin{pmatrix} I & 0 & 0 \\ -A_2 & I & 0 \\ B_2A_2 & -B_2 & I \end{pmatrix} \times \begin{pmatrix} I & -A_1 & A_1B_1 \\ 0 & I & -B_1 \\ 0 & 0 & I \end{pmatrix}$$

Application Example:

In this example, we propose to use matrices A_1 , A_2 , B_1 and B_2 of size 2×2 not necessarily invertible. The construction matrices of M_1 and M_2 are of dimension 6×6 .

$$\text{Let: } A_1 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 1 \\ 2 & 4 \end{pmatrix} \text{ and } B_1 = \begin{pmatrix} 2 & 2 \\ 3 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 6 & 2 \\ 3 & 1 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} I & A_1 & 0 \\ 0 & I & B_1 \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

And:

$$M_2 = \begin{pmatrix} I & 0 & 0 \\ A_2 & I & 0 \\ 0 & B_2 & I \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 6 & 2 & 1 & 0 \\ 0 & 0 & 3 & 1 & 0 & 1 \end{pmatrix}$$

So:

$$H = M_1 \times M_2 = \begin{pmatrix} 7 & 9 & 1 & 2 & 0 & 0 \\ 14 & 20 & 3 & 4 & 0 & 0 \\ 2 & 1 & 19 & 6 & 2 & 2 \\ 2 & 4 & 21 & 8 & 3 & 1 \\ 0 & 0 & 6 & 2 & 1 & 0 \\ 0 & 0 & 3 & 1 & 0 & 1 \end{pmatrix}$$

The inverse matrix of M_1 is:

$$M_1^{-1} = \begin{pmatrix} I & -A_1 & A_1B_1 \\ 0 & I & -B_1 \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 & -2 & 8 & 4 \\ 0 & 1 & -3 & -4 & 18 & 10 \\ 0 & 0 & 1 & 0 & -2 & -2 \\ 0 & 0 & 0 & 1 & -3 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{mod } 256 = \begin{pmatrix} 1 & 0 & 255 & 254 & 8 & 4 \\ 0 & 1 & 253 & 252 & 18 & 10 \\ 0 & 0 & 1 & 0 & 254 & 254 \\ 0 & 0 & 0 & 1 & 253 & 255 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse matrix of M_2 is:

$$M_2^{-1} = \begin{pmatrix} I & 0 & 0 \\ -A_2 & I & 0 \\ B_2A_2 & -B_2 & I \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & -1 & 1 & 0 & 0 & 0 \\ -2 & -4 & 0 & 1 & 0 & 0 \\ 16 & 14 & -6 & -2 & 1 & 0 \\ 8 & 7 & -3 & -1 & 0 & 1 \end{pmatrix}$$

$$\text{mod } 256 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 254 & 255 & 1 & 0 & 0 & 0 \\ 254 & 252 & 0 & 1 & 0 & 0 \\ 16 & 14 & 250 & 254 & 1 & 0 \\ 8 & 7 & 253 & 255 & 0 & 1 \end{pmatrix}$$

Then the inverse matrix is:

$$H^{-1} = (M_2^{-1} \times M_1^{-1}) \text{mod } 256$$

$$= \begin{pmatrix} 1 & 0 & 255 & 254 & 8 & 4 \\ 0 & 1 & 253 & 252 & 18 & 10 \\ 254 & 255 & 6 & 8 & 220 & 236 \\ 254 & 252 & 14 & 21 & 165 & 207 \\ 16 & 14 & 192 & 166 & 143 & 218 \\ 8 & 7 & 224 & 211 & 199 & 110 \end{pmatrix}$$

To increase the complexity of our approach, we can choose the elementary matrices (A_1 , A_2 , B_1 et B_2) of a fairly large size.

2.2 Chaotic maps

2.2.1 The logistics map

The logistic map [26] is mathematically expressed by a quadratic recurrence relation governed by Eq. (1):

$$x_0 \in 0,5; 1 \quad [\mu_1 \in [3,57; 4]]$$

$$x_{n+1} = \mu_1 x_n (1 - x_n) \quad (1)$$

With $\mu_1 \in [3.57, 4]$ is the interval where the control parameter guarantees chaotic behavior.

Bifurcation diagram

The bifurcation diagram [27] shows the evolution of the x_{n+1} iterations of the logistic sequence (on the y-axis) as a function of the value of the control parameter μ_1 (on the x-axis). Figure 1 represents this bifurcation diagram, where we observe that the chaotic behavior manifests itself when the parameter $\mu_1 \geq 3.57$. The ideal values of μ_1 are around 4, because in this range, the amplitude of x_n covers the entire range between 0 and 1.

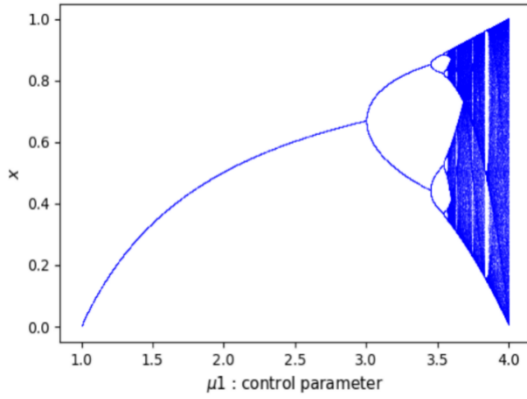


Figure 1. Logistic map bifurcation diagram

Lyapunov exponent

The Russian mathematician Alexander Markus-Lyapunov (1857-1918) developed a measure called the Lyapunov exponent [28], which allows to evaluate the rate of divergence between two chaotic trajectories with very close initial conditions. This exponent is expressed by the following equations.

We have:

$$\begin{aligned} f(x_i) &= \mu_1 x_i (1 - x_i) \\ f'(x_i) &= \mu_1 (1 - 2x_i) \\ \lambda(\mu_1, x_0) &= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^n \ln |\mu_1 (1 - 2x_i)| \end{aligned}$$

So:

The Lyapunov exponent diagram of the logistic map is given in the following Figure 2.

