Civitas Proceedings
Pancasila and Civic Education Program Study
Faculty of Education and Teacher Training
Universitas Nusantara PGRI Kediri, Indonesia
Volume 1 2026

Entropy of Encrypted Grayscale Image Via ECCBHC

Siti Nuraisah Roslan[1], Faridah Yunos [*,2], Nurfarrah Syazleen Abdul Halim[2]
[1, 2] Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor
[2] Department of Mathematic and Statistics, Universiti Putra Malaysia, 43400 Serdang, Selangor
faridahy@upm.edu.my
* Corresponding Author

Abstract: Elliptic Curve Cryptography (ECC) is a complex asymmetric key encryption method, while Hill Cipher (HC) employs a simpler symmetric encryption approach. A previous study found that a self-invertible key matrix (SI), which is built up from a domain of elliptical curve parameters over a prime field, has been implemented while sending a grayscale image through the combination system of ECC and HC. This paper focuses on evaluating image encryption algorithms by analyzing the entropy of encrypted grayscale images while utilizing an elliptic curve over a binary field to replace the role of the previously used elliptic curve, particularly during the key generation process. As a result, the transmission of grayscale images through this new technique achieves an entropy of approximately 3.9998.

Keywords: Image encryption; self-invertible key; entropy; Elliptical Curve Cryptography; encryption; Hill Cipher.

INTRODUCTION

Cryptography, derived from the Greek words "Kryptos" (secret) and "Graphein" (writing) (Rathidevi et al., 2017), emerged as an important field around 1400. Over the next 450 years, it was primarily characterized by the use of a "nomenclator," where each letter in a message was substituted with another letter based on a predefined substitution table, effectively concealing the message's meaning (plaintext) (Davies, 1997). At its core, cryptography involves the transformation of intelligible messages (plaintext) into unintelligible counterparts (ciphertext) using encryption algorithms. These encrypted messages can then be deciphered back into their original form using decryption algorithms, provided that the appropriate keys are used. This technique was used for various purposes, including securing sensitive information and ensuring confidentiality.

Cryptography can be classified into two primary categories: symmetric and asymmetric cryptographic keys. Symmetric cryptography employs a single shared secret key for both encrypting the original text and decrypting the ciphertext, which may undergo simple transformations to derive the encryption key from the decryption key. This category further divides into classical (including transposition and substitution ciphers) and modern (such as stream and block ciphers). An example of a symmetric key scheme is the HC, which performs linear transformations on a message column represented as an integer vector of dimension m in Z_m (Hill, 1929). On the other hand, asymmetric cryptography utilizes two distinct keys: a private key and a publicly accessible public key, which are intricately connected. This approach provides enhanced security and represents a notable advancement in cryptography. One notable example of asymmetric cryptography is Elliptic Curve Cryptography (ECC) (Hankerson et al., 2006).

According to Christensen (2006), in the HC, the numerical form of the original text (plaintext) is often represented as a matrix denoted by P, while the matrix K is chosen as the secret key and C represents the ciphertext. The encryption process involves multiplying the plaintext matrix P by the secret key matrix K and then taking the result in modulo N, where N is a positive integer, resulting in a ciphertext represented as C. The encryption expression from P to C can be expressed as $C \equiv KP \pmod N$. However, the decryption from C to P is carried out as $P \equiv K^{-1}C \pmod N$ where K^{-1} represents the inverse of K modulo N. The HC algorithm has a simple structure and allows fast calculations if K^{-1} can be easily obtained.

This HC algorithm has a simple structure and fast calculations, particularly when the inverse of the main key, K, can be easily obtained. However, if the main key lacks an inverse, the decryption process

ISSN 3164-7081 (online) DOI https://doi.org/10.29407/qvyv3t25

becomes more challenging as the key dimensions increase. Typically, elementary row operations and the concept of multiplication inverses in modulo arithmetic are employed to obtain the inverse of a matrix. To circumvent this, SI matrices are utilized as HC's secret key, eliminating the need to find the inverse key before decryption. Acharya et al. (2007) have presented various methods to generate such SI keys for any dimension. This approach is also adopted in the modified HC system, as seen in the works of Acharya et al. (2009), Acharya et al. (2010), Dey, (2012), Naveenkumar dan Panduranga (2013), Dawahdeh et al. (2018), Rajvir et al. (2020).

The HC algorithm has a relatively low security level because both the sender and the receiver must use and share the same private key over a potentially insecure channel. To enhance its security, Acharya et al. (2009) proposed an SI key generation method, which enables the creation of distinct key patterns for each data encryption block. Furthermore, Acharya et al. (2010) proposed a technique from Sastry dan Shankar (2008) to obtain biometric features using a modified HC using SI keys and robust cryptosystems. This solves the weakness of conventional HC by using iteration and interlacing.

Encryption techniques provide an ideal way to protect images from attackers. Dey, (2012) recommends SDAEI for image encryption, using a modified HC technique with a SI matrix. This matrix is generated using the same password as the previous encryption, strengthening the security. Naveenkumar dan Panduranga (2013) introduced a two-level HC method, using square blocks and SI matrices to control the pixel encryption rates. They employed the Latin Square Image Cipher technique to generate matrix blocks and compared it with the four-level HC to improve camera efficiency and expand the application possibilities.

