Applied Mathematics and Computational Intelligence
Volume 15, No. 2, 2026 [40 – 58]

Double Encryption of Grayscale Images Using Hill Cipher with Self-Invertible Keys

Faridah Binti Yunos[1], Nurul Aina Binti Rosli[2*], Witriany Basri[3]

[1,2,3]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.

* Corresponding author: faridahy@upm.edu.my, ainans1205@gmail.com

Received: 19 September 2025
Revised: 23 September 2025
Accepted: 7 February 2026

ABSTRACT

The Hill Cipher (HC) algorithm is a symmetric technique with a simple structure and fast computations; however, it has a low level of security because both parties need to use and share the same private key through an insecure channel. In this study, a double encryption technique for grayscale images has been proposed by combining the HC with self-invertible keys (SI). In this method, the encryption and decryption processes are carried out using a key matrix that is SI, where the elements of the key are Integer Matrices with Integer Eigenvalues (IMIE). To assess the quality and resilience of the encryption, a thorough analysis of the grayscale image that has been encrypted twice is performed using metrics such as Entropy, Peak Signal-to-Noise Ratio (PSNR), and Unified Average Changing Intensity (UACI). As a result, the transmission of grayscale images through this new variant of the HC is found to be competitive in terms of security and efficiency compared to several existing encryption techniques, where the proposed method achieves an entropy of 7.9947, PSNR of 8.5230, and UACI of 30.0486% for the double encrypted image, demonstrating its competitiveness compared to existing techniques.

Keywords: Hill Cipher, self-invertible key, encryption, decryption, grayscale image

1 INTRODUCTION

Cryptology, derived from the Greek words 'kryptós', meaning 'hidden', and 'lógos', meaning 'word', encompasses two main branches: cryptography and cryptanalysis. Cryptography is the art of secret writing that has been practiced for centuries and has now become an essential tool to protect our digital information. Meanwhile, cryptanalysis is the art of breaking these secret codes. Therefore, cryptology is not just an academic endeavor, but a vital foundation for information security, supporting critical technological aspects, from internet communication to digital currencies and secure government operations.

With the advancement of technology, cryptology has evolved from a specialized field to a cornerstone of cybersecurity. In this interconnected world, the issues of data misuse and cyber threats are increasingly concerning. Strong cryptographic techniques are essential not only for protecting sensitive information but also for verifying user identities. The landscape of cryptography is

diverse, featuring three primary schemes: symmetric cryptography, which uses a single key for both encryption and decryption; public-key cryptography, which utilizes different keys for the two processes; and hash functions, which convert data into an irreversible format to ensure integrity.

A HC method is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a modulo 26 number. It was first established by [1]. HC is a well-known symmetrical key scheme that affects the linear transformation in the message space. It consists of an integer vector with m dimensional. The numerical form of the original text in HC is often written as a matrix P. The matrix K is selected to be the main key that is kept secret and C is the ciphertext. The encryption process of the matrix P to C can be expressed by $C \equiv KP \pmod{N}$ with N being a positive integer. Meanwhile, the decryption process from C to P is expressed by $P \equiv K^{-1}C \pmod{N}$ with K^{-1} being the inverse of K in modulo N. This HC technique is a simple structure and is quick to calculate if we get K^{-1}. If the main key has an inverse, decrypting the ciphertext becomes more complex as the key dimensions increase. This is because the inverse of this key must first be searched. The inverse of a matrix is typically obtained using the elementary row operation and the concept of multiplication inverse idea in modulo arithmetic. Using SI matrices as the secret key of HC eliminates the need to identify its inverse before the decryption process.

Previous research has shown that this algorithm has limitations, particularly in its vulnerability to known plaintext attacks. This has led to the need to modify its encryption algorithm to enhance security. Rivest, Shamir, and Adleman (RSA) and Elliptic Curve Cryptosystem (ECC), on the other hand, use an asymmetric encryption algorithm, which is a security system that is still employed by the industry today, even in the face of challenges from the quantum era.

Several variations of the HC have been developed by previous researchers such as, Acharya et al. [2], [3], [4] introduced involutory and permuted key matrices to eliminate the need for inverse computation; Naveen Kumar et al. [5] proposed a technique combining bit rotation and extended Hill Cipher; and Dawahdeh et al. [6] integrated ECC to enhance security in image encryption. [7] refined a method to find two orthogonal column vectors, as required in the method to construct the SI matrix when applying Integer Matrices with Integer Eigenvalues (IMIE) compared to previous method by [8]. Such keys can accelerate the encryption process in HC systems. This is because we do not need to calculate the inverse of the key, which usually requires a lot of computation. However, the probability that an attacker compromises the secrecy of this key may increase, due to the smaller key space of SI keys compared to the total number of invertible keys. An attacker could also exploit the characteristics of orthogonal vectors as a basis for SI. Future researchers, especially those using HC in image processing, need to consider methods to prevent this exploitation, in addition to standard analyses involving grayscale images, such as entropy, PSNR, and UACI.

In today's era of digital communication, the security of sensitive information is extremely important. Grayscale image data, which are often used in various applications ranging from medical imaging to remote sensing, is highly susceptible to unauthorized access and modification. The question arises: can modifications to the combined system proposed by [7] be utilized in images-related communications, and what metrics can be used to assess its effectiveness? In response to this challenge, this paper introduces a new encryption technique that employs a two-layered approach to HC encryption while using an SI matrix based on IMIE in the construction of the encryption key. The efficiency of this technique is further evaluated using three types of analysis: Entropy,

PSNR, and UACI.

The rest of this paper is organized as follows. Section 2 presents the literature review. Section 3, the preliminaries of this study are presented. Meanwhile, Section 4 explains the proposed method for the key generation, encryption, and decryption processes. An example of implementation is given in Section 5. The security analysis based on entropy, PSNR, and UACI is explained in Section 6. This was followed by a conclusion in the last section.