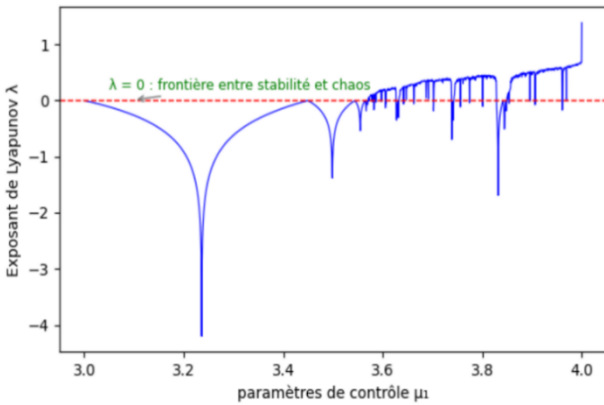


Figure 2. Lyapunov exponent of the logistic map

2.2.2 PWLCM card

The Piecewise Linear Chaotic Map (PWLCM) [29] is another example of a piecewise linear chaotic map, described by the following equation:

$$y_n = F(y_{n-1}, \mu_2) = \begin{cases} \frac{y_{n-1}}{\mu_2}, & 0 \leq y_{n-1} \leq \mu_2 \\ \frac{y_{n-1} - \mu_2}{0.5 - \mu_2}, & \mu_2 \leq y_{n-1} \leq 0.5 \\ F(1 - y_{n-1}, \mu_2), & \text{elsewhere} \end{cases} \quad (2)$$

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^n \ln |f'(x)| \quad (3)$$

Bifurcation diagram

The bifurcation diagram of the PWLCM map illustrates how the values of the iteration y_{n+1} evolve as a function of a control parameter. This diagram allows visualizing the dynamic regimes of the map, showing how the system changes from stable to chaotic behavior as the parameter varies. Figure 3 illustrates the bifurcation diagram of the PWLCM map.

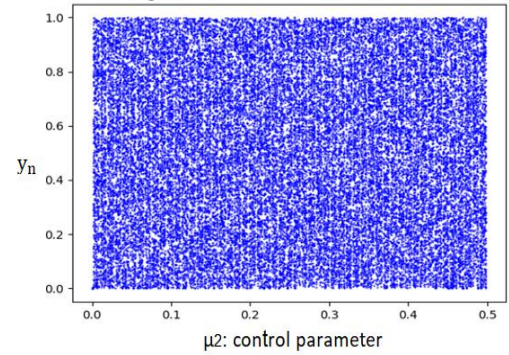


Figure 3. The bifurcation diagram of the PWLCM map

Lyapunov exponent

The Lyapunov exponent diagram of the PWLCM map is given in the following Figure 4:

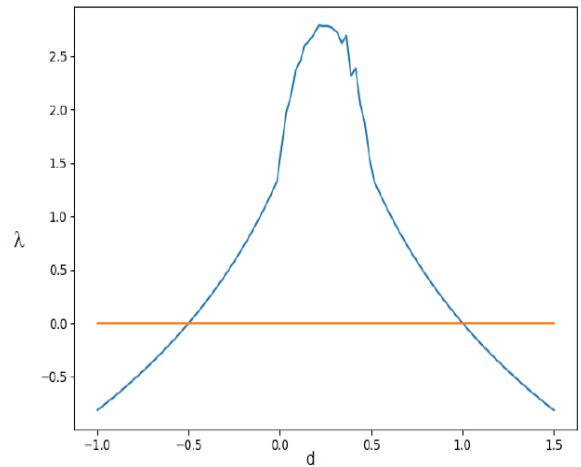


Figure 4. Lyapunov exponent of the PWLCM map

2.2.3 The sine map

The sine map [30] is another chaotic dynamical system, often used for its interesting properties in cryptography and chaotic simulation applications. It is based on a trigonometric function; the sine function is represented by the following expression:

$$x_{n+1} = \sin(ax_n)$$

x_n : represents the state of the system at iteration n .

a : a control parameter that determines the chaotic behavior of the system. It is often chosen between 0 and 2 to observe chaotic behavior. \sin is the trigonometric sine function.

3. PROPOSED METHOD

In this work, we present a new cryptographic system based on an improved version of the Hill cipher. This method uses a

large Hill matrix, reinforced by the integration of chaotic principles. Our study is organized around the following axes:

Axis 1: Preparation and adaptation of the vector representing the original image ($N \times M$)

Axis 2: Choice of the Hill matrix H

Axis 3: Confusion phase

Axis 4: Diffusion phase

Axis 5: Decryption process

3.1 Preparation and adaptation of the vector representing the original image of size ($N \times M$)

3.1.1 Construction of two pseudo-random tables K and T

From the sequences x and y , generated respectively by the logistic map and the PWLCM map, we construct two chaotic tables K and T of dimensions $(3NM, 4)$, whose coefficients belong to the ring $\mathbb{Z}/256\mathbb{Z}$. These tables are used in the fusion and diffusion steps of the encryption process. In addition, we also generate a binary control table B , of dimensions $(3NM, 2)$, according to the Algorithm 1 presented below:

