



Menemui Matematik (Discovering Mathematics)

journal homepage: <https://persama.org.my/dismath/home>



Grayscale Image Encryption using Generalized Lucas Matrices and Hill Cipher

Putri Zalia Megat Johar¹, Faridah Yunos^{2*}, Ji-Jian Chin³ and Muhammad Asyraf Asbullah⁴

^{1,2}*Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang, Selangor, Malaysia*

^{2,4}*Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, Serdang, Selangor, Malaysia*

³*School of Engineering, Computing and Mathematics, Faculty of Science and Engineering, University of Plymouth, United Kingdom*

⁴*Malaysia Cryptology Technology and Management Centre, c/o Universiti Putra Malaysia, Serdang, Selangor, Malaysia*

⁴*Centre For Foundation Studies in Science of Universiti Putra Malaysia, Universiti Putra Malaysia, Serdang, Selangor, Malaysia*

¹putrizaliamj02@gmail.com, ^{2*}faridahy@upm.edu.my, ³ji-jian.chin@plymouth.ac.uk, ⁴ma_asyraf@upm.edu.my

*Corresponding author

Received: 2 October 2025

Accepted: 8 December 2025

ABSTRACT

Secure and effective methods of image encryption are increasingly important in protecting sensitive digital content from being attacked by today's increasingly knowledgeable and aggressive adversaries. This is no exception to attacks carried out on the secret content generated by grayscale images. Stemming from this problem, this research introduces a new approach to encrypt grayscale images using a combination of the Generalized Lucas Matrix and Hill Cipher (GLMHC) with a self-invertible encryption key. This type of encryption key eliminates the need to find the inversion key during the decryption process and reduces the computational burden. Encrypted image quality is measured using entropy, Mean Two-Dimensional Error (MSE), Signal to Peak Noise ratio (PSNR), Number of Pixels Change Rate (NPCR), and Average Integrated Change Intensity (UACI). This measure is compared with two other methods which is based on Elliptical Curve Cryptography (ECC) that also consider the scope of the self-invertible key matrix 4×4 dimension and using a 256×256 -pixel image. The results of the experiment showed that the GLMHC-based scheme achieved a standard value of entropy close to 8, indicating a strong randomness in the encryption process. The high MSE value of 8493.8882 compared to the lower PSNR of 8.8397 for certain images indicates the level of distortion in the decrypted image compared to the original one that has complied with the standards and even the quality is better than the other two methods. However, a UACI value of 29.7454% lower than the standard may imply less effective encryption, as certain pixels on the encrypted image still retain the same structure as the original image. This highlights the potential to improve the diffusion process in the certain area of image for future studies.

Keywords: Generalized Lucas Matrices, Hill Cipher, Self-invertible, Entropy, MSE, PSNR, NPCR, UACI

INTRODUCTION

In an era of rapidly evolving modern communication complexity, the security of transmitting sensitive information over secure networks is essential. Most of the data generated by digital images must be protected from unauthorized access and malicious attacks. To address these challenges, the implementation of effective cryptographic techniques is essential, as it can ensure the confidentiality, integrity, and authenticity of the transmitted data.

The term cryptographic comes from the Greek words 'Crypto', which means secret, and 'Graphein', which means writing. It involves the science and art of translating an understandable message into a message that cannot be interpreted in its original meaning and vice versa. This transformation relies on encryption and decryption algorithms. Its implementation uses two types of cryptography, namely symmetry and asymmetry. Symmetric cryptography uses a single shared key for encryption and decryption, while asymmetric cryptography uses a pair of keys consisting of a public key for encryption and a private key for decryption.

In today's modern world, communication systems that use grayscale images are developing extensively. The importance of protecting it from enemy interference and improving its security features should be a priority for cryptosystem developers. Domains such as medical imaging are heavily relying on formats such as MRI and X-ray scans, which contain confidential information that must be safeguarded through effective encryption techniques. Likewise, grayscale visuals from surveillance footage and biometric data, including facial recognition and fingerprint images, require secure handling to prevent privacy breaches and identity theft. These use cases highlight the need for encryption schemes specifically optimized for grayscale image structures.

Among the various cryptographic methods, HC proposed by Hill (1929) stands out as a classic symmetric encryption technique. The symmetric block cipher technique that has been introduced requires both the sender and receiver to use a similar private key, making the HC method a simple and fast algorithm. Its simplicity and computational efficiency make it a preferred choice despite its vulnerability to security breaches (Dawahdeh et al., 2018). However, its security system is weak due to its vulnerability to various cryptographic attacks. Therefore, several researchers have modified the algorithm to improve its security.

Acharya et al. (2007) addressed the challenge of invertibility in the HC algorithm, noting that the inability to find the inverse of the encryption matrix can prevent decryption. To avoid this issue, they introduced self-invertible matrices for key encryption, ensuring that the matrix used is inherently invertible, eliminating the need for inverse calculation during decryption. This method significantly reduces computational complexity, particularly in scenarios where matrix inversion is impractical. To further enhance security and practicality, Acharya et al. (2009) proposed involutory, permuted, and reiterative key matrix generation techniques. Involutory matrices eliminate the requirement for matrix inversion during decryption, since an involutory matrix is its own inverse. This innovation not only accelerates the decryption process, but also mitigates vulnerabilities associated with known plain text attacks. Acharya et al. (2010) extended the application of HC to biometric security by introducing an involutory key matrix with integer elements, addressing the limitations of the original HC and improving the encryption performance for biometric data.

Dawahdeh et al. (2018) introduced a novel technique called ECCHC, which is specifically designed for grayscale images. This cryptosystem eliminates the need to calculate the inverse of the encryption key, thus reducing the computational load compared to the original HC method. To

evaluate encryption efficiency, they used metrics such as entropy, PSNR, and UACI to compare the encrypted images with the original. Their research was carried out using MATLAB R2013a on a Core i5 computer. The findings showed that ECCHC offers greater security and better preservation of image quality compared to Naveen Kumar et al. (2012) and Panduranga and Naveen Kumar (2012).

Rajvir et al. (2020) proposed MECCHC, which also have a better encryption efficiency involving entropy, PSNR, and UACI rather than Naveen Kumar et al. (2012) and Panduranga and Naveen Kumar (2012). The encryption method was evaluated using security measures, excluding transmission channel loss, and showed no difference in intensity between the original and decrypted images. They found that MECCHC has high computational efficiency, as ECC strengthens weak keys of HC using the Discrete Logarithm Problem (DLP). The reduced computation time achieved using self-invertible matrices is a major factor in making MECCHC suitable for practical applications, such as processing Red, Green and Blue (RGB) images.