2 LITERATURE REVIEW

Several methods have been proposed to generate a SI key for the HC algorithm by [2], including the involutory key method [3], the permuted and reiterative matrix approach [4], the biometric protection using modified HC [5] and the two-stage image encryption using bit rotation and extended HC [6] because the inverse of the matrix used to encrypt the plaintext is not always available or exists. Furthermore, these methods encompass less computational complexity since the inverse of the matrix is not required when decrypting in HC. Moreover, due to its linear nature, the basic HC succumbs to known-plaintext attacks.

[3] has proposed an involutory, permuted, and reiterative key matrix generation method for the HC system. Involutory matrices eliminated the need for matrix inverses in hill decryption. The permuted matrix and the reiterative key generation approach create different keys for each block of data encryption, increasing resilience to various attacks. Moreover, [4] proposed a technique to secure biometric traits using the modified HC with an involutory key and a robust cryptosystem.

[5] presents a novel image encryption method with two main stages. In the first stage, the value of each pixel in the input image is transformed to an eight-bit binary representation. A number of bits equal to the length of the supplied password are rotated and reversed. The second stage leverages the extended HC approach, which uses an involutory matrix built from the same password to increase security. The effectiveness of the method is evaluated using statistical evaluations, differential analysis, and PSNR, implemented on several images. As a result, the evaluations indicate that the combined approach is more effective in securing images than individual methods.

[9] worked on the solutions of the SI matrix based on IMIE construction. The system introduces an efficient way to construct matrices from a given set of eigenvectors, offering greater flexibility in selecting eigenvalues. This method also applies to non-diagonalizable matrices when a basis of generalized eigenvectors is available. Eigenvalues can be chosen to exclude zero, control multiplicities, or include one, simplifying the factorization of the characteristic equation. The process of diagonalizing a square matrix involves expressing it as a SI matrix (detail in Section 3.2). This technique is used by [8] and [7].

[6] presented on image encryption technique that combines ECC with HC (ECCHC). This approach effectively blends the cryptographic strengths of an ECC with the operational efficiency of the HC. Employing a SI key matrix, the ECCHC eliminates the need to find an inverse key matrix during the decryption process. The approach exhibits strong performance metrics that include entropy, PSNR, and UACI, making it a promising solution for secure image transmission in various fields.

[10] proposed an image encryption algorithm that uses a hybrid of chaotic maps. The original image is divided into smaller parts using the Discrete Wavelet Transform. These parts are then shuffled and substituted to achieve permutation and diffusion properties. The system is a unique two-dimensional (2D), nonlinear, discrete-time chaotic designed to improve encryption quality and efficiency by using chaotic behavior, which is unpredictable and complex. This makes the encryption process faster than that of the stochastic process and more secure, as demonstrated by dynamical analysis and sample entropy approaches. This makes it an excellent and useful tool for applications such as wireless communications. Furthermore, combining different types of chaotic map into a single image makes the encryption process even more robust, providing greater security against future attacks.

[11] presented an innovative enhancement to classical HC by incorporating RSA to strengthen its security against common vulnerabilities, particularly known plaintext attacks. Traditional weaknesses of HC, such as its limited key space and susceptibility to attacks, are mitigated through a dynamic and secure key matrix generation process. The effectiveness of the proposed system is validated through statistical analyzes, which demonstrate strong randomness and resilience against various attacks, representing a significant advancement over the HC and its existing modifications. In this version, the plaintext (P) is encrypted using two transformations: briefly, in the first step, the P is transformed into the second text (C_H) by multiplying the SI key (K) with P modulo 256. This encryption process is similar to that of the HC. Next, C_H is transformed into ciphertext (C_R) through the expression $(C_H)^e$ modulo N, where the public key (e) must meet the requirements of the generation of RSA keys. The decryption of C_R also occurs in two stages. However, this process saves the computation time for the decryption key (K^{-1}), while the computation time for the secret key d is such that $ed \equiv 1 \pmod{\phi(N)}$ still depends on the public key e.

3 PRELIMINARIES

In Section 3.1, we restated a method for identifying two column vectors that are orthogonal presented by [7]. Followed by the method for generating SI matrices based on IMIE in Section 3.2.

3.1 Method of Identifying Two Column Vectors that are Orthogonal

[7] provides a method to determine the appropriate value for two orthogonal vectors. Specifically, if we know the values of $\vec{u}$, we can identify the values of $\vec{v}$ that are orthogonal to $\vec{u}$. We present an approach as follows:

Let $\vec{u} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}$ be a given vector. We seek to find vectors $\vec{v} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$ such that $\vec{u} \cdot \vec{v} \equiv 0 \pmod{N}$, where the dot product is expressed as:

$$\vec{u} \cdot \vec{v} = a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n. \tag{1}$$

Setting this expression equal to 0, we obtain the orthogonality condition:

$$a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n \equiv 0 \pmod{N}. \tag{2}$$

To find $\vec{v}$, we need to consider x_i for all $i = 1, 2, 3, \ldots, n$ as the header of (2). Consider x_1 as follows:

$$a_1x_1 \equiv -(a_2x_2 + a_3x_3 + \cdots + a_nx_n) \pmod{N} \tag{3}$$

$$x_1 \equiv -(a_2x_2 + a_3x_3 + \cdots + a_nx_n) \cdot a_1^{-1} \pmod{N}. \tag{4}$$

with $(a_1, N) = 1$.
The parametric vector form of the solution is given by:

$$\vec{v} \equiv x_2 \begin{bmatrix} -a_2a_1^{-1} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -a_3a_1^{-1} \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + x_n \begin{bmatrix} -a_na_1^{-1} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \pmod{N}. \tag{5}$$

This can be expressed as:

$$\vec{v} \equiv \begin{bmatrix} -(a_2x_2 + a_3x_3 + \cdots + a_nx_n)a_1^{-1} \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \pmod{N}. \tag{6}$$

Therefore, the answer is in the plane of

$$\text{Span} \left\{ \begin{bmatrix} -a_2a_1^{-1} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} a_3a_1^{-1} \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_na_1^{-1} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\} \pmod{N}. \tag{7}$$

where each element represents all solutions of $\vec{u} \cdot \vec{v} \equiv 0 \pmod{N}$.
We can verify that (for example) the first element is orthogonal vectors with $\vec{u}$ as follows:

$$\begin{bmatrix} -a_2a_1^{-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \perp \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \text{ because } \begin{bmatrix} -a_2a_1^{-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \equiv 0 \pmod{N} \tag{8}$$

For the next solution, we setup (2) as x_2 as a header:

$a_2x_2 \equiv -(a_1x_1 + a_3x_3 + \cdots + a_nx_n) \pmod{N}$.