Algorithm 1. Generation of chaotic vectors

//Generation of table K and T of size $(3NM, 4)$

For $i = 0$ to $3NM-1$

For $j = 0$ to 4

$K(i, j) = (\text{int})(\max(x(i), y(i)) * 10^8) \% 256$

$T(i, j) = (\text{int})(\max(2x(i), 3y(i)) * 10^8) \% 256$

// Generation of the control table B of size $(3NM, 2)$

For $i = 0$ to $3NM-1$

If $x(i) \geq y(i)$ then

$B(i, 1) = 0$

else $B(i, 1) = 1$:

end if

If $x(i) > y(2*i)$ then

$B(i, 2) = 0$

Else

$B(i, 2) = 1$

end if

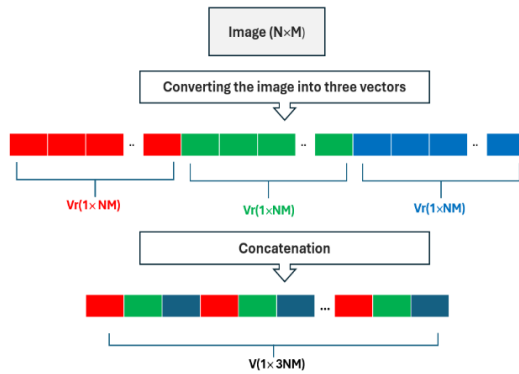


Figure 5. Vectorization

3.1.2 Vectorization

To reduce the correlation between adjacent pixels and increase the entropy, the original image is obscured by transforming it into a vector V of size $1 \times 3NM$, as shown in the following Figure 5.

The vector V is obtained by applying the following Algorithm 2.

Algorithm 2. Switching to vector (V)

For $i = 0$ to $NM-1$

If $B(i, 1) = 0$ Then

$V(3i) = Vb(i) \oplus \text{Inf}(K(i, 3); K(3i, 2))$

$V(3i + 1) = Vr(i) \oplus \text{Sup}(K(i+2, 4); K(3i, 1))$

$V(3i + 2) = Vg(i) \oplus \text{Sup}(K(2i+3); K(2i, 3))$

Else

$V(3i) = Vg(i) \oplus \text{Inf}(K(3i, 3); K(2i, 1))$

$V(3i + 1) = Vb(i) \oplus \text{Sup}(K(2i, 2); K(2i, 3))$

$V(3i + 2) = Vr(i) \oplus \text{Sup}(K(i, 1); K(2i, 2))$

End if

Next i

We can consider this first step as a lightweight encryption, safe from any statistical attack. However, a second round needs to be introduced to ensure that the system is protected against differential attacks.

3.1.3 Partitioning the vector v into blocks of size $(1 \times 3R)$

The choice of $3R$ is explained by the fact that the Hill matrix that will be used during encryption is of dimension $3R$. This size allows a combination to be made with the vector U , of dimension $(1 \times 3RS)$, and the Hill matrix, of dimension $(3R \times 3R)$, in order to perform the encryption.

The vector V is subdivided into two subvectors:

-A vector U of dimension $(1 \times 3RS)$

-A vector W of size $(1 \times L)$.

With:

$L = (3NM \% 3R)$

$S = \text{Int}(3NM/3R)$

S is the number of blocks of size $3R$ and L is the size of the vector W to be amputated. This situation is illustrated in the following Figure 6.

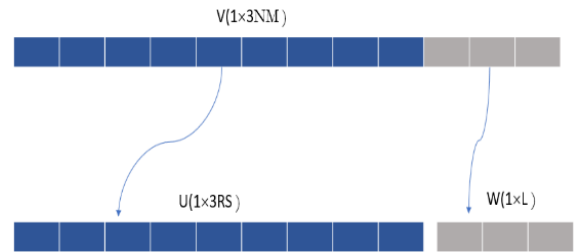


Figure 6. Image adaptation

3.2 Choosing the encryption matrix

In this approach, we propose using a Hill matrix H of size 9×9 . To construct this matrix, we choose the matrices A_1 , B_1 , A_2 , and B_2 , each of size 3×3 . And vector V representing the original image is divided into two subvectors:

- U of size $(1 \times 9S)$

-The vector to be amputated W of size $(1 \times L)$.

Let us choose the elements of the matrices A_1 , B_1 , A_2 , and B_2 from the pseudo-random arrays K and T . Using the following expressions for $0 \leq i \leq 2$ and $0 \leq j \leq 2$:

$$a_{ij} = T \left(\sum_{h=0}^N (K(h, 2) \% 300, K(i, 3)) \% 5 \right)$$

$$aa_{ij} = K \left(\sum_{h=0}^N (T(h, 2) \% 300, K(i, 3)) \% 5 \right)$$

$$b_{ij} = T \left(\sum_{h=0}^N (K(h, 3) \% 300, T(i, 1)) \% 5 \right)$$

$$bb_{ij} = K \left(\sum_{h=0}^N (T(h, 4) \% 300, K(i, 2)) \% 5 \right)$$

With a_{ij} , aa_{ij} , b_{ij} and bb_{ij} representing respectively the elements of matrices A_1 , A_2 , B_1 , and B_2 .

Then the Hill matrix is expressed in the following form:

$$H = M_1 \times M_2 = \begin{pmatrix} I & A_1 & O \\ O & I & B_1 \\ O & O & I \end{pmatrix} \times \begin{pmatrix} I & O & O \\ A_2 & I & O \\ O & B_2 & I \end{pmatrix}$$

3.3 Confusion phase

3.3.1 Initialization vector

Our encryption process initiates with a diffusion phase employing a (1×9) initialization vector P derived from the original image. This vector, generated through Algorithm 3, serves as the foundational cryptographic primitive for the subsequent diffusion operations.