Based on previous research by Laoli et al. (2020), it was concluded that the HC algorithm is one of the more difficult symmetric cryptographic methods to break. This is because the HC uses matrix multiplication in its encryption and decryption processes, so it does not replace each identical letter in the plaintext with the same letter in the ciphertext. However, to improve security, researchers have explored hybrid approaches such as the combination of GLMHC cryptosystem (Prasad et al., 2022). This method is used generalized Lucas matrices (GLM) (Prasad and Mahato, 2022) as a secret key matrix in the HC cryptosystem. The GLM is derived from the Lucas sequence, which is similar to the Fibonacci sequence (Koshy, 2019) but has different initial values. The recursive sequence is then represented as an encryption key matrix in the Hill Cipher System. GLMHC was applied as a core component in cryptographic systems, significantly expanding the key space and reducing the time and space complexity involved in the generation of keys during cryptographic operations. The method relies on three key components: λ , s , and the shift vector B . The value λ , known only to authorized parties (Alice and Bob), ensures that the shift vector B , constructed based on λ , remains confidential. This setup improves security by preventing unauthorized access to λ .

Both studies by Dawahdeh et al. (2018) and Rajvir et al. (2020) employed 4×4 self-invertible key matrices generated using Acharya's method. This approach was not adopted in the study by Prasad et al. (2022). In line with those works, the technique proposed in our paper maintains a similar scope but differs in terms of the input matrix used for key generation. Unlike prior studies, our method introduces a matrix self-invertible generation mechanism, in which the input matrix is constructed using the Generalize Lucas Matrix. Also provides increased diffusion variance in the algorithm for encryption.

Despite the downtime and having certain security features when using GLMHC, we remain uncertain whether GLMHC can outperform existing encryption techniques in efficiency and performance, which involve grayscale images. While methods such as Elliptical Curve Cryptography over Prime Field and Hill Cipher (ECCHC) and its modification (MECCHC) have shown promising results, comparative evaluation with GLMHC is needed. The question arises whether GLMHC can provide better quality measures than ECCHC and MECCHC in grayscale image encryption or not?

This study presents a comprehensive evaluation of the proposed GLMHC encryption scheme by employing established quantitative metrics: entropy, Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Number of Pixels Change Rate (NPCR) and Unified Average Changing

Intensity (UACI) to assess its effectiveness in preserving image quality, enhancing security, and ensuring sensitivity to pixel variations. The evaluation compares the performance of GLMHC with ECCHC and MECCHC using the 256×256 pixel Einstein image as the primary reference and further validates its robustness through additional testing on standard grayscale images including Angry Bird, Baboon, and Cameraman.

The remainder of this paper is organized as follows: Section 2 presents the methodology of GLMHC and demonstrated its implementation. Section 3 gives the fundamentals of the image quality criteria followed by the performance comparisons with existing approaches. Section 4 concludes the paper.

RESEARCH METHODS

Hill Cipher (HC)

Hill (1929) introduced a symmetric block cipher technique known as the Hill Cipher. The key matrix used for encryption and decryption must share between the sender and receiver. The basic concept of this method consists of assigning a numerical value to each letter, with A = 0, B = 1, ..., Z = 25. The plain text message is subsequently divided into matrix blocks based on the size of the key matrix. For example, if the block size chosen is two ($P_{2 \times 1} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$), then the key matrix ($K_{2 \times 2} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$) must also be of size 2×2 . The encryption procedure then generates a cipher text block consisting of two numerical values ($C_{2 \times 1}$), as demonstrated in the following process:

$$C \equiv KP \equiv \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \equiv \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \pmod{26} \quad (1)$$

After that, the recipient of the message needs to decrypt the cipher text C back to the plain text P through the following transformation:

$$P \equiv K^{-1} \cdot C \equiv \frac{1}{\det(K)} \text{adj} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \pmod{26} \quad (2)$$

where K^{-1} is the inverse matrix of K and $\gcd(\det(K), 26) = 1$ (Agrawal and Gera, 2014 and Dawahdeh et al., 2018). By performing all arithmetic modulo the number of letters in the chosen alphabet, the cipher can be customized for use with alphabets of any size, extending beyond the standard modulo 26 (Yunos et al., 2023 and Yunus and Buhari, 2022).

Example 1. Suppose the key matrix is:

$$K_{4 \times 4} = \begin{bmatrix} 8 & 1 & 9 & 1 \\ 8 & 1 & 8 & 0 \\ 9 & 1 & 8 & 1 \\ 8 & 0 & 8 & 1 \end{bmatrix}.$$

Let the equivalent sequence numbers of plain text CRYPTOGRAPHY are 21724151914617015724 arranged to

$$P_{4 \times 3} = \begin{bmatrix} 2 & 17 & 24 \\ 15 & 19 & 14 \\ 6 & 17 & 0 \\ 15 & 7 & 24 \end{bmatrix}.$$

and encryption process will produce a block of cipher text by 12 numerical values:

$$C_{4 \times 3} \equiv \begin{bmatrix} 22 & 3 & 22 \\ 1 & 5 & 24 \\ 18 & 3 & 20 \\ 1 & 19 & 8 \end{bmatrix} \pmod{26}.$$

Then the alphabetical sequence that are equivalent to the numerical values are arranged as WDWBFYSDUBTI. To decrypt this cipher text, receiver need to use the following decryption formula

$$P \equiv K^{-1} \cdot C \equiv \begin{bmatrix} 22 & 7 & 23 & 7 \\ 4 & 19 & 4 & 18 \\ 23 & 7 & 22 & 7 \\ 4 & 18 & 4 & 19 \end{bmatrix} \begin{bmatrix} 22 & 3 & 22 \\ 1 & 5 & 24 \\ 18 & 3 & 20 \\ 1 & 19 & 8 \end{bmatrix} \pmod{26}$$

to get the original text.

Generation of Self-Invertible 4×4 matrix

A matrix does not always have an inverse. In the HC technique, decryption requires the inverse of the key matrix. If the chosen key matrix does not have an inverse, the encrypted message cannot be decrypted. To address this issue, Acharya et al. (2007) proposed a method to generate self-invertible matrices. This approach ensures that the key matrix always has an inverse, eliminating the need to compute it separately for HC decryption. The method can generate keys for square matrices of any dimension, ensuring a valid decryption process. But in our research, we only consider the generation of a self-invertible matrix for 4×4 as follows:

Theorem 1. A is called a self-invertible matrix if $A = A^{-1}$.

Let $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$ be a self-invertible matrix partitioned as $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ where $A_{11} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, $A_{12} = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}$, $A_{21} = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}$ and $A_{22} = \begin{bmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix}$. Then, to get

self-invertible of A we need to apply the following algorithm:

Algorithm 1 Generation of Self-invertible Matrix

Input: An arbitrary matrix A_{22} of size $\frac{n}{2} \times \frac{n}{2}$, scalar constant b

Output: A self-invertible matrix A of size $n \times n$

Computation:

1. Set $A_{11} \leftarrow -A_{22}$

2. Compute $A_{12} \leftarrow b(I \pm A_{11})$ // Choose sign based on design
3. Compute $A_{21} \leftarrow \frac{1}{b}(I \mp A_{11})$ // Choose sign based on design
4. Construct matrix, $A \leftarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$
5. Return (A)

Unlike ECCHC and MECCHC, our method introduces a matrix self-invertible generation mechanism with input matrix that implements the generalization Lucas matrix (see Definition 2).

Generalized Lucas Matrices (GLM)

The following are some definitions and theorems in Prasad et al. (2022) to facilitate this study.