$x_2 \equiv -(a_1x_1 + a_3x_3 + \cdots + a_nx_n)a_2^{-1} \pmod{N}$ with $(a_2, N) = 1$.

The parametric vector form of the solution is given by:

$$\vec{v} \equiv x_1 \begin{bmatrix} 1 \\ -a_1a_2^{-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ -a_3a_2^{-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + x_n \begin{bmatrix} 0 \\ -a_na_2^{-1} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \pmod{N}. \tag{9}$$

This can be expressed as:

$$\vec{v} \equiv \begin{bmatrix} x_1 \\ -(a_1x_2 + a_3x_3 + \cdots + a_nx_n)a_2^{-1} \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \pmod{N}. \tag{10}$$

Therefore, the answer is in the plane of

$$\text{Span}\left\{ \begin{bmatrix} 1 \\ -a_1a_2^{-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -a_3a_2^{-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ -a_na_2^{-1} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\} \pmod{N}. \tag{11}$$

where each element represents all solutions of $\vec{u} \cdot \vec{v} \equiv 0 \pmod{N}$.
We can find all solutions by using the similar technique as mentioned above. However, we only select certain values of $\vec{v}$ to ensure that the encryption key is SI using the method discussed in Section 3.2.

Example 1. Let $\vec{u} \equiv \begin{bmatrix} 251 \\ 67 \\ 126 \\ 578 \end{bmatrix} \pmod{851}$ be a given vector where $a_1 = 251, a_2 = 67, a_3 = 126$ and $a_4 = 578$. We seek to find vectors $\vec{v} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ such that $\vec{u} \cdot \vec{v} \equiv 0 \pmod{851}$.

Solution:
The first solution for $\vec{v}$ when x_1 is a header of Eqn. (2) is produced as follows:

$$\vec{v} \equiv x_2 \begin{bmatrix} -a_2 a_1^{-1} \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -a_3 a_1^{-1} \\ 0 \\ 1 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -a_4 a_1^{-1} \\ 0 \\ 0 \\ 1 \end{bmatrix} \pmod{851}.$$

Substituting the values $a_1 = 251, a_2 = 67, a_3 = 126$ and $a_4 = 578$, we get:

$$\vec{v} \equiv x_2 \begin{bmatrix} 827 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 247 \\ 0 \\ 1 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 174 \\ 0 \\ 0 \\ 1 \end{bmatrix} \pmod{851}.$$

Therefore, the solution are all bases in the plane of

$$\text{Span} \left\{ \begin{bmatrix} 827 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 247 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 174 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} \pmod{851}.$$

3.2 Generation of Self-Invertible Key with IMIE

A simple method was introduced to construct such a matrix from a predetermined set of eigenvectors. Concisely, construct $A = BDB^{-1}$ which is IMIE, where D is a diagonal matrix whose diagonal entries are the eigenvalues of A while the column of matrix B which is in an invertible matrix form as the basis of the eigenvectors for A. The following are some principles to be considered in [9].

Definition 1. [9] A matrix A is an *IMIE* if it is an integer entry matrix with (all) integer eigenvalues. In other words, the characteristic polynomial of A factors completely over $\mathbb{Z}$.

Theorem 1. [9] Given two n-vectors $\vec{u}$, $\vec{v} \in \mathbb{Z}^n$ with $\vec{u} \cdot \vec{v} = \beta$, the matrix $B = I_n + \vec{u}\vec{v}^\intercal$ has $\det B = 1 + \beta$. In addition, if $\beta \neq -1$, then $B^{-1} = I_n - \frac{1}{1+\beta}\vec{u}\vec{v}^\intercal$.
Here, $\vec{u}\vec{v}^\intercal$ is just the $n \times n$ "outer product" matrix, sometimes written as $\vec{u} \times \vec{v}$.

A direct consequence of this theorem forms the following property when $\beta = 0$.

Corollary 1. [9] Given two n-vectors $\vec{u}$, $\vec{v} \in \mathbb{Z}^n$ which are orthogonal, the matrix $B = I_n + \vec{u}\vec{v}^\intercal$ has $\det B = 1$ and its inverse $B^{-1} = I_n - \vec{u}\vec{v}^\intercal$.

The following theorem was utilized to generate a self-invertible matrix A with IMIE.

Theorem 2. [9] Let B be an integer matrix $n \times n$ with determinant $\delta \neq 0$. Let D be a diagonal matrix whose diagonal entries are all integers that are mutually congruent modulo δ. Then $A = BDB^{-1}$ is an integer matrix.
More concisely: Let $B \in GL_n(\mathbb{Z})$ and $D = (\lambda_1, \ldots, \lambda_n)$ with $\lambda_i \in \mathbb{Z}$. If $\lambda_1 \equiv \lambda_2 \equiv \cdots \equiv \lambda_n \mod \delta$, then BDB^{-1} is a diagonalizable IMIE.

Therefore, we restated the technique to find the SI key that was produced by [7] as follows:

Algorithm 1. Generate Self-invertible Matrices with IMIE
Input: Vectors $\vec{u}$ and $\vec{v}$ that are orthogonal where the inner product of $\vec{u}$ and $\vec{v}$ are equal to 0.
Output: A SI matrix A
Calculation:

1. Compute the outer product of $\vec{u}$ and $\vec{v}$ by $\vec{u}\vec{v}^{\intercal} \equiv \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \pmod{N}$.
2. Add the identity matrix I_n to the outer product to construct $B \equiv I_n + \vec{u}\vec{v}^{\intercal} \pmod{N}$.
3. Compute the inverse matrix of B. That is, $B^{-1} \equiv I_n - \vec{u}\vec{v}^{\intercal} \pmod{N}$.
4. Identify all 2^n combination of diagonal elements D in the form of $D = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ where the eigenvalues $\lambda_n = \pm 1$.
5. For each diagonal matrix D, compute $A \equiv BDB^{-1}$.
6. If $A^2 = I$, store all possible diagonal matrices D to generate all valid SI matrices A; otherwise, discard and try another D.
7. Return to A.