Algorithm 3. Construction of the initialization vector

```

P[0] = 0
For i = 1 to 3 × N × M
P[i] = P[i] ⊕ U[i]
Next i
P[1] = P[0] ⊕ U[1]
P[2] = P[1] ⊕ U[2]
P[3] = P[2] ⊕ U[3]
P[4] = P[3] ⊕ U[4]
P[5] = P[4] ⊕ U[5]
P[6] = P[5] ⊕ U[6]
P[7] = P[6] ⊕ U[7]
P[8] = P[7] ⊕ U[8]

```

3.3.2 Modification of the first U_0 block

P is employed to transform the pixel values of the first image block by means of the following operations:

$$\begin{aligned}
U[0] &= U[0] \oplus P[0] \\
U[1] &= U[1] \oplus P[1] \\
U[2] &= U[2] \oplus P[2] \\
U[3] &= U[3] \oplus P[3] \\
U[4] &= U[4] \oplus P[4] \\
U[5] &= U[5] \oplus P[5] \\
U[6] &= U[6] \oplus P[6] \\
U[7] &= U[7] \oplus P[7] \\
U[8] &= U[8] \oplus P[8]
\end{aligned}$$

3.3.3 Generation of a pseudo-random translation vector V_c

To avoid the linearity problem during encryption, we create a pseudo-random vector V_c using the following Algorithm 4:

Algorithm 4. Construction of translation vector

```

For i = 0 to NM-1
If B(i,2) = 0 Then
Vc(i) = K(i,2) ⊕ K(i,3)
Else
Vc(i) = T(i,2) ⊕ T(i,3)
End If
Next i

```

Each block U_i of size (1×9) is subdivided into three sub-blocks of size (1×3) , as shown in Figure 7.

$$U_i = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \end{pmatrix} \text{ the } i \text{ block of the original image.}$$

$$C_i = \begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \end{pmatrix} \text{ the } i \text{ block of the encrypted image.}$$

$V_c(i)$ represents block i of V_c , composed of 9 elements. With X_0 , X_1 , X_2 , Y_0 , Y_1 et Y_2 are sub-blocks of size 3 each.

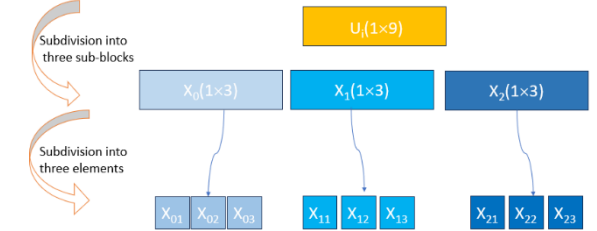


Figure 7. Subdivision of each block i into three sub-blocks of size 3

Then:

$$\begin{aligned}
C_i &= (H \times U_i) \oplus V_c(i) = (M_1 \times (M_2 \times U_i)) \oplus V_c(i) \\
&= \left(M_1 \times \left(\begin{pmatrix} I & O & O \\ A_2 & I & O \\ O & B_2 & I \end{pmatrix} \times \begin{pmatrix} X_0 \\ X_1 \\ X_2 \end{pmatrix} \right) \right) \oplus V_c(i) \\
&= \left(M_1 \times \begin{pmatrix} X_0 \\ A_2 X_0 + X_1 \\ B_2 X_1 + X_2 \end{pmatrix} \right) \oplus V_c(i)
\end{aligned}$$

So:

$$\begin{aligned}
\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \end{pmatrix} &= \left(\begin{pmatrix} I & A_1 & O \\ O & I & B_1 \\ O & O & I \end{pmatrix} \times \begin{pmatrix} X_0 \\ A_2 X_0 + X_1 \\ B_2 X_1 + X_2 \end{pmatrix} \right) \oplus \begin{pmatrix} V_c[9i] \\ V_c[9i+1] \\ V_c[9i+2] \\ V_c[9i+3] \\ V_c[9i+4] \\ V_c[9i+5] \\ V_c[9i+6] \\ V_c[9i+7] \\ V_c[9i+8] \end{pmatrix} \\
&= \begin{pmatrix} X_0 + A_1 A_2 X_0 + A_1 X_1 \\ A_2 X_0 + X_1 + B_1 B_2 X_1 + B_1 X_2 \\ B_2 X_1 + X_2 \end{pmatrix} \oplus \begin{pmatrix} V_c[9i] \\ V_c[9i+1] \\ V_c[9i+2] \\ V_c[9i+3] \\ V_c[9i+4] \\ V_c[9i+5] \\ V_c[9i+6] \\ V_c[9i+7] \\ V_c[9i+8] \end{pmatrix}
\end{aligned}$$

3.4 Diffusion phase

Algorithm 5. Diffusion process

For $i = 0 \dots S-1$

$$\begin{pmatrix} U[9(i+1)] \\ U[9(i+1)+1] \\ U[9(i+1)+2] \\ U[9(i+1)+3] \\ U[9(i+1)+4] \\ U[9(i+1)+5] \\ U[9(i+1)+6] \\ U[9(i+1)+7] \\ U[9(i+1)+8] \end{pmatrix} = \begin{pmatrix} U[9(i+1)] \\ U[9(i+1)+1] \\ U[9(i+1)+2] \\ U[9(i+1)+3] \\ U[9(i+1)+4] \\ U[9(i+1)+5] \\ U[9(i+1)+6] \\ U[9(i+1)+7] \\ U[9(i+1)+8] \end{pmatrix} \oplus \begin{pmatrix} Y[9i] \\ Y[9i+1] \\ Y[9i+2] \\ Y[9i+3] \\ Y[9i+4] \\ Y[9i+5] \\ Y[9i+6] \\ Y[9i+7] \\ Y[9i+8] \end{pmatrix}$$

By adopting the CBC (Cipher Block Chaining) mode, the

diffusion phase in the encryption process significantly improves security. Each plaintext block is combined with the previous ciphertext block using an XOR operation. This step can be described and implemented using the following Algorithm 5.

Diffusion creates a dependency between blocks, making the cipher more resistant to differential attacks. Figure 8 illustrates the different steps in the U(1×9S) vector encryption process.