Definition 1. A sequence $\{l_{k,n}\}$, akin to the generalized Fibonacci sequence of order $k \geq 2$ and integer $n \geq 0$. This sequence is governed by the recurrence relation:

$$l_{k,k+n} = l_{k,k+n-1} + l_{k,k+n-2} + \cdots + l_{k,n} \quad (3)$$

where the initial values $l_{k,n}$ are defined as follows:

$$l_{k,n} = f_{k,k+n-1} + 1f_{k,k+n-2} + 2f_{k,k+n-4} + \cdots + (k-2)f_{k,n} + (k-1)f_{k,n-1} \quad (4)$$

The sequence $\{l_{k,n}\}$ is defined with initial values $k, 1, 3, 7, 15, 31, 63, 127, 255, 511, \dots, 2^{k-1} - 1$ and is referred to as the generalized Lucas sequence of order k , where $k = 2$ corresponds to the standard Lucas sequence of order 2.

Definition 2. The generalized Lucas matrices $(L_k^{(n)})$ associated with $\{l_{k,n}\}$ is defined by

$$L_k^{(n)} = \begin{bmatrix} l_{k,k+n-1} & l_{k,k+n-2} + l_{k,k+n-3} + \cdots + l_{k,n} & \cdots & l_{k,k+n-2} \\ l_{k,k+n-2} & l_{k,k+n-3} + l_{k,k+n-4} + \cdots + l_{k,n-1} & \cdots & l_{k,k+n-3} \\ \vdots & \vdots & \ddots & \vdots \\ l_{k,k+n-(k-1)} & l_{k,n} + l_{k,n-1} + \cdots + l_{k,-k+n+2} & \cdots & l_{k,n} \\ l_{k,k+n-k} & l_{k,n-1} + l_{k,n-2} + \cdots + l_{k,-k+n+1} & \cdots & l_{k,n-1} \end{bmatrix} \quad (5)$$

The initial Lucas matrix $L_k^{(0)}$ for the order k is:

$$L_k^{(0)} = \begin{bmatrix} 2^{k-1} - 1 & 2^{k-1} & 2^{k-1} - k & \cdots & 7 \cdot 2^{k-4} & 3 \cdot 2^{k-3} & 2^{k-2} - 1 \\ 2^{k-2} - 1 & 2^{k-2} & 2^{k-2} + 1 & \cdots & 7 \cdot 2^{k-5} & 3 \cdot 2^{k-4} & 2^{k-3} - 1 \\ 2^{k-3} - 1 & 2^{k-3} & 2^{k-3} + 1 & \cdots & 7 \cdot 2^{k-6} & 3 \cdot 2^{k-5} & 2^{k-4} - 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & \cdots & k-2 & k-1 & k \\ k & 1-k & 2-k & \cdots & -3 & -2 & -1 \end{bmatrix} \quad (6)$$

Theorem 2. Determinant of generalized Lucas matrices is given by

$$\det(L_k^{(n)}) = \begin{cases} \det(L_k^{(0)}), & \text{if } k \text{ is odd} \\ (-1)^n \det(L_k^{(0)}), & \text{if } k \text{ is even} \end{cases} \quad (7)$$

Theorem 3. Suppose $L_k^{(n)}$ is a generalized Lucas matrix $GLM(k, n)$ and let $H = (L_k^{(0)})^2$ be invertible, then the inverse is given by

$$\text{Inv}(L_k^{(n)}) = L_k^{(-n)} \cdot H \quad (8)$$

Proposed Technique

This research has proposed a technique that combined GLM and HC. Let us assume that the message receiver, Bob, has a public key represented by $pk(p; E_1; E_2)$, which is generated by steps 1 to 5 in Algorithm 2. Next, in steps 6 to 8, Alice uses the public key pk and selects a specific integer e to create a secret key λ . Then, in step 9, she constructs the GLM matrix $L_\lambda^{(s)}$, which incorporates her signature. Then, use the created matrix to develop a self-invertible matrix (K) in step 10. Following Algorithm 3 and the HC technique, Alice sends encrypted images of a grayscale using KP . Unlike ECC and MECCHC, our method provides increased diffusion variance from steps 2 to 4. Upon receiving the encrypted message along with Alice's signature, Bob extracts the secret key λ and, after performing the necessary computations, successfully reconstructs the original image via Algorithm 4. The following algorithms outline this process.

Algorithm 2 Key Generation Procedure

Input: Selects a prime number p

Output: Public key (p, E_1, E_2) and private key (D, K)

Computation:

1. Receiver (Bob) selects a prime number p
2. Choose an integer D such that $1 < D < \phi(p)$
3. Select E_1 from the primitive roots modulo p
4. Compute $E_2 \leftarrow E_1^D \pmod{p}$
5. Bob's public key: $pk \leftarrow (p, E_1, E_2)$; secret key: $sk(D)$
6. Sender (Alice) selects an integer e such that $1 < e < \phi(p)$
7. Compute $s \leftarrow E_1^e \pmod{p}$
8. Compute $\lambda \leftarrow E_2^e \pmod{p}$ using Bob's E_2
9. Construct the recursive matrix $L_\lambda^{(s)} \pmod{p}$
10. Generate self-invertible matrix K from $L_\lambda^{(s)}$ using Algorithm 1
11. Return (p, E_1, E_2, D, K)

Algorithm 3 Image Encryption Procedure

Input: Read Green Blue (RGB) image with 256×256 pixel, self-invertible key matrix K with size 4×4

Output: Encrypted grayscale image

Computation:

1. Convert the RGB image to grayscale using the formula:

$$P \leftarrow 0.299R + 0.587G + 0.114B.$$

The use of this formula is very popular among previous researchers and has been studied in terms of its suitability compared to some other formulas (Nguyen and Brown, 2017). Specifically, let $R = [r_{i,j}]$, $G = [g_{i,j}]$ and $B = [b_{i,j}]$, for $i, j = 1, 2, \dots, 256$. The numerical value of grayscale image can be obtained by $P = [p_{i,j}]$ with $p_{i,j} \leftarrow 0.299r_{i,j} + 0.587g_{i,j} + 0.114b_{i,j}$.

2. Flatten the grayscale image values into a one-dimensional array:

$$P_{\text{flatten}} \leftarrow [p_{1,1}, p_{1,2}, \dots, p_{1,256}, p_{2,1}, p_{2,2}, \dots, p_{2,256}, \dots, p_{256,1}, p_{256,2}, \dots, p_{256,256}].$$

Set the index of all elements in P_{flatten} : index $1 \leftarrow p_{1,1}$ until index $65536 \leftarrow p_{256,256}$.