Example 2. Since the inner product of $\vec{u}$ and $\vec{v}$ is 0 (mod 851), we choose two vectors where $\vec{u} = \begin{bmatrix} 251 \\ 67 \\ 126 \\ 578 \end{bmatrix}$ and $\vec{v} = \begin{bmatrix} 174 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ in modulo 851 that are orthogonal from Example 1.

Solution:
Step 1 : The outer product of $\vec{u}$ and $\vec{v}$ which yields:

$$\vec{u}\vec{v}^{\intercal} \equiv \begin{bmatrix} 251 \\ 67 \\ 126 \\ 578 \end{bmatrix} \begin{bmatrix} 174 & 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} 273 & 0 & 0 & 251 \\ 595 & 0 & 0 & 67 \\ 649 & 0 & 0 & 126 \\ 154 & 0 & 0 & 578 \end{bmatrix} \pmod{851}.$$

Step 2 : Next, matrix B by $B = I_n + \vec{u}\vec{v}^{\intercal}$ which yields :

$$B \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 273 & 0 & 0 & 251 \\ 595 & 0 & 0 & 67 \\ 649 & 0 & 0 & 126 \\ 154 & 0 & 0 & 578 \end{bmatrix} \equiv \begin{bmatrix} 274 & 0 & 0 & 251 \\ 595 & 1 & 0 & 67 \\ 649 & 0 & 1 & 126 \\ 154 & 0 & 0 & 579 \end{bmatrix} \pmod{851}.$$

Step 3 : Then, its inverse by $B^{-1} = I_n - \vec{u}\vec{v}^{\intercal}$.

$$B^{-1} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 273 & 0 & 0 & 251 \\ 595 & 0 & 0 & 67 \\ 649 & 0 & 0 & 126 \\ 154 & 0 & 0 & 578 \end{bmatrix} \equiv \begin{bmatrix} 579 & 0 & 0 & 600 \\ 256 & 1 & 0 & 784 \\ 202 & 0 & 1 & 725 \\ 697 & 0 & 0 & 274 \end{bmatrix} \pmod{851}.$$

Step 4 : Assume that D is a diagonal matrix where its diagonal element is an integer from interval [1, 255] and $n = 4$. We select diagonal [1, 255] because image encryption typically operates with 8-bit values, where grayscale pixel intensities range from 0 to 255. This selection ensures compatibility with standard image processing techniques. The possible diagonal configurations for D are as follows:

Table 1 : Diagonal Matrix Representations

Diagonal	**Diagonal Values**	**Diagonal**	**Diagonal Values**
1	[1, 1, 1, 1]	9	[1, 1, 1, 255]
2	[255, 1, 1, 1]	10	[255, 1, 1, 255]
3	[1, 255, 1, 1]	11	[1, 255, 1, 255]
4	[255, 255, 1, 1]	12	[255, 255, 1, 255]
5	[1, 1, 255, 1]	13	[1, 1, 255, 255]
6	[255, 1, 255, 1]	14	[255, 1, 255, 255]
7	[1, 255, 255, 1]	15	[1, 255, 255, 255]
8	[255, 255, 255, 1]	16	[255, 255, 255, 255]

Step 5 : After generating all combinations, there are eight valid diagonal configurations with their corresponding SI matrices A as shown in Table 2 :

Table 2 : Valid Diagonal Configurations with their Corresponding SI Matrices A

Diagonal	Self-invertible Matrix A	Diagonal	Self-invertible Matrix A
1	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	10	$\begin{bmatrix} 255 & 0 & 0 & 0 \\ 503 & 1 & 0 & 849 \\ 603 & 0 & 1 & 517 \\ 0 & 0 & 0 & 255 \end{bmatrix}$
3	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 348 & 255 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	12	$\begin{bmatrix} 255 & 0 & 0 & 0 \\ 0 & 255 & 0 & 0 \\ 603 & 0 & 1 & 517 \\ 0 & 0 & 0 & 255 \end{bmatrix}$
5	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 248 & 0 & 255 & 334 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	14	$\begin{bmatrix} 255 & 0 & 0 & 0 \\ 503 & 1 & 0 & 849 \\ 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 255 \end{bmatrix}$
7	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 348 & 255 & 0 & 2 \\ 248 & 0 & 255 & 334 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	16	$\begin{bmatrix} 255 & 0 & 0 & 0 \\ 0 & 255 & 0 & 0 \\ 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 255 \end{bmatrix}$

4 PROPOSED TECHNIQUE

Assume that user A wants to send an image M to user B and they agreed to use IMIE as a SI key matrix. If user A wants to send the grayscale image 256×256 pixels to user B, the first thing to do is to get a numerical value that matches the red, green, blue (RGB) colors for each pixel for color images, then both User A and User B should follow the following algorithms:

Algorithm 2. Key Generation Process
Input: A large prime number $N > 256$.
Output: Two self-invertible matrices A_1 and A_2.
Computation:

1. User A and User B choose a large prime number (private key), that is, $N > 256$.
2. Both User A and User B choose two different SI matrices, i.e. $A_1 \equiv BD_1B^{-1} \mod N$ and $A_2 \equiv BD_2B^{-1} \mod N$ using Algorithm 1. To ensure that the original plaintext is not preserved after transformation, the matrices A_1 and A_2 must satisfy $A_2A_1 \neq I_n$.
3. Return to A_1 and A_2.

Algorithm 3. Encryption Process
Input: The numerical values in RGB colors.
Output: The value of entropy, PSNR and UACI.
Computation:

1. User A has to obtain the numerical values in RGB colors for each pixel of the original colored image.
2. Convert the original image to grayscale by mapping its pixel values using the formula: $Y = 0.299R + 0.587G + 0.114B$. This formula has been extensively used by previous researchers and its suitability against other formulas has been evaluated [12].