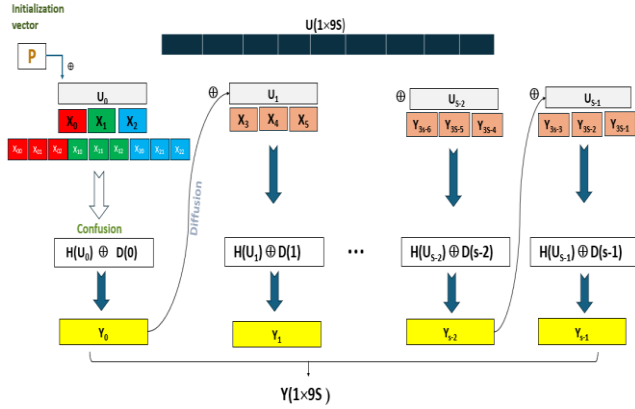


Figure 8. Encryption process

3.5 Encryption process of W(1×L)

The amputated vector is encrypted using the XOR operator, using the following Algorithm 6:

Algorithm 6. Encryption of the amputated vector

```

For i=0 to L-1:
    Z(i)= W(i) ⊕ K(N+i)
End For

```

The encrypted image is represented as a vector Yc (1×3NM), formed by the concatenation of Y and Z based on Algorithm 7.

Algorithm 7. Construction of the encrypted image vector

```

For i=0 to 3RS-1
    Yc[i]=Y[i]
Next i
For i= 0 to l-1
    Yc[i+3RS]=Z[i]
Next i

```

Figure 9 illustrates the vector Yc and the corresponding encrypted image, obtained by the concatenation of the vectors Y(1×9S) and Z(1×L), representing respectively the encryption of the vectors U(1×9S) and W(1×9S).

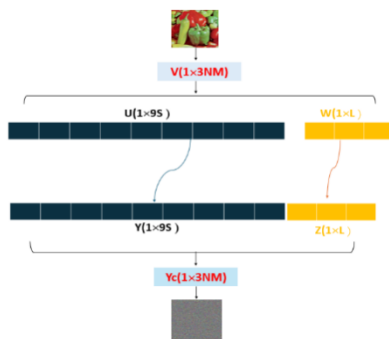


Figure 9. Construction of the encrypted image

3.6 Decryption process

Decryption takes place in the following steps:

-Load the encrypted image

-Transform the image into a row vector of dimensions 1×3NM

-Generating the Chaotic sequences.-

Find the inverse of H, noted as H⁻¹

Let us calculate H⁻¹ the inverse matrix of the matrix H.

H=M₁ × M₂ So H⁻¹= M₂⁻¹ × M₁⁻¹

We find:

$$H^{-1} = \begin{pmatrix} I & 0 & O \\ -A_2 & I & O \\ B_2A_2 & -B_2 & I \end{pmatrix} \times \begin{pmatrix} I & -A_1 & A_1B_1 \\ O & I & -B_1 \\ O & O & I \end{pmatrix}$$

$$= \begin{pmatrix} I & -A_1 & A_1B_1 \\ -A_2 & A_2A_1+I & -A_2A_1B_1-B_1 \\ B_2A_2 & -B_2A_2A_1-B_2 & B_2A_2A_1B_1+B_2B_1+I \end{pmatrix} \text{ mod } 256$$

4. RESULTS AND DISCUSSION

To validate the effectiveness of our cryptosystem, a series of tests were conducted on a collection of 50 standard images with varying resolutions, and the results were consistently positive.

4.1 Visual test

We present in Figure 10 the results for four iconic images used in image cryptography. Visually, the encrypted images differ completely from the original images and do not reveal any information or Similarity.

4.2 Key space

The key space should be as large as possible, ideally larger than 2¹⁰⁰. In our approach, two initial conditions x₀ and y₀, as well as two parameters μ₁ and μ₂, are encoded on 32 bits, providing a total key space of 2¹²⁸, much larger than 2¹⁰⁰.

Number of possible matrices

Matrices A1 and B1, each 3×3 in size, contain 9 elements that can take values between 0 and 255. Therefore:

The number of possible combinations for the elements of matrix A1 is: 256⁹ = (28)⁹ = 2⁷².

Similarly, the number of possible combinations for matrix B1 is also: 272.

Thus, the total number of combinations for matrix M1 formed from A1 and B1 is: 2⁷² × 2⁷² = 2¹⁴⁴.

Similarly, matrix M2, constructed from matrices A2 and B2, also admits :2¹⁴⁴ possible combinations.

Consequently, the number of possible matrices for matrix H results from the two matrices M1 and M2: 2¹⁴⁴ × 2¹⁴⁴ = 2²⁸⁸.

Consequently, the total number of possible configurations for matrix H, resulting from the combination of the two matrices M1 and M2, is: 2¹⁴⁴ × 2¹⁴⁴ = 2²⁸⁸.

Note:

If we use a matrix of size (12×12), the total number of possible matrices is equal to 2³⁸⁴.

This demonstrates the robustness of the method against brute force attacks, making any attempt at exhaustive exploration practically impossible.



Figure 10. Visual test

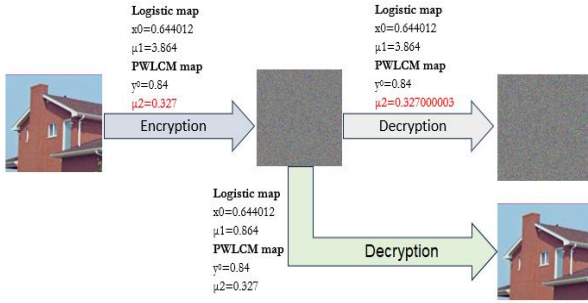


Figure 11. Key sensitivity

4.3 Key sensitivity

Our encryption system leverages two widely-recognized chaotic maps, selected for their cryptographic suitability and extreme sensitivity to initial conditions. As demonstrated in Figure 11, this characteristic guarantees the robust key responsiveness central to our security approach.

4.4 Correlation analysis

To thwart statistical attacks, the encryption system must eliminate the inherent pixel correlation present in uncompressed images [31]. Our method achieves this by reducing adjacent-pixel correlation to statistically insignificant levels, quantified by:

$$\text{corr}_{x,y} = \frac{\text{cov}(x,y)}{\sqrt{D(x) \times D(y)}}$$

Table 2 presents the values of the correlation coefficients in the three directions horizontal, vertical, and diagonal for several images encrypted using our proposed algorithm.