3. Generate a random permutation of the array indices: The *rng* function *rng*(12345) in MATLAB is used to control the random number generator, allowing us to set the seed and algorithm for producing random numbers. By using *rng*(12345), we initialize the random number generator with a specific seed, ensuring that the sequence of random numbers generated is repeatable and apply the permutation to shuffle grayscale values: $P_{\text{random permutation}} = [s_{i,1}]$ where $s_{i,1}$ for $i = 1, 2, \dots, 65536$ are random elements generate from P_{flatten} .
4. Re-shape the permuted array back to the original image dimensions:

$$P_{\text{reshape}} \leftarrow \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,256} \\ s_{1,257} & s_{1,258} & \dots & s_{1,512} \\ s_{1,513} & s_{1,514} & \dots & s_{1,768} \\ s_{1,769} & s_{1,770} & \dots & s_{1,1024} \\ \vdots & \vdots & \dots & \vdots \\ s_{1,65281} & s_{1,65282} & \dots & s_{1,65536} \end{bmatrix}$$

5. Divide the grayscale values of P_{reshape} into column vectors of size 4×1 :

$$P_1 = \begin{bmatrix} s_{1,1} \\ s_{1,257} \\ s_{1,513} \\ s_{1,769} \end{bmatrix}, P_2 = \begin{bmatrix} s_{1,2} \\ s_{1,258} \\ s_{1,514} \\ s_{1,770} \end{bmatrix}, \dots, P_{16384} = \begin{bmatrix} s_{1,64768} \\ s_{1,65024} \\ s_{1,65280} \\ s_{1,65536} \end{bmatrix}.$$

6. For each vector P_v for $v = 1, 2, \dots, 16384$, compute the encrypted vector:

$$C_v \leftarrow K \cdot P_v \pmod{256}.$$

7. Concatenate all C_v vectors to reconstruct the encrypted image values:

$$C \leftarrow \begin{bmatrix} C_1 & C_2 & \dots & C_{256} \\ C_{257} & C_{258} & \dots & C_{512} \\ \vdots & \vdots & \dots & \vdots \\ C_{16129} & C_{16130} & \dots & C_{16384} \end{bmatrix}.$$

8. Return to encrypted grayscale image

Algorithm 4 Image Decryption Procedure

Input: Encrypted image matrix with 256×256 pixel, self-invertible key matrix K with size 4×4

Output: Decrypted Image

Computation:

1. Split the encrypted image values into blocks of size 4×4
1. For each block C_v , compute the decrypted block $P_v \leftarrow K \cdot C_v \pmod{256}$
2. Concatenate all P_v vectors to form the decrypted grayscale image matrix
3. Flatten the reconstructed grayscale image into a one-dimensional array
4. Apply the inverse permutation to restore the original pixel order
5. Re-shape the array to match the original grayscale image dimensions
6. Return to decrypted image

Example of Implementation

Before applying the GLMHC encryption process, the first step is to create a self-invertible key matrix, known as the key generation process. Initially, a matrix 2×2 is generated using the GLM method. This matrix is then substituted into the formula proposed by Acharya et al. (2007) to produce a self-invertible square matrix.

Key Generation

1. Receiver (Bob) chooses a prime $p = 269$.
2. Select an integer $D = 31$ such that $1 < D < 268$.
3. Choose $E_1 = 102$ from the primitive roots of p .
4. Compute $E_2 \equiv E_1^D \equiv 102^{31} \equiv 88 \pmod{269}$.
5. Bob's public key is $pk(269, 102, 88)$, and the secret key is $sk(31)$.
6. Sender (Alice) selects an integer $e = 245$ such that $1 < e < 268$.
7. Compute $s \equiv E_1^e \equiv 102^{245} \equiv 147 \pmod{269}$.
8. Compute $\lambda \equiv E_2^e \equiv 88^{245} \equiv 2 \pmod{269}$.
9. Construct a matrix $2 \times 2, L_\lambda^{(s)} \pmod{269}$ as follows:

$$L_2^{(147)} \equiv \begin{bmatrix} l_{2,147+1} & l_{2,147} \\ l_{2,147} & l_{2,147-1} \end{bmatrix} \equiv \begin{bmatrix} 111 & 218 \\ 218 & 35 \end{bmatrix} \pmod{269}$$

where the entry elements are derived from the following Table 1:

Table 1: Lucas Sequence

| Index (n) | ... | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ... | 146 | 147 | 148 | ... |
|-----------------------------|-----|---|---|---|---|---|----|----|-----|------------|-----------|------------|-----|
| Lucas Seq. ($l_{2,n}$) | ... | 2 | 1 | 3 | 4 | 7 | 11 | 18 | ... | 325233 ... | 526239 .. | 851472 ... | ... |

10. Let $A_{22} = L_2^{(147)}$ and we obtain the self-invertible key as follows:

$$K = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 145 & 38 & 112 & 218 \\ 38 & 221 & 218 & 36 \\ 146 & 38 & 111 & 218 \\ 38 & 222 & 218 & 35 \end{bmatrix}$$

11. Return (p, E_1, E_2, D, K)

Encryption

1. Let the numerical values in RGB colors for each pixel of the original-colored image be represented as follows:

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

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

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

2. Convert the grayscale image into numerical values using:

$$P = 0.299R + 0.587G + 0.114B$$

As a result, the numerical representation of the grayscale image is given by:

$$P = \begin{bmatrix} 161 & 161 & 161 & \dots \\ 162 & 161 & 161 & \dots \\ 161 & 161 & 160 & \dots \\ 160 & 159 & 158 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

3. Flatten the grayscale values into a one-dimensional array:

$$P_{\text{flatten}} = [161, 161, 161, \dots, 162, 161, 161, \dots, 161, 161, 160, \dots, 159, 158, \dots]$$

4. Generate a random permutation of the array indices and apply a random permutation to shuffle the grayscale values. The permuted vector is:

$$P_{\text{random permutation}} = [185, 43, 124, \dots, 95, 220, 145, \dots, 56, 141, 142, \dots, 41, 39, 96, \dots]$$

5. Re-shape the permuted grayscale values to match the image dimensions:

$$P_{\text{reshape}} = \begin{bmatrix} 185 & 43 & 124 & \dots \\ 95 & 220 & 145 & \dots \\ 56 & 141 & 142 & \dots \\ 41 & 39 & 96 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

6. Divide the permuted grayscale values into 4×1 column matrices:

$$P_1 = \begin{bmatrix} 185 \\ 95 \\ 56 \\ 41 \end{bmatrix}, P_2 = \begin{bmatrix} 43 \\ 220 \\ 141 \\ 39 \end{bmatrix}, P_3 = \begin{bmatrix} 124 \\ 145 \\ 142 \\ 96 \end{bmatrix}, \dots$$

7. Multiply the matrix P_1 by the self-invertible key matrix:

$$C_1 \equiv KP_1 \equiv \begin{bmatrix} 145 & 38 & 112 & 218 \\ 38 & 221 & 218 & 36 \\ 146 & 38 & 111 & 218 \\ 38 & 222 & 218 & 35 \end{bmatrix} \begin{bmatrix} 185 \\ 95 \\ 56 \\ 41 \end{bmatrix} \equiv \begin{bmatrix} 77 \\ 237 \\ 206 \\ 35 \end{bmatrix} \pmod{256}$$

and repeat the same process for other blocks.

8. Concatenate the encrypted matrices to reconstruct the encrypted image. The numerical value of encrypted image is:

$$C = \begin{bmatrix} 77 & 233 & 162 & \dots \\ 237 & 220 & 1 & \dots \\ 206 & 135 & 144 & \dots \\ 35 & 145 & 50 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Return to encrypted grayscale image.