3. Separate the pixel values of the grayscale image into blocks of size four, ensuring that the block size corresponds to the dimensions of the plaintext matrix.
4. The four-size block is now the plain text message in the form of $[P_1|P_2|P_3|\ldots]$.
5. The first transformation from P_1 to C_1 will be calculated using

   $$C_1 \equiv A_1 \cdot P_1 \pmod{256}$$

   and the same process will be repeated for $P_2, P_3, \ldots$.
6. The second transformation from C_1 to C_2 will be calculated using

   $$C_2 \equiv A_2 \cdot C_1 \pmod{256}$$

   and the same process will be repeated for $P_2, P_3, \ldots$.
7. Construct the ciphered image C_2 from the values of the ciphered vectors.
8. Send the ciphered image C_2 to User B.
9. Generate a chaotic sequence using the logistic map:

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

   with normalization and scaling of the sequence to the range [0, 255] and reshaping it into a matrix of the same dimensions as the encrypted image. In this case x_n is the sequence value in iteration n and r is the control parameter that determines the chaotic behavior.
10. Apply pixel-level XOR diffusion using the $Chaotic\_matrix$:

    $$C_{final} = C_2 \oplus Chaotic\_matrix$$

    where $Chaotic\_matrix$ represents the normalized chaotic sequence, scaled and transformed into the range [0,255] using formula: $C[i] = \text{mod}(\lfloor x_i \times 256 \rfloor, 256)$.
11. Do entropy, PSNR and UACI analyses on C_{final}.
12. Return to entropy, PSNR and UACI values.

Algorithm 4. Decryption Process
Input: Ciphered image from User A
Output: Original Image
Computation:

1. Split the pixel value of the ciphered image into four size blocks.
2. The transformation from C_2 to C_1 will be calculated using

   $$C_1 \equiv A_2 \cdot C_2 \pmod{256}$$

   and the same process will be repeated for other blocks.
3. The transformation from C_1 to P_1 will be calculated using

   $$P_1 \equiv A_1 \cdot C_1 \pmod{256}$$

   and the same process will be repeated for the other blocks.
4. Construct the original image from the values in the deciphered vectors.
5. Return to original image

5 EXAMPLE OF IMPLEMENTATION

The key generation, encryption, and decryption of images in this study are based on Algorithms 2, 3 and 4 respectively. These steps are explained in detail through the following example:

Key Generation

1. Choose $N = 851$ so that $N > 256$.
2. Choose two different SI matrices from Example 2 i.e.

$$A_1 \equiv \begin{bmatrix} 255 & 0 & 0 & 0 \\ 503 & 1 & 0 & 849 \\ 603 & 0 & 1 & 517 \\ 0 & 0 & 0 & 255 \end{bmatrix} \pmod{851}$$

and

$$A_2 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 348 & 255 & 0 & 2 \\ 248 & 0 & 255 & 334 \\ 0 & 0 & 0 & 1 \end{bmatrix} \pmod{851}.$$

Encryption (User A)

1. User A obtains the numerical values in RGB colors for each pixel of the original colored image.

$$R = \begin{bmatrix} 225 & 225 & 225 & 226 & \cdots \\ 226 & 225 & 225 & 225 & \cdots \\ 225 & 225 & 226 & 226 & \cdots \\ 226 & 225 & 224 & 225 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$G = \begin{bmatrix} 135 & 135 & 135 & 136 & \cdots \\ 136 & 135 & 136 & 136 & \cdots \\ 136 & 136 & 134 & 134 & \cdots \\ 134 & 134 & 133 & 132 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$B = \begin{bmatrix} 124 & 124 & 124 & 125 & \cdots \\ 125 & 124 & 122 & 122 & \cdots \\ 122 & 122 & 119 & 119 & \cdots \\ 119 & 116 & 115 & 115 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

2. Map elements in the original image matrix to the corresponding grayscale values using $Y = 0.299R + 0.587G + 0.114B$.

$$\begin{bmatrix} 161 & 161 & 161 & 162 & \cdots \\ 162 & 161 & 161 & 161 & \cdots \\ 161 & 161 & 160 & 160 & \cdots \\ 160 & 159 & 158 & 158 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

3. Separate the grayscale image's pixel values into blocks of size four as highlighted shown above ensuring that the block size matches the dimensions of matrix A.
4. The block of size four is now the plaintext message.

$$P_1 = \begin{bmatrix}161\\162\\161\\160\end{bmatrix}, P_2 = \begin{bmatrix}161\\161\\161\\159\end{bmatrix}, P_3 = \begin{bmatrix}161\\161\\160\\158\end{bmatrix}, P_4 = \begin{bmatrix}162\\161\\160\\158\end{bmatrix}, \cdots$$

5. The first transformation from P_1 to C_1 will be calculated using
$$C_1 \equiv A_1 \cdot P_1 \equiv \begin{bmatrix}255 & 0 & 0 & 0\\503 & 1 & 0 & 849\\603 & 0 & 1 & 517\\0 & 0 & 0 & 255\end{bmatrix}\begin{bmatrix}161\\162\\161\\160\end{bmatrix} \equiv \begin{bmatrix}95\\153\\252\\96\end{bmatrix} \pmod{256}.$$
Do the same process for the other blocks.
6. The second transformation from C_1 to C_2 will be calculated using
$$C_2 \equiv A_2 \cdot C_1 \equiv \begin{bmatrix}1 & 0 & 0 & 0\\348 & 255 & 0 & 2\\248 & 0 & 255 & 334\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}95\\153\\252\\96\end{bmatrix} \equiv \begin{bmatrix}95\\75\\76\\96\end{bmatrix} \pmod{256}.$$
Do the same process for the other blocks. Therefore, the pixel value of the ciphered image C_2 is

$$\begin{bmatrix}95 & 95 & 95 & 94 & \cdots\\75 & 159 & 242 & 159 & \cdots\\76 & 159 & 243 & 160 & \cdots\\96 & 97 & 98 & 98 & \cdots\\\vdots & \vdots & \vdots & \vdots & \vdots\end{bmatrix}$$