In Dawahdeh et al. (2018), an image encryption technique known as ECC with HC (ECCHC) was proposed to improve security and efficiency against hackers by transforming HC from symmetric to asymmetric encryption method. This method utilizes the SI key matrix, derived from the parameters of the elliptic curve represented as $E_p: y^2 \equiv x^3 + ax + b \pmod{p}$ over the prime field F_p, for the generation of secret encryption and decryption keys. Rajvir et al. (2020) further improved image security by introducing Modified Elliptic Curve Cryptography with HC (MECCHC), specifically adapted for grayscale images. The analysis proves that the computation time for image encryption and decryption is faster than that of the ECCHC technique. The evaluation of efficiency using entropy, PSNR and UACI shows the same effect as ECCHC.

In another development, Yunos dan Buhari (2022) and Yunos et al. (2023) refined ECCHC by replacing prime-field elliptic curves with binary-field elliptic curves, marking a significant advance especially during the key generation process, resulting in the ECCBHC system. Rajvir et al. (2020) emphasize the role of ECC's in securing image transfer over the website. ECCBHC inherits these benefits and enhances its applicability to image security. ECCBHC, like ECC, offers space efficient cryptography with enhanced security by providing high security features with smaller key size and less memory and power consumption (Dawahdeh et al., 2018). However, a comprehensive analysis of its efficiency, including metrics such as entropy, PSNR, UACI, and runtime tracking, is yet to be conducted to assess its effectiveness.

Recognizing this gap, this research aims to conduct a critical evaluation of ECCBHC through an analysis of its entropy values. Rarher than framing the findings as a performance improvement, this work actually provides a realistic insight into its security capabilities and image quality. Entropy is a statistical measure that is used to assess the effectiveness of image encryption, mainly applied to grayscale images. Quantifies the level of uncertainty or randomness within an encrypted image, providing insight into security and image quality.

The paper is structured as follows. Section 1 covers the implementation of the SI matrix in HC, its modifications and advantages, along with an analysis of image encryption algorithms through entropy in encrypted grayscale images. Section 2 provides an overview of the conventional HC, the algorithm to generate self-invertible keys, the elliptic curve operation, and the entropy formula used in this study. Section 3 introduces a cryptosystem that combines ECC based on an elliptic curve over a binary field with HC and includes an entropy evaluation briefing. Section 4 presents an illustrative implementation example, and Section 5 focuses on image encryption analysis through entropy measures. Finally, the concluding remarks are presented in the last chapter.

PRELIMINARIES

2.1. Hill Cipher Algorithm

HC is a symmetric block cipher technique introduced by the mathematician Lester Hill in 1929. The sender and receiver use and exchange the same private key to perform the encryption and decryption process. This method offers a straightforward structure and quick computations (Dawahdeh et al., 2018). To implement HC, the original image or plaintext must first be converted to numerical values. The plaintext is then arranged into a matrix, P depending on the size of the secret key. The core concept of this method involves assigning numerical values to letters, such as $A = 0$, $B = 1, C = 2, \ldots Z = 25$.

For instance, if the secret key size is 3×3, the sequence of numbers from the original text is organized as $P_{3\times4}$ in size, that is $K_{3\times3} = \begin{bmatrix} 9 & 2 & 7 \\ 15 & 8 & 1 \\ 3 & 6 & 4 \end{bmatrix}$, then the corresponding number sequence of the original text M A T H E M A T I C A L is 120019070412001908020011 should be arranged to $P_{3\times4} = \begin{bmatrix} 12 & 00 & 19 & 07 \\ 04 & 12 & 00 & 19 \\ 08 & 02 & 00 & 11 \end{bmatrix}$, and the encryption process $C_{3\times4} \equiv K_{3\times3} P_{3\times4} (\mathrm{mod}\ 26)$ will produce a block of cipher text with twelve numerical values in the form of $\begin{bmatrix} 16 & 12 & 15 & 22 \\ 12 & 14 & 25 & 08 \\ 14 & 02 & 05 & 23 \end{bmatrix}$. The alphabet sequence corresponding to it is then arranged as Q M P W M O Z I O C F X. To decrypt this message, the receiver needs to calculate the inverse matrix of $K_{3\times3}$ such that $K_{3\times3} \cdot K_{3\times3}^{-1} = I$ where I is the identity matrix, then use equation $P_{3\times4} \equiv K_{3\times3}^{-1} \cdot C_{3\times4} (\mathrm{mod}\ 26)$ to obtain the original message $P_{3\times4}$.