The experimental results demonstrate that all evaluated image metrics converged to values approaching zero, statistically confirming the algorithm's resistance to known

statistical cryptanalysis techniques.

Table 2. Correlation coefficients

Image		V	H	D
House 256 × 256	R	-0.0001	0.0012	-0.0013
	G	0.0003	-0.0022	-0.0004
	B	-0.0013	-0.0036	-0.0016
Baboon 512×512	R	0.0017	-0.0018	0.0032
	G	0.0003	0.0010	-0.0021
	B	0.0013	-0.0012	0.0008
Lena 512 × 512	R	-0.0004	0.0014	-0.0012
	G	0.0002	-0.0032	-0.0003
	B	-0.0023	-0.0046	-0.0017
Peppers 512×512	R	0.0024	0.0026	-0.0012
	G	-0.0008	-0.0021	0.0008
	B	0.0009	-0.0004	0.0001
	R	0.0002	-0.0017	0.0003
	G	0.0014	0.0019	-0.0009
	B	0.0005	0.0007	0.0015

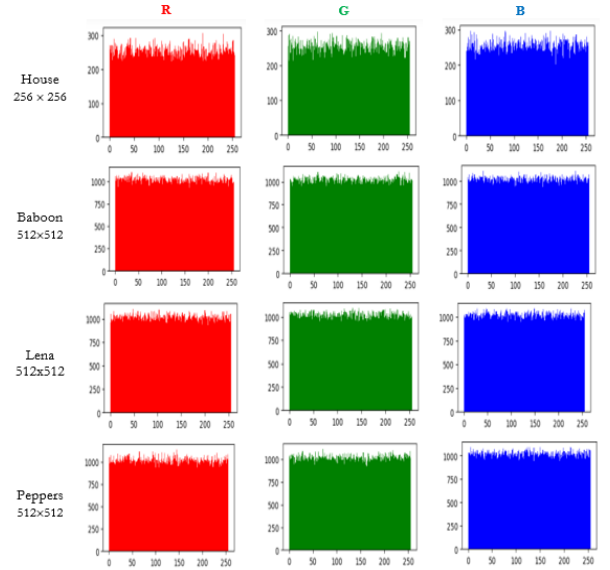


Figure 12. Histograms of encrypted images

4.5 Histogram analysis

As a fundamental analytical tool, the histogram visualizes pixel value distribution, enabling quantitative assessment of image contrast, brightness, and information content. A good encryption algorithm should scatter the values randomly or pseudo-randomly. The following Figure 12 shows the histograms of the encrypted images.

We observe that the encrypted images display a uniform histogram, ensuring a balanced distribution of pixel gray levels.

This result highlights the robustness of our approach against attacks based on histogram analysis.

4.6 Entropy analysis

As defined in Eq. (4), entropy [32] measures the level of random information contained within the ciphertext

$$H(m) = - \sum_{i=0}^{255} p(m_i) \log_2 (p(m_i)) \quad (4)$$

with $p(m_i)$ is the probability of occurrence of the symbol of class m_i in the encrypted data. The following Table 3 illustrates the entropy of the original image as well as that of the encrypted image.

Table 3. Entropy analysis

Image	Original Image	Encrypted Image
House	7.2718	7.9992
Peppers	7.2978	7.9993
Lena	7.2718	7.9994
Baboun	7.6444	7.9993
Airplane	6.5768	7.9992
Boat	7.1913	7.9993
Splash	7.2428	7.9993
Cameraman	6.9046	7.9991

4.7 NPCR and UACI

To rigorously evaluate encryption quality, we employ two established metrics [33] NPCR measures the percentage of differing pixels between original and encrypted images, while UACI quantifies the average intensity variation. These are computed as:

$$\text{NPCR} = \left(\frac{1}{NM} \sum_{i,j=1}^{NM} D(i,j) \right) * 100$$

$$\text{with : } D(i,j) = \begin{cases} 1 & \text{if } C_1(I_j) \neq C_2(I_j) \\ 0 & \text{if } C_1(I_j) = C_2(I_j) \end{cases} \tag{5}$$

$$\text{UACI} = \left(\frac{1}{NM} \sum_{i,j=1}^{NM} \frac{\text{ABS}(C_1(i,j) - C_2(i,j))}{255} \right) * 100 \tag{6}$$

Table 4. NPCR, UACI

Images	NPCR (%)	UACI (%)
House	99.64	33.45
Baboun	99.62	33.51
Lena	99.68	33.48
Peppers	99.65	33.47
Airplane	99.63	33.46
Boat	99.62	33.43
Splash	99.63	33.44
Cameraman	99.62	33.47

C1 and C2 represent two encrypted images derived from original images that differ by only a slight modification. Table

Table 6. Comparison with other approaches

Parameter	Image	Our Approach	Ref. [35]	Ref. [36]	Ref. [16]
Entropy	House	7.9992	7.9988	---	---
	Baboun	7.9993	---	7.9987	7.9993
	Lena	7.9994	7.9996	7.9991	7.9993
NPCR	House	99.64	99.61	---	---
	Baboun	99.62	---	99.63	99.61
	Lena	99.68	99.74	99.57	99.61
UACI	House	33.45	33.65	---	---
	Baboun	33.51	---	33.17	33,46
	Lena	33.48	33.52	33.35	33,46
Vertical Correlation	House	0.0005	---	---	---
	Baboun	0.0008	---	-0.0002	-0009
	Lena	-0.0008	---	-0.0040	0.00045

5. CONCLUSION

In this article, we employed a large invertible square matrix,

4 presents the values obtained using our method.

Our experimental results demonstrate NPCR and UACI values surpassing the critical thresholds of 99.6% and 33.4% respectively, confirming the algorithm's robust resistance against differential cryptanalysis attacks.