7. Construct the ciphered image C_2 from the values of the ciphered vectors.
8. Send the ciphered image C_2 to User B.
9. Consider the following logistic map parameters for chaotic diffusion in image encryption:

$$x_0 = 0.5, \quad r = 3.99$$

where we choose $x_0 = 0.5$ to ensure that the sequence stays within the chaotic regime while avoiding a very small or very large value. For $r = 3.99$, the logistic map exhibits chaotic behavior when $3.57 \leq r \leq 4$ and so r that we choose is close to the upper chaotic limit that maximizes the unpredictability of the sequence. We compute 16 values of the chaotic sequence by using logistic map $x_{n+1} = r \cdot x_n \cdot (1 - x_n)$ as follows :

$$\begin{aligned}
x_1 &= 3.99 \times 0.5 \times (1 - 0.5) = 0.9975\\
x_2 &= 3.99 \times 0.9975 \times (1 - 0.9975) = 0.0099\\
x_3 &= 3.99 \times 0.0099 \times (1 - 0.0099) = 0.0389\\
&\ldots\\
x_{15} &= 3.99 \times 0.6823 \times (1 - 0.6823) = 0.8672\\
x_{16} &= 3.99 \times 0.8672 \times (1 - 0.8672) = 0.4574
\end{aligned}$$

We scale the chaotic values to the range $[0, 255]$ using:

$$C[i] = \text{mod}(\lfloor x_i \times 256 \rfloor, 256)$$

Multiply each value by 256 and take the floor: (Eg. $\lfloor 0.9975 \times 256 \rfloor = 255$)
Resulting in the *Chaotic_matrix*:

$$[C(i)] = \begin{bmatrix} 255 & 2 & 9 & 38 \\ 129 & 252 & 10 & 38 \\ 130 & 80 & 195 & 63 \\ 42 & 14 & 152 & 77 \end{bmatrix}$$

10. Compute the XOR values between C_2 and *Chaotic_matrix* as follows:

$$\begin{aligned} C_{final} =& C_2 \oplus Chaotic\_matrix \\ =& \begin{bmatrix} 95 \oplus 255 & 95 \oplus 2 & 95 \oplus 9 & 94 \oplus 38 & \cdots \\ 75 \oplus 129 & 159 \oplus 252 & 242 \oplus 10 & 159 \oplus 38 & \cdots \\ 76 \oplus 130 & 159 \oplus 80 & 243 \oplus 195 & 160 \oplus 63 & \cdots \\ 96 \oplus 42 & 97 \oplus 14 & 98 \oplus 152 & 98 \oplus 77 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ =& \begin{bmatrix} 160 & 93 & 86 & 120 & \cdots \\ 202 & 99 & 248 & 185 & \cdots \\ 206 & 207 & 48 & 159 & \cdots \\ 74 & 111 & 250 & 47 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{aligned}$$

11. Do entropy, PSNR and UACI analyses on C_{final}.

Decryption (User B)

1. User B receives the ciphered image C_2 from user A.
2. Convert the ciphered image into a pixel value.
3. Split the pixel value of the ciphered image into four size blocks.

$$\begin{bmatrix} 95 & 95 & 95 & 94 & \cdots \\ 75 & 159 & 242 & 159 & \cdots \\ 76 & 159 & 243 & 160 & \cdots \\ 96 & 97 & 98 & 98 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

4. The multiplication of A_2 by the first vector of C_2 will be calculate for the first decryption:

$$C_1 = A_2 \cdot C_2 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 348 & 255 & 0 & 2 \\ 248 & 0 & 255 & 334 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 95 \\ 75 \\ 76 \\ 96 \end{bmatrix} \equiv \begin{bmatrix} 95 \\ 153 \\ 252 \\ 96 \end{bmatrix} \pmod{256}.$$

Do the same process for the other blocks.

5. The multiplication of A_1 by the first vector of C_1 will be calculated for the second decryption:

$$P_1 = A_1 \cdot C_1 \equiv \begin{bmatrix} 255 & 0 & 0 & 0 \\ 503 & 1 & 0 & 849 \\ 603 & 0 & 1 & 517 \\ 0 & 0 & 0 & 255 \end{bmatrix} \begin{bmatrix} 95 \\ 153 \\ 252 \\ 96 \end{bmatrix} \equiv \begin{bmatrix} 161 \\ 162 \\ 161 \\ 160 \end{bmatrix} \pmod{256}.$$

Do the same process for the other blocks.

6. Construct the original image from the values in the deciphered vectors.

6 SECURITY ANALYSIS

This section will provide the formulas for entropy, PSNR, and UACI that will be used to analyze the cipher image in the form of a grayscale image. It aims to evaluate the effectiveness of image encryption.

6.1 Entropy

Entropy is one of the statistical parameters used to evaluate the encrypted image. Reflects the frequency of occurring patterns and is based on the probability distribution of pixel values, measuring the degree of randomness in an image. For a grayscale image of size 256×256, the ideal theoretical entropy value is 8. A higher entropy value, especially one close to 8, indicates a more effective encryption process [6]. In general, the greater the entropy, the more difficult it is to break the cryptosystem. The entropy value in this study was computed using the following formula:

$$\text{Entropy } = \sum_{x=0}^{255} [P(x) \times \log_2 \left(\frac{1}{P(x)} \right)] \tag{12}$$

where $P(x)$ is the probability of the pixel value x and computed by

$$P(x) = \frac{\text{The frequency of pixel value } x}{\text{Total number of image pixels}}$$

6.2 Peak Signal to Noise Ratio (PSNR)

PSNR is a metric used to evaluate the performance of an image encryption algorithm. It quantifies the quality of encryption by measuring the distortion in the decrypted image compared to the original. A higher PSNR value indicates minimal or no data loss in the decrypted image, indicating that it closely matches the original, demonstrating the high efficiency of the encryption technique [13]. In this study, the PSNR value was calculated using the following formula:

$$\text{PSNR} = 20 \times \log_{10} \left(\frac{255}{\text{MSE}} \right) \tag{13}$$

where MSE is Mean Square Error between the original image and decrypted image and computed by

$$\text{MSE} = \frac{1}{256 \times 256} \sum_{i=1}^{256} \sum_{j=1}^{256} (A_{ij} - B_{ij})^2 \tag{14}$$

where A_{ij} is the pixel value of the original image and B_{ij} is the pixel value of the decrypted image. When comparing PSNR value, the minimum value is better. Furthermore, when comparing the original image with the encrypted image; if MSE increases, then PSNR decreases, and this indicates that the encrypted image is more randomness. High value of MSE and low value of PSNR indicates that two images are completely different. On the other hand, high value of PSNR is indicates the high quality image [14].