Decryption

1. Split the cipher value of the image pixel in blocks:

$$C_1 = \begin{bmatrix} 77 \\ 237 \\ 206 \\ 35 \end{bmatrix}, C_2 = \begin{bmatrix} 233 \\ 220 \\ 135 \\ 145 \end{bmatrix}, C_3 = \begin{bmatrix} 162 \\ 1 \\ 144 \\ 50 \end{bmatrix}, \dots$$

2. Multiply each block by the self-invertible key matrix:

$$P_1 \equiv K \cdot C_1 \equiv \begin{bmatrix} 145 & 38 & 112 & 218 \\ 38 & 221 & 218 & 36 \\ 146 & 38 & 111 & 218 \\ 38 & 222 & 218 & 35 \end{bmatrix} \begin{bmatrix} 77 \\ 237 \\ 206 \\ 35 \end{bmatrix} \equiv \begin{bmatrix} 185 \\ 95 \\ 56 \\ 41 \end{bmatrix} \pmod{256}$$

and repeat the similar process for other blocks.

3. Concatenate decrypted blocks to reconstruct the numerical value of grayscale image:

$$\begin{bmatrix} 185 & 43 & 124 & \dots \\ 95 & 220 & 145 & \dots \\ 56 & 141 & 142 & \dots \\ 41 & 39 & 96 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

4. Flatten the grayscale values into a one-dimensional array:

$$[185, 43, 124, \dots, 95, 220, 145, \dots, 56, 141, 142, \dots, 41, 39, 96, \dots]$$

5. Apply the inverse permutation to restore the original pixel order.
6. Re-shape the values into the original image dimensions to obtain the decrypted grayscale image:

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

7. Return to decrypted image.

RESULTS AND DISCUSSION

Fundamentals of the Image Quality Criteria

In this section, the fundamentals of image quality criteria are described in detail.

Entropy

One of the main statistical features used for assessing image encryption is Entropy. It measures the level of unpredictability in an image by analyzing the probability distribution of pixel values. The efficiency of an encrypted image is higher since the entropy value is close to eight for the 256×256 grayscale image, which is considered the theoretical ideal number. According to Dawahdeh et al. (2018) and Pandey and Sharma (2025), entropy is a measure of how difficult it is to crack a cryptosystem. The entropy of an image is calculated using the following formula:

$$\text{Entropy} = \sum_{x=0}^{255} \left[P(x) \times \log_2 \frac{1}{P(x)} \right] \quad (9)$$

where $P(x)$ represents the probability of an intensity of pixel x , where $x \in \{0,1,2, \dots, 255\}$ determined as:

$$P(x) = \frac{\text{Frequency of pixel value } x}{\text{Total number of pixels in the image}} \quad (10)$$

Peak Signal to Noise Ratio (PSNR)

One of the metrics used to evaluate how effective an image encryption algorithm is PSNR. Quantifies the level of distortion in the decrypted image compared to the original, providing information on the quality of encryption. A higher PSNR value indicates less data loss in the decrypted image, which means that it is more like the original, as stated in Rajput and Gulve (2014). This illustrates how effective the encryption method is. To calculate PSNR, the following formula is applied:

$$\text{PSNR} = 20 \times \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right) \quad (11)$$

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

$$\text{MSE} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (A_{ij} - B_{ij})^2 \quad (12)$$

where H, W, A_{ij} and B_{ij} represent the height, the width, the pixel value of the original and the pixel value of the decrypted images, respectively (Panduranga and Naveen Kumar, 2012). If MSE increases while comparing between the encrypted and the original images, then PSNR decreases, suggesting that the encrypted image is more random (Naveen Kumar et al., 2012). An effective encryption method results from a high MSE and a low PSNR value, which shows that the two images are different and not identical (Naskar and Chaudhuri, 2014).

Number of Pixels Change Rate (NPCR)

NPCR serves as a measure for evaluating the sensitivity of an encryption algorithm to slight changes in the plain image. It measures how significantly the encrypted output changes when the input (plain image) is altered by just one pixel. The high value of NPCR indicates a strong defense against differential attacks, as a minor change in the original image results in a significant change in the encrypted image (Wu et al., 2011). The NPCR can be calculated with the following formula:

$$\text{NPCR} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W D(i, j) \times 100\% \quad (13)$$

where H and W denote the height and width of the image, respectively. The binary value of $D(i, j)$ is determined by comparing the pixel values of the two cipher images, $A(i, j)$ and $B(i, j)$, at position (i, j) :

$$D(i, j) = \begin{cases} 0 & \text{if } A(i, j) = B(i, j) \\ 1 & \text{if } A(i, j) \neq B(i, j) \end{cases} \quad (14)$$

For an 8-bit grayscale image, the theoretical ideal value for NPCR is 99.6094% (Wu et al. 2011). An NPCR value close to this ideal percentage suggests that the encryption algorithm exhibits excellent diffusion properties, making it highly resistant to differential cryptanalysis.

Unified Average Changing Intensity (UACI)

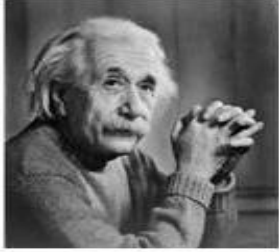
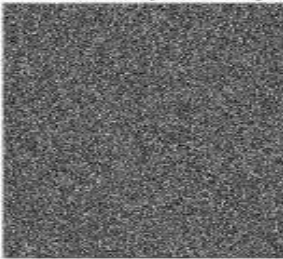
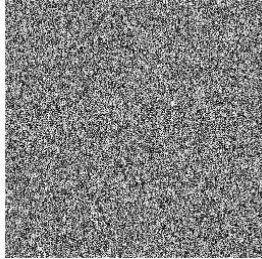
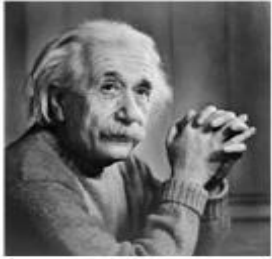
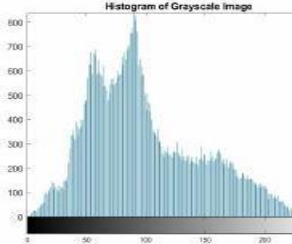
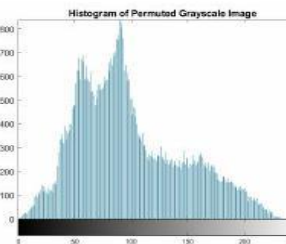
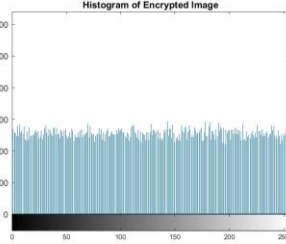
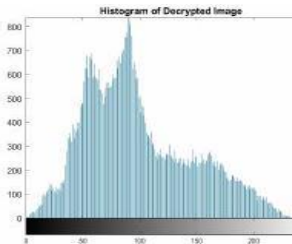
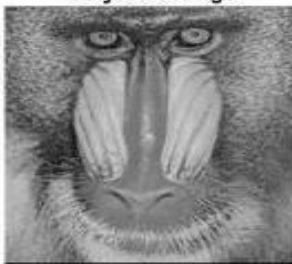
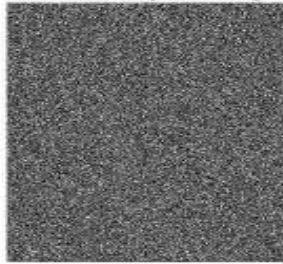
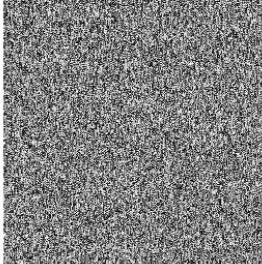
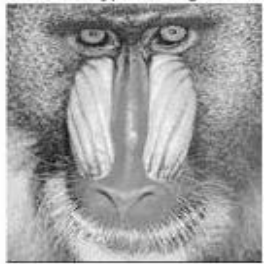
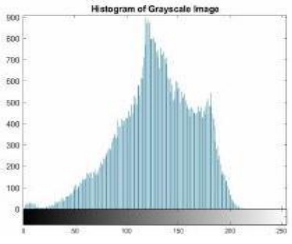
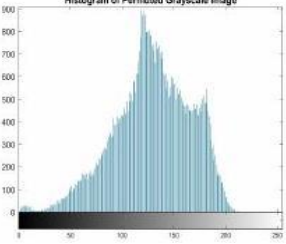
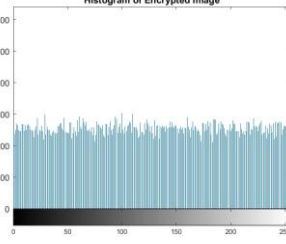
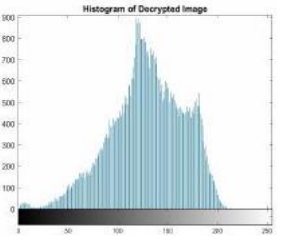