4.8 Avalanche effect

Following cryptographic best practices [34], we evaluate the avalanche characteristic using the standardized metric AE, where optimal encryption demands that single-bit input changes affect approximately 50% of output bits. The precise calculation is:

$$\text{AE} = \frac{\text{Number of changed bits}}{\text{Total number of bits in encrypted image}} \tag{7}$$

The following Table 5 shows the different results.

Table 5. Avalanche analysis

Images	Avalanche Effect (%)
House	50.33
Baboun	50.02
Lena	52.41
Peppers	50.12
Airplane	49.63
Boat	50.03

All observed values exceed 50%. Therefore, our system is resistant to any known attack.

4.9 Comparison

Table 6 presents a comprehensive performance comparison between our method and existing techniques.

The results obtained demonstrate the robustness and effectiveness of our approach. However, some limitations remain, particularly for real-time use. Indeed, encryption based on large invertible pseudo-random matrices requires significant computing resources, which slows down processing. It is therefore difficult to achieve the speed required for encrypting live videos or images. We plan to optimize this system in our future work.

which is derived from the multiplication of two block triangular matrices—one upper and one lower. After vectorizing and adapting the original image, we implemented

confusion and diffusion mechanisms to initiate the encryption process. The results of various tests conducted on standard images using a 9×9 Hill matrix demonstrate the robustness of our approach, showing strong resistance to known attacks.

REFERENCES

- [1] Dartois, P., Leroux, A., Robert, D., Wesolowski, B. (2024). SQISignHD: New dimensions in cryptography. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 3-32. https://doi.org/10.1007/978-3-031-58716-0_1
- [2] Thabit, F., Can, O., Aljahdali, A.O., Al-Gaphari, G.H., Alkhzaimi, H.A. (2023). Cryptography algorithms for enhancing IoT security. Internet of Things, 22: 100759. <https://doi.org/10.1016/j.iot.2023.100759>
- [3] Kaur, M., AlZubi, A.A., Walia, T.S., Yadav, V., Kumar, N., Singh, D., Lee, H.N. (2023). EGCrypto: A low-complexity elliptic galois cryptography model for secure data transmission in IoT. IEEE Access, 11: 90739-90748. <https://doi.org/10.1109/ACCESS.2023.3305271>
- [4] Döttling, N., Kolonelos, D., Lai, R.W., Lin, C., Malavolta, G., Rahimi, A. (2023). Efficient laconic cryptography from learning with errors. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 417-446. https://doi.org/10.1007/978-3-031-30620-4_14
- [5] JarJar, A. (2022). Vigenere and genetic cross-over acting at the restricted ASCII code level for color image encryption. Medical & Biological Engineering & Computing, 60(7): 2077-2093. <https://doi.org/10.1007/s11517-022-02566-4>
- [6] Iqbal, N., Hussain, I., Khan, M.A., Abbas, S., Yousaf, S. (2023). An efficient image cipher based on the 1D scrambled image and 2D logistic chaotic map. Multimedia Tools and Applications, 82(26): 40345-40373. <https://doi.org/10.1007/s11042-023-15037-1>
- [7] Alawida, M. (2023). A novel chaos-based permutation for image encryption. Journal of King Saud University-Computer and Information Sciences, 35(6): 101595. <https://doi.org/10.1016/j.jksuci.2023.101595>
- [8] Qobbi, Y., Abid, A., Jarjar, M., El Kaddouhi, S., Jarjar, A., Benazzi, A. (2023). Adaptation of a genetic operator and a dynamic S-box for chaotic encryption of medical and color images. Scientific African, 19: e01551. <https://doi.org/10.1016/j.sciaf.2023.e01551>
- [9] Kattass, M., Rrghout, H., Jarjar, M., Jarjar, A., Gmira, F., Benazzi, A. (2024). Chaotic image encryption using an improved vigenere cipher and a crossover operator. In Computing, Internet of Things and Data Analytics. ICCIDA 2023, pp. 181-191. https://doi.org/10.1007/978-3-031-53717-2_17
- [10] Erundu, U.I., Asani, E.O., Arowolo, M.O., Tyagi, A.K., Adebayo, N. (2023). An encryption and decryption model for data security using vigenere with advanced encryption standard. IGI Global, 141-159. <https://doi.org/10.4018/978-1-6684-5741-2.ch009>
- [11] Putra, N.B., Andika, B.C., Purba, A.D., Ridwan, M. (2023). Implementasi Sandi Vigenere Cipher dalam Mengenkripsikan Pesan. JOCITIS-Journal Science Infomatica and Robotics, 1(1): 42-50.
- [12] El Bourakkadi, H., Chemlal, A., Tabti, H., Kattass, M., Jarjar, A., Benazzi, A. (2024). Improved Vigenere approach incorporating pseudorandom affine functions for encrypting color images. International Journal of Electrical and Computer Engineering (IJECE), 14(3): 2684.
- [13] Chemlal, A., Tabti, H., El Bourakkadi, H., Rrghout, H., Jarjar, A., Benazzi, A. (2024). DNA-level action accompanied by Vigenere using strong pseudo random S-box for color image encryption. Multimedia Tools and Applications, 1-32. <https://doi.org/10.1007/s11042-024-19774-9>
- [14] Tabti, H., Abid, A., Jarjar, M., Jarjar, A., Najah, S., Zenkouar, K. (2024). A Feistel Network Followed by a Bitwise Crossover for Image Encryption. In International Conference on Digital Technologies and Applications, 288-297. https://doi.org/10.1007/978-3-031-68650-4_28
- [15] San Jose, C.C.G., Lazaro Jr, S.G. (2020). NHAF-512: New hash algorithm applying feistel cipher structure. International Journal, 8(8). <https://doi.org/10.30534/ijeter/2020/89882020>
- [16] Abid, A., Jarjar, M., Kattass, M., Rrghout, H., Jarjar, A., Benazzi, A. (2024). Genetic algorithm using feistel and genetic operator acting at the bit level for images encryption. International Journal of Safety & Security Engineering, 14(1). <https://doi.org/10.18280/ijssse.140102>
- [17] Rrghout, H., Kattass, M., Benazzi, N., Jarjar, M., Jarjar, A., Benazzi, A. (2024). Image encryption using hill cipher under a chaotic vector's control. In International Conference on Digital Technologies and Applications, 298-309. https://doi.org/10.1007/978-3-031-68650-4_29
- [18] Naim, M., Pacha, A.A. (2023). A novel image encryption algorithm based on advanced hill cipher and 6D hyperchaotic system. International Journal of Network Security, 25(5): 829-840. [https://doi.org/10.6633/IJNS.20230925\(5\).13](https://doi.org/10.6633/IJNS.20230925(5).13)
- [19] Lone, M.A., Qureshi, S. (2023). Encryption scheme for RGB images using chaos and affine hill cipher technique. Nonlinear Dynamics, 111(6): 5919-5939. <https://doi.org/10.1007/s11071-022-07995-2>
- [20] Mfungo, D.E., Fu, X., Wang, X., Xian, Y. (2023). Enhancing image encryption with the Kronecker xor product, the Hill Cipher, and the Sigmoid Logistic Map. Applied Sciences, 13(6): 4034. <https://doi.org/10.3390/app13064034>
- [21] Gietaneh, M.D., Akele, T.B. (2023). Enhancing the Hill cipher algorithm and employing a one time pad key generation technique. Abyssinia Journal of Engineering and Computing, 3(1): 1-10. <https://doi.org/10.20372/ajec.2023.v3.i1.808>
- [22] Billore, V., Patel, N. (2023). Cryptography utilizing the affine-hill cipher and extended generalized fibonacci matrices. Electronic Journal of Mathematical Analysis and Applications, 11(2): 1-12.
- [23] Zheng, Y., Huang, Q., Cai, S., Xiong, X., Huang, L. (2025). Image encryption based on novel Hill Cipher variant and 2D-IGSCM hyper-chaotic map. Nonlinear Dynamics, 113(3): 2811-2829. <https://doi.org/10.1007/s11071-024-10324-4>
- [24] Haryono, W. (2020). Comparison encryption of how to work caesar cipher, hill cipher, Blowfish and Twofish. Data Science: Journal of Computing and Applied Informatics, 4(2): 100-110. <https://doi.org/10.32734/jocai.v4.i2-4004>
- [25] Chilakala, H.S., Preeti, N. (2022). Advanced hill cipher