6.3 Unified Average Changing Intensity (UACI)

UACI is measured the difference between the original image and the ciphered image [6]. This transformation alters the pixel values and structure of the image to make it unintelligible, preventing unauthorized access. It is used to assess the strength of the encryption technique. Its value depends on the size and format of the image [15] and the ideal value is 30-35% [16]. UACI measures the average change in intensity between the original and ciphered images. The highest UACI means that the proposed technique is resistant to

differential attacks, which means a cryptanalysis technique that examines how small changes in the original image affect the ciphered image. The UACI value was obtained by the following equation:

$$\text{UACI} = \frac{1}{256 \times 256} \sum_{i=1}^{256} \sum_{j=1}^{256} \left| \frac{A(i,j) - B(i,j)}{255} \right| \times 100\% \tag{15}$$

where $A(i,j)$ is the pixel value of the original image and $B(i,j)$ is the pixel value of the encrypted image. The formula is used for the grayscale image of size 256×256.

6.4 Result and Discussion

Figure 1 shows the original grayscale image for Cameraman with 256×256 pixels, the first and the double-ciphered image. MATLAB R2024b (24.2.0.2806996) 64-bit (win64) software, executed on a Intel(R) Core(TM) i5-7200U computer with a CPU of 2.50 GHz and 4 GB of RAM, is used for encryption and decryption processes.

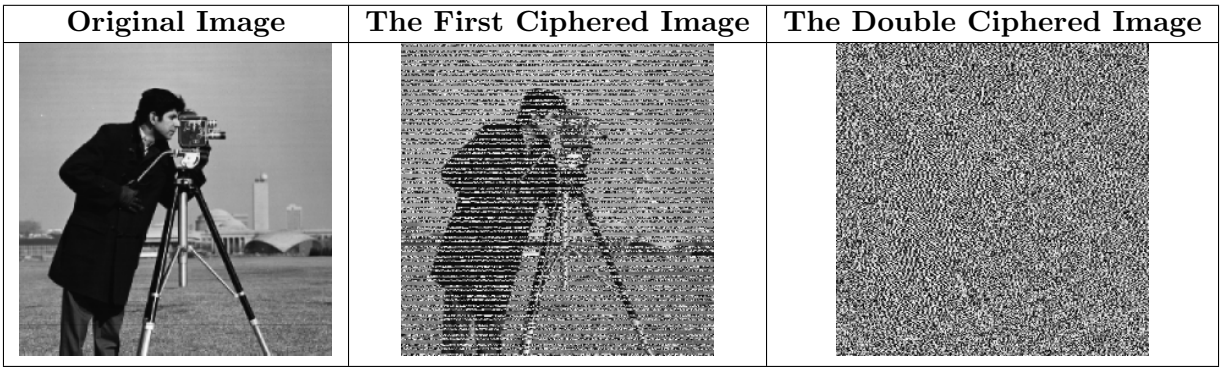

Figure 1 : The Original of Grayscale Image, the First and the Double-Ciphered Image for Cameraman

Table 3 : Entropy, PSNR, and UACI for Cameraman Image

The Method	**Cameraman**		
	Entropy	**PSNR**	**UACI**
The proposed technique	7.9947	8.5230	30.0486
Yasser et al. (2020) [10]	7.9991	8.3785	33.6425
Dawahdeh et al. (2018) [6]	7.9848	6.9999	35.5263
Naveen Kumar et al. (2012) [5]	NA	8.4221	35.6000
Expected values	**8**	**Minimum value is better**	**30-35%**

The analysis results comparing the efficiency and performance of the proposed encryption method to previous methods are shown in Table 3. For the Cameraman grayscale image, the proposed technique has a slightly lower entropy than [10], but is still close to the ideal value of 8. High entropy indicates a uniform distribution of pixel intensities in the encrypted image, showing strong randomness and resistance to statistical attacks. Statistical attacks use patterns in encrypted data to uncover information about the original data or encryption key. According to [6], a higher entropy makes it harder to break the encryption system by reducing predictability.

The PSNR values for the Cameraman grayscale image show how effective the proposed encryption technique is. Lower PSNR values are better because they mean more distortion between the original and decrypted images, making it harder for attackers to get useful information. The proposed technique has a PSNR value of 8.5230, which is slightly higher than [6] but competitive with [10] and [5]. The proposed method balances distortion and security well. Although [6] achieves lower PSNR values, it can increase computational complexity or decrease entropy. The proposed technique maintains competitive PSNR values and high entropy, ensuring a strong and efficient encryption system.

UACI values measure how sensitive an image encryption technique is to pixel changes. A high UACI value indicates that small changes in the original image lead to large differences in the encrypted image, enhancing resistance to differential attacks. The proposed technique achieves a UACI value of 30.0486%, which falls within the ideal range of 30–35% for effective image encryption. Although it is lower than the values reported in [6],[10] and [5], it still demonstrates the method's ability to produce significant pixel intensity variations and provides a reasonable level of security against differential cryptanalysis.