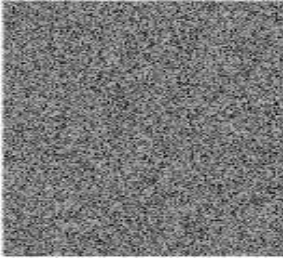
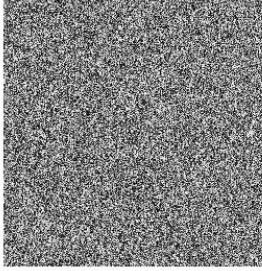

UACI calculates the difference between the plain image and the ciphered image. It is used to evaluate the strength of the encryption method. The average intensity difference between the original and ciphered pictures is measured by UACI. The highest UACI indicates resistance to differential attacks for the suggested approach. Dawahdeh et al. (2018) stated that the size and format of the image will determine its value. UACI is computed for the grayscale image of dimensions 256×256 using the following formula:

$$UACI = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{|A(i, j) - B(i, j)|}{255} \times 100 \quad (15)$$

where 255 represents the highest value of a pixel and $A(i, j)$ denotes the pixel value in the original image and $B(i, j)$ indicates the pixel value in the encrypted image (Panduranga and Naveen Kumar, 2012). The expected value of UACI for a 256×256 image is 33.46% (Dawahdeh et al., 2018). The value of UACI greater than 33.46% indicates that the encryption algorithm is highly sensitive to minor changes in the plain text image, leading to significant differences in the cipher texts. Although this suggests strong diffusion properties, excessively high UACI values might imply over-randomization, potentially affecting image quality or decryption accuracy (Wu et al., 2011). A UACI value lower than 33.46% suggests that the encryption algorithm has weaker sensitivity to small changes, resulting in less variation between the cipher texts. This could indicate inadequate diffusion, making the encryption scheme more susceptible to differential attacks (Wu et al., 2011).

Performance Comparisons with the Existing Approaches

The encryption and decryption process of the GLMHC technique running on MATLAB R2023b (9.14.0.2182215) 64-bit software, executed on a Core i7 computer with a CPU of 2.30 GHz and 8 GB of RAM. The output that reads and displays the image of Einstein, Baboon, Cameraman, and Angry Bird is shown in Figure 1.

| Grayscale Image | Permuted Grayscale Image | Encrypted Image | Decrypted Image |
|--|---|---|--|
| (a) Einstein | | | |
| Grayscale Image  | Permuted Grayscale Image  | Encrypted Image  | Decrypted Image  |
| Histogram of Grayscale Image  | Histogram of Permuted Grayscale Image  | Histogram of Encrypted Image  | Histogram of Decrypted Image  |
| (b) Baboon | | | |
| Grayscale Image  | Permuted Grayscale Image  | Encrypted Image  | Decrypted Image  |
| Histogram of Grayscale Image  | Histogram of Permuted Grayscale Image  | Histogram of Encrypted Image  | Histogram of Decrypted Image  |
| (c) Cameraman | | | |
| Grayscale Image  | Permuted Grayscale Image  | Encrypted Image  | Decrypted Image  |