- hybrid cryptography model. In 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijaypur, India, pp. 1-5. <https://doi.org/10.1109/NKCon56289.2022.10126741>
- [26] Wang, Y., Liu, S., Khan, A. (2023). On fractional coupled logistic maps: Chaos analysis and fractal control. *Nonlinear Dynamics*, 111(6): 5889-5904. <https://doi.org/10.1007/s11071-022-08141-8>
- [27] Nandi, S. (2024). Bifurcation and chaotic behavior of two parameter family of generalized logistic maps. *Adv. Fixed Point Theory*, 14: Article-ID 20. <https://scik.org/index.php/afpt/article/view/8562>.
- [28] He, D., Parthasarathy, R., Li, H., Geng, Z. (2023). A fast image encryption algorithm based on logistic mapping and hyperchaotic Lorenz system for clear text correlation. *IEEE Access*, 11: 91441-91453. <https://doi.org/10.1109/ACCESS.2023.3305637>
- [29] Daoui, A., Mao, H., Yamni, M., Li, Q., Alfarraj, O., Abd El-Latif, A.A. (2023). Novel integer shmaliy transform and new multiparametric piecewise linear chaotic map for joint lossless compression and encryption of medical images in IoMTs. *Mathematics*, 11(16): 3619. <https://doi.org/10.3390/math11163619>
- [30] Demir, F.B., Tuncer, T., Kocamaz, A.F. (2020). A chaotic optimization method based on logistic-sine map for numerical function optimization. *Neural Computing and Applications*, 32: 14227-14239. <https://doi.org/10.1007/s00521-020-04815-9>
- [31] Kumar, S., Sharma, D. (2024). A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm. *Artificial Intelligence Review*, 57(4): 87. <https://doi.org/10.1007/s10462-024-10719-0>
- [32] Erkan, U., Toktas, A., Lai, Q. (2023). 2D hyperchaotic system based on Schaffer function for image encryption. *Expert Systems with Applications*, 213: 119076. <https://doi.org/10.1016/j.eswa.2022.119076>
- [33] Sheikh, A., Singh, K.U., Jain, A., Chauhan, J., Singh, T., Raja, L. (2024). Lightweight symmetric key encryption to improve the efficiency and safety of the IoT. In 2024 IEEE International Conference on Contemporary Computing and Communications (InC4), Bangalore, India, pp. 1-5. <https://doi.org/10.1109/InC460750.2024.10649289>
- [34] Mohammad Shah, I.N., Ismail, E.S., Samat, F., Nek Abd Rahman, N. (2023). Modified generalized feistel network block cipher for the Internet of Things. *Symmetry*, 15(4): 900. <https://doi.org/10.3390/sym15040900>
- [35] Hraoui, S., Gmira, F., Abbou, M.F., Oulidi, A.J., Jarjar, A. (2019). A new cryptosystem of color image using a dynamic-chaos hill cipher algorithm. *Procedia computer science*, 148, 399-408. <https://doi.org/10.1016/j.procs.2019.01.048>
- [36] Ghazvini, M., Mirzadi, M., Parvar, N. (2020). A modified method for image encryption based on chaotic map and genetic algorithm. *Multimedia Tools and Applications*, 79(37): 26927-26950. <https://doi.org/10.1007/s11042-020-09058-3>