Table 4 : Image Security Measures for Ciphered Image of Cameraman, Lena, Baboon, Angry Bird and Albert Einstein

The Image	**Entropy**	**PSNR**	**UACI**
Lena	7.9948	9.3884	27.5939
Baboon	7.9929	10.0404	25.9319
Angry Bird	9.9583	6.0001	40.3205
Albert Einstein	7.9968	7.5897	33.1688

Table 5 : Analysis of Encryption and Decryption Time

Image	**Encryption and Decryption Time (seconds)**			
	Proposed Technique	**Yasser et al. (2020)**	**Dawahdeh et al. (2018)**	**Naveen Kumar et al. (2012)**
Cameraman	2.0270	NA	1.2588	NA
Lena	2.1460	NA	1.2615	NA
Baboon	2.3453	NA	1.2635	NA
Angry Bird	2.2525	NA	NA	NA
Albert Einstein	2.0998	NA	1.2736	NA

Table 4 shows the security metrics (entropy, PSNR, and UACI) for four test images: Lena, Baboon, Angry Bird, and Albert Einstein. These metrics help evaluate the encryption technique's performance on images with different textures and colors. Entropy: The Lena, Baboon, and Albert Einstein images have entropy values close to 8, indicating strong randomness and resistance to statistical attacks. The Angry Bird image has a lower entropy value, suggesting weaker randomness. PSNR: The Baboon image has a higher PSNR value, indicating that the encryption technique may be less effective for detailed images. The Angry Bird image has a lower PSNR value, showing higher distortion and better security. UACI: The Albert Einstein image has an UACI value in the range of ideal, indicating good security against differential attacks. Overall, the Albert Einstein image, with its balanced texture and detail, appears to yield the best results, while exhibiting lower complexity or limited variations in pixel intensities like Lena, Angry Bird, and Baboon, which show slightly reduced effectiveness.

Table 5 shows the time analysis for the encryption and decryption processes for various grayscale images (Cameraman, Lena, Baboon, Angry Bird, Albert Einstein). The proposed technique takes more time for both encryption and decryption compared to [6]. This extra time reflects the robustness and thoroughness of

the approach, mainly due to complex matrix operations and chaotic key transformations. Despite the higher computational cost, the method is efficient and practical for real-time applications, balancing security in terms of entropy, PSNR, and UACI.

7 CONCLUSION

In conclusion, this study introduces a double encryption technique for grayscale images using the Hill Cipher with self-invertible keys featuring IMIE characteristics. Comprehensive analysis through metrics such as entropy that indicating effective pixel randomness, competitive PSNR reflecting image quality preservation, and a UACI within the ideal range, not only demonstrates the quality and effectiveness of image encryption but ultimately makes it more robust for secure image transmission. Future studies are encouraged to investigate various analyzes in more depth, including key sensitivity analysis, histogram analysis, correlation coefficient analysis, differential attack analysis, and time complexity analysis. These analyses provide comprehensive information on the robustness and efficiency of the proposed image encryption technique. By exploring these aspects, we hope to improve existing cryptographic methods, thereby contributing to the advancement of secure image transmission technologies.

REFERENCES

[1] L. S. Hill, "Cryptography in an algebraic alphabet," *The American Mathematical Monthly*, vol. 36, no. 6, pp. 306–312, 1929.

[2] B. Acharya, G. S. Rath, S. K. Patra, and S. K. Panigrahy, "Novel methods of generating self-invertible matrix for hill cipher algorithm," *International Journal of Security*, vol. 1, pp. 14–21, 2007.

[3] B. Acharya, S. K. Patra, and G. Panda, "Involutory, permuted and reiterative key matrix generation methods for hill cipher system," *International Journal of Recent Trends in Engineering*, vol. 1, no. 4, pp. 106–108, 2009.

[4] B. Acharya, M. D. Sharma, S. Tiwari, and V. K. Minz, "Privacy protection of biometric traits using modified hill cipher with involutory key and robust cryptosystem," *Procedia Computer Science*, vol. 2, pp. 242–247, 2010.

[5] S. K. Naveen Kumar, H. S. Sharath Kumar, and H. T. Panduranga, "Encryption approach for images using bits rotation reversal and extended hill cipher techniques," *International journal of computer applications*, vol. 59, no. 16, 2012.

[6] Z. E. Dawahdeh, S. N. Yaakob, and R. R. bin Othman, "A new image encryption technique combining elliptic curve cryptosystem with hill cipher," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 349–355, 2018.

[7] F. Yunos, N. R. Nik Roslan, and A. Hamzah, "Self-invertible keys based orthogonal vectors for hill cipher system," *Submitted to Numerical Algebra, Control and Optimization*, 2025.

[8] F. B. Yunos, N. A. F. B. Ayub, N. F. A. B. M. Rasyidi, and M. A. B. Asbullah, "Self-invertible key on cipher polygraphic polyfunction with eigen matrix based imie," *Applied Mathematics and Computational Intelligence (AMCI)*, vol. 13, no. 3, pp. 13–25, 2024.

[9] C. Towse and E. Campbell, "Constructing integer matrices with integer eigenvalues," *Applied Probability Trust*, vol. 3, pp. 1–8, 2016.

[10] I. Yasser, F. Khalifa, M. A. Mohamed, and A. S. Samrah, "A new image encryption scheme based on hybrid chaotic maps," *Complexity Hindawi*, vol. 2020, pp. 1–23, 2020.

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

[12] R. M. Nguyen and M. S. Brown, "Why you should forget luminance conversion and do something better," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2017, pp. 6750–6758.

[13] Y. Rajput and A. Gulve, "A comparative performance analysis of an image encryption technique using extended hill cipher," *International Journal of Computer Applications*, vol. 95, no. 4, pp. 16–20, 2014.

[14] P. K. Naskar and A. Chaudhuri, "A secure symmetric image encryption based on bit-wise operation," *International Journal of Image, Graphics and Signal Processing*, vol. 6, no. 2, pp. 30–38, 2014.

[15] Wu, Yue, P. N. Joseph, and A. Sos, "Npcr and uaci randomness tests for image encryption," *Cyber Journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[16] C. Rajvir, S. Satapathy, S. Rajkumar, and L. Ramanathan, "Image encryption using modified elliptic curve cryptography and hill cipher," in *Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 1*. Springer, 2020, pp. 675–683.