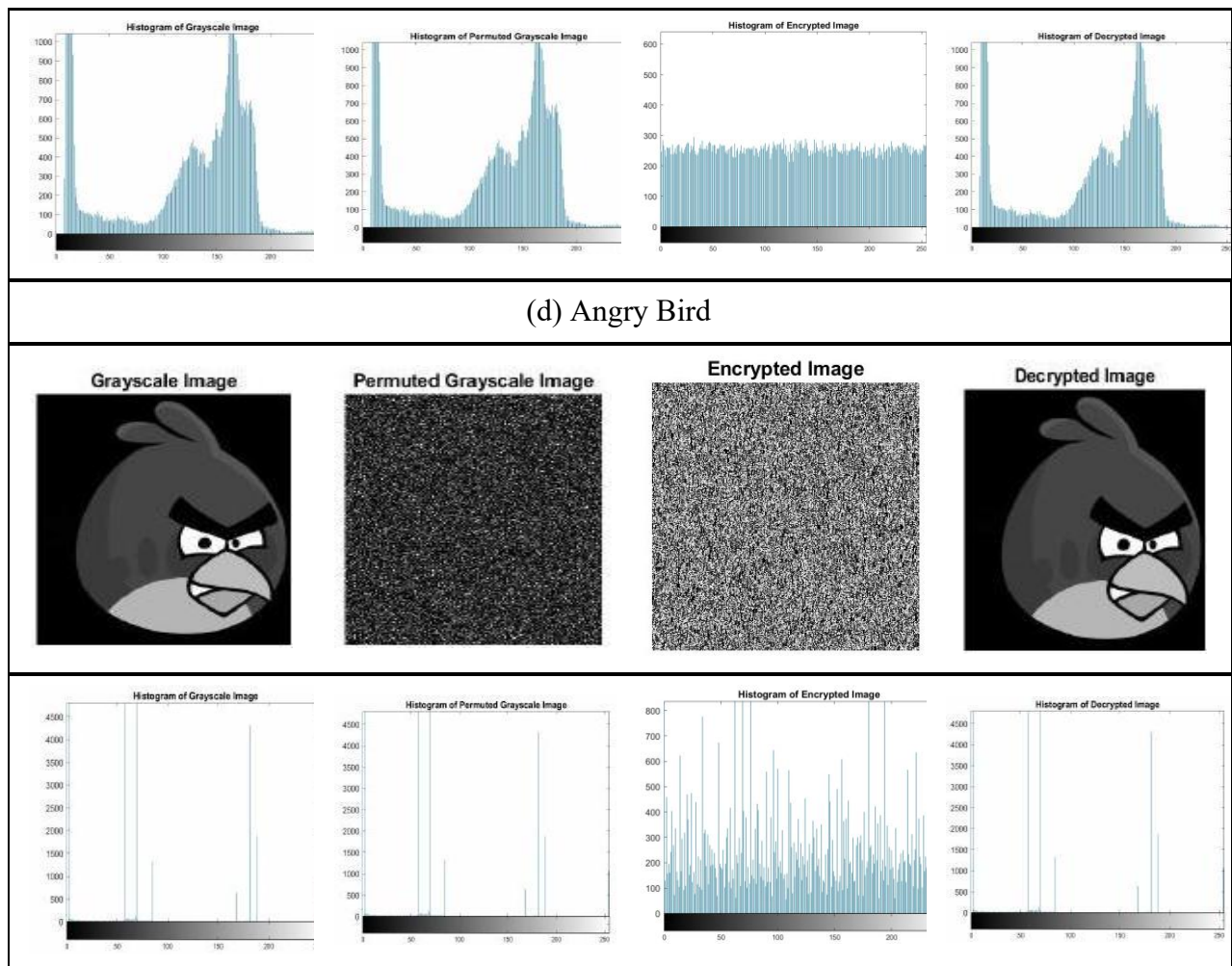


Figure 1: Grayscale, Permuted Grayscale, Encrypted, and Decrypted Images for Einstein, Baboon, Cameraman, and Angry Bird with their Histograms

Table 2 presents a comparison of image quality for the GLMHC encryption method, specifically for the Einstein image, in relation to previous techniques. From this table, all the methods achieve a good entropy, which is very close to 8. This suggests that the encryption process introduces a high level of randomness in the pixel distribution. Although GLMHC exhibits a good entropy value, this alone does not necessarily imply superior security compared to other encryption techniques. The effectiveness of an encryption method must be assessed using some criteria, including MSE, PSNR, NPCR and UACI. In terms of MSE and PSNR, GLMHC is a less effective encryption system compared to MECCHC because MSE decreases from 8630.184 to 8493.8882 while PSNR increases from 8.7706 to 8.8397, resulting in an image that is less random than the original, making it easier to visually attack. Although the encrypted Einstein image using GLMHC is sufficiently different from the original, there is room for improvement in reducing the resemblance further to enhance encryption quality.

The NPCR value for GLMHC, which is 99.5712%, is very close to the ideal value of 99.6094%. In comparison, MECCHC records a lower NPCR of 97.8989%. This suggests that the GLMHC encryption algorithm demonstrates excellent diffusion properties, as even a slight change in the original image results in significant pixel variation in the encrypted output.

GLMHC achieves an UACI value of 29.7454%, which is less than the expected value. This suggests that while the encryption method introduces substantial pixel differences, it may not be as effective in maximizing pixel diffusion as some other encryption techniques. An ideal encryption technique should ensure that the encrypted image exhibits significant unpredictability to resist statistical attacks.

Table 2: Image Quality Measurements for Einstein Image

| Method | Entropy | MSE | PSNR | NPCR% | UACI |
|-------------------------------|---------|-----------|--------|---------|---------|
| ECCHC (Dawahdeh et al., 2018) | 7.9899 | N/A | 9.7483 | N/A | 26.9087 |
| MECCHC (Rajvir et al., 2020) | 7.9954 | 8630.184 | 8.7706 | 97.8989 | 29.7502 |
| Proposed GLMHC Technique | 7.9977 | 8493.8882 | 8.8397 | 99.5712 | 29.7454 |

The comparative quality of different encrypted images using GLMHC is shown in Table 3.

Table 3: Image Quality Measurements for Several Images when Using GLMHC

| Image | Entropy | MSE | PSNR | NPCR% | UACI |
|------------|---------|------------|--------|---------|---------|
| Einstein | 7.9977 | 8493.8882 | 8.8397 | 99.5712 | 29.7454 |
| Baboon | 7.9967 | 6774.9171 | 9.8218 | 99.5941 | 27.0815 |
| Cameraman | 7.9894 | 6774.9171 | 8.0803 | 99.5941 | 32.2278 |
| Angry Bird | 7.6674 | 15190.0985 | 6.3152 | 98.1445 | 39.7138 |

Table 3 presents the encryption performance of GLMHC in different test images. Their entropy values are close to 8, indicating a strong randomness in the encryption process. However, the entropy for the Angry Bird image is not close to this standard value. This is because the Angry Bird image contains many repeated sharp colors. Since entropy measures the randomness in an image, having large areas of the same color reduces the unpredictability. This makes it harder to achieve high randomness in encryption, as there is less variation in pixel values to work with. The PSNR value of the Baboon image is the highest among all images. The higher PSNR value during decryption indicates that the data loss in the decrypted image is minimal. This signifies that the decrypted image matches the original image, which determines the higher-quality encryption method. The NPCR values for all the images are close to 99.6094% unless for Angry Bird. The UACI values for all images, except for Angry Bird, are lower than 33.46%, indicating less drastic changes in the pixels. A lower UACI can mean less effective encryption, as some parts of the original image might retain structural similarity, indicating room for improvement in diffusion. However, the Angry Bird image exhibits a higher UACI value of 39.7138% but has a relatively low entropy of 7.6674, indicating a lower randomness and suggesting that encryption for this image may not be as effective in maintaining statistical unpredictability. Similarly, the Baboon image achieves an entropy of 7.9967 but has a UACI of 27.0815%, which is lower than the expected value, indicating room for improvement in diffusion.

CONCLUSION

The results presented in Table 2 indicate that the GLMHC method achieves an ideal entropy value, demonstrating the effectiveness of the encryption technique in introducing randomness into the image. The higher MSE and lower PSNR values compared to ECCHC reflect the encryption quality and the level of distortion in the decrypted image relative to the original, suggesting that the proposed technique is more efficient. Furthermore, the NPCR value for GLMHC surpasses that of MECCHC, indicating that the encryption algorithm exhibits excellent diffusion properties, making it highly resistant to differential cryptanalysis. However, the UACI values reveal that the method does not consistently achieve the ideal value, suggesting that the diffusion process in the encrypted images could be further improved. Table 3 presents strong results for certain images, further supporting the efficiency of the proposed technique.

The use of image quality metrics like entropy, MSE, PSNR, NPCR, and UACI shows the effectiveness and quality of the image encryption while also strengthening its reliability for secure image transmission. This study demonstrates that the GLMHC algorithm can yield a standard entropy value and low PSNR, indicating a good level of encryption. However, there are some limitations to consider. In practical cases, images are frequently compressed (e.g., in JPEG format) and transmitted across communication channels that are susceptible to errors. The integrity of the authenticated image and the efficacy of encryption algorithms can be compromised by both factors. Consequently, more research is suggested to assess how well GLMHCs perform in settings that mimic lossy compression and transmission interference. Incorporating error correction mechanisms or adaptations to compression formats can enhance the practicality of these algorithms in fields like telemedicine, security surveillance, and biometric image transmission.

Moreover, other significant elements to investigate are key sensitivity, histogram analysis, correlation coefficients, resistance to differential attacks, and evaluation of time complexity. The investigations will yield a more comprehensive understanding of both the resilience and efficiency of the image transmission technique that relies on the GLMHC method.

While the proposed scheme demonstrates encouraging statistical properties for image encryption, an important limitation lies in its reliance on the Hill cipher as the core cryptographic primitive. Despite the enhancements provided by the generalized Lucas matrix, the Hill cipher remains fundamentally linear and vulnerable to known-plaintext and chosen-plaintext attacks. This means construction does not provide semantic security, a standard requirement in modern cryptography, and therefore cannot be considered secure against adversaries with realistic capabilities.

To address this, future work should extend the current design by integrating the generalized Lucas matrix with a formally secure authenticated encryption (AE) scheme such as AES-GCM or AES-SIV. In such a hybrid construction, the Lucas-based permutation and masking steps would serve as pre-processing layers to improve diffusion and reduce visible redundancies in image data, while the AES component would ensure provable confidentiality and integrity. This combination

would allow the system to retain the favourable entropy and NPCR values observed in this study while also satisfying widely accepted cryptographic security definitions.

Looking forward, this line of research will focus on developing and validating practical implementations of these hybrid schemes. Specific directions include building efficient key derivation mechanisms to tie the Lucas-based transformations to AES keys, testing the impact on computational overhead for large-scale image datasets, and formally analysing resistance to adaptive attacks. By doing so, the work can progress from demonstrating statistical robustness toward a secure, standards-compliant encryption system suitable for deployment in real-world multimedia applications.

In summary, this work introduced the generalized Lucas matrix as a novel enhancement to the Hill cipher, demonstrating its potential to generate self-invertible key matrices and achieve strong statistical properties in image encryption. Through experimental evaluation, the scheme achieved high entropy, strong resistance to differential attacks as reflected in NPCR and UACI values and improved visual obfuscation of image data. These results highlight the effectiveness of the generalized Lucas matrix in strengthening diffusion and confusion processes, and provide a solid foundation upon which more secure, formally proven encryption frameworks can be developed.

ACKNOWLEDGMENT

As authors, we acknowledge the support from PUTRA GRANT GP/2023/9753100 the Special Postgraduate Scheme (SGRA) provided by Universiti Putra Malaysia.

REFERENCES

- Acharya, B., Rath, G. S., Patra, S. K. and Panigrahy, S. K. (2007), Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm. *International Journal of Security*, **1(1)**:14–21.
- Acharya, B., Patra, S. K. and Panda, G. (2009), Involutory, Permuted and Reiterative Key Matrix Generation Methods for Hill Cipher System, *International Journal of Recent Trends in Engineering*, **1(4)**:106–108.
- Acharya, B., Sharma, M. D., Tiwari, S. and Minz, V. K. (2010), Privacy Protection Of Biometric Traits using Modified Hill Cipher with Involutory Key and Robust Cryptosystem. In *Procedia Computer Science*, vol. 2, pp. 242-247.
- Agrawal, K. and Gera, A. (2014), Elliptic Curve Cryptography with Hill Cipher Generation for Secure Text Cryptosystem. *International Journal of Computer Applications*, **106(1)**: 18–24.

- Dawahdeh, Z., Yaakob, S. N. and Othman, R. R. (2018), A New Image Encryption Technique Combining Elliptic Curve Cryptosystem with Hill Cipher. *Journal of King Saud University –Computer and Information Sciences*, **30(3)**: 349–355.
- Hill, L. S. (1929), Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly*, **36(6)**:306–312.
- Koshy, T. (2019), *Fibonacci and Lucas Numbers with Applications*. Hoboken, New Jersey: John Wiley & Sons.
- Laoli, D., Sinaga, B. and Sinaga, R. J. (2020), Penerapan Algoritma Hill Cipher dan Least Significant Bit (LSB) untuk Pengamanan Pesan pada Citra Digital. *JISKA (Jurnal Informatika Sunan Kalijaga)*, **4(3)**: 1–11.
- Naskar, P. K. and Chaudhuri, A. (2014), A Secure Symmetric Image Encryption based on Bit-Wise Operation. *International Journal of Image, Graphics and Signal Processing*, **6(2)**: 30–38.
- Naveen Kumar, S. K., Sharath Kumar, H. S. and Panduranga, H.T. (2012), Encryption Approach for Images Using Bits Rotation Reversal and Extended Hill Cipher Techniques. *International Journal of Computer Applications*, **59(16)**: 10–14.
- Nguyen, R. M. and Brown, M. S. (2017), Why You Should Forget Luminance Conversion and Do Something Better. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 21–26, pp. 6750-6758.
- Pandey, K. and Sharma, D. (2025), Novel Image Encryption Algorithm Utilizing Hybrid Chaotic Maps and Elliptic Curve Cryptography with Genetic Algorithm,” *Journal of Information Security and Applications*, 89: 1–12.
- Panduranga, H. T. and Naveen Kumar, S. K. (2012), Advanced Partial Image Encryption using Two-Stage Hill Cipher Technique. *International Journal of Computer Applications*, **60(16)**: 14–19.
- Prasad, K., Mahato, H. and Kumari, M. (2022), A Novel Public Key Cryptography based on Generalized Lucas Matrices. *arXiv preprint arXiv:2202.08156*.
- Prasad, K. and Mahato, H. (2022), Cryptography using Generalized Fibonacci Matrices with Affine-Hill Cipher. *Journal of Discrete Mathematical Sciences and Cryptography*, **25(8)**: 2341–2352.
- Rajput, Y. and Gulve, A. K. (2014), A Comparative Performance Analysis of an Image Encryption Technique using Extended Hill Cipher. *International Journal of Computer Applications*, **95(4)**: 16–20.
- Rajvir, C., Satapathy, S., Soundrapandiyam, R. and Lakshmanan, R. (2020), Image Encryption using Modified Elliptic Curve Cryptography and Hill Cipher: In *Smart Intelligent Computing and Applications*. In *Proceedings of the Third International Conference on Smart Computing and Informatics*, Springer, vol. 1, pp. 675-683.

- Wu, Y., Noonan, J. P. and Agaian, S. (2011), NPCR and UACI Randomness Tests for Image Encryption. *Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT)*, 31–38.
- Yunos, F. and Buhari, M. N. A. (2022), Gabungan Kriptografi Lengkuk Eliptik dan Saifer Hill dalam Perutuan Imej Berskalar Samar. *Jurnal Asas Ilmu Matematik*, **1(4)**: 68–81.
- Yunos, F., Jamaludin, S. and Basri, W. (2023), Solution of $L^2 = A$ Matrix To Generate Involutory Matrices for Cipher Trigraphic Polyfunction. *Appliedtoathematics and Computational Intelligence*, **12 (1)**: 70–86.