2.2. Generation of Self-invertible Matrix

This method was introduced by Acharya et al. (2007). Let us say $K_3 = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$ where $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$. Thus, the value of the other partition in K_3 is obtained by solving $K_{12} = I - K_{11}$, $K_{21} = I + K_{11}$ and $K_{11} + K_{22} = 0$, where I is an identity matrix. Alternatively, the following algorithm can be applied to determine SI:

Algorithm 1. Generation of Self-invertible Matrix
Input: An arbitrary matrix K_{22} of size 2×2, scalar constant k
Output: A self-invertible matrix K_3 of size 4×4
Computation:

(1) Set $K_{11} \leftarrow -K_{22}$
(2) Compute $K_{12} \leftarrow k\,(I \pm K_{11})$ / / Choose sign based on design
(3) Compute $K_{21} \leftarrow \frac{1}{k}(I \mp K_{11})$ / / Choose sign based on design
(4) Construct matrix, $K_3 \leftarrow \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$
(5) Return (K_3)

2.3. Elliptic Curve Operation

ECC is an encryption technique suitable for use in mobile devices and embedded systems, as it can provide high-security features with smaller key sizes and lower memory and power consumption (Dawahdeh et al., 2018). The security and efficiency of ECC are derived from the mathematical operations performed on elliptic curve points, which are elaborated in the following:

The non-supersingular curve E over the binary field $\mathbb{F}_{2^m}$ is defined by the equation:

$$E: y^2 + xy \equiv x^3 + ax^2 + b \pmod{f(z)}$$

and exhibits some of the following properties:

(1) Identity: If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then $P + \infty = \infty + P = P$ for all $P \in E(\mathbb{F}_{2^m})$.

(2) Negative: If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then $(x, y) + (x, x + y) = \infty$. The point $(x, x + y)$ denoted by $-P$ and called negative P; Note that the point $-P$ is an element of $E(\mathbb{F}_{2^m})$. Also, $-\infty = \infty$.

(3) Point addition: Let $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$ and $Q = (x_2, y_2) \in E(\mathbb{F}_{2^m})$, where $P = \pm Q$. Next, $P + Q = (x_3, y_3)$, with

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2}$$
$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$

and

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$

(4) Point doubling: Let $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$, where $P = -P$. Therefore $2P = (x_3, y_3)$, with

$$\lambda = x_1 + \frac{y_1}{x_1}$$
$$x_3 = \lambda^2 + \lambda + a = x_1^2 + \frac{b}{x_1^2}$$

and

$$y_3 = x_1^2 + \lambda x_3 + x_3$$

(5) Scalar multiplication: The integer scalar k for the point $Q = (x_1, y_1)$ is located on the curve E by repeating the addition of the point Q to itself as many as k times, resulting in a point R that also lies on the same curve, E.

$$R = kQ = Q + Q + \cdots + Q$$

For example, the calculation of $15Q$ can be done using point addition and point multiplication as follows:

$$15Q = 2(2(2Q + Q) + Q) + Q$$

2.4. Entropy

Entropy is one of the statistical scalar parameters to evaluate the effectiveness of image encryption. This analysis aims to measure the level of random information in the encrypted image compared to the original image to determine the performance of the proposed system approach. The theoretical and ideal entropy value for the grayscale image of size 256×256 is eight, and the efficiency of the encrypted image is better if the entropy value is close to eight (Panduranga & Naveenkumar, 2012). The following formula is used to calculate the entropy:

$$\text{Entropy} = \sum_{x=0}^{255} \left[P(x) \cdot \log_2 \frac{1}{P(x)} \right] \tag{2.1}$$

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

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

METHODS

This section introduces the proposed key generation, encryption and decryption algorithms, as well as the entropy evaluation framework. All simulations were conducted using MATLAB R2023a (64-bit, version 9.14.0.2306882) on a system equipped with an AMD Ryzen 53500 U processor (2.10 GHz) and 8.00 GB of RAM. To demonstrate the system's implementation, this study considers a communication scenario in which a sender (User A) transmits an image to a receiver (User B). Both parties must agree on the elliptical curve E and establish a shared domain defined by Both parties must agree on the elliptic curve E and establish a shared domain defined by $a, b, f(z), G$, where a and b represent the coefficients of $E, f(z)$ is an irreducible polynomial, and G is a generator point. Subsequently, each party selects a private key, which must be a random value less than degrees of $f(z)$. Each party then chooses a private key n_A for Users A and n_B for User B to generate their respective public keys, P_A and P_B i.e.

$$P_A = n_A \cdot G \text{ and } P_B = n_B \cdot G$$

Each user multiplies his private key with the other user's public key to get the initial key, $K_I = (x, y)$ through

$$K_I = n_A \cdot P_B = n_B \cdot P_A = n_A \cdot n_B \cdot G = (x, y)$$

To generate the secret key matrix, K_3, one must first compute

$$K_1 = x \cdot G = (k_{11}, k_{12}) \text{ and } K_2 = y \cdot G = (k_{21}, k_{22}).$$

To address the issue where this matrix is not always invertible, the SI key matrix is defined as $K_3 = K_3^{-1}$. This allows for the sane key to be used for both encryption and decryption. This method is implemented on a 256×256 pixel grayscale image, segmented into four-pixel blocks, with each user creating a 4×4 key K_3 via Algorithm 1.

The encryption process involves segmenting the image into blocks of four, with each block represented as a 4×1 vector ($P_1, P_2, P_3, \ldots$). For every block, calculate the cipher vector ($C_1, C_2, C_3, \ldots$) by multiplying the vector by K_3 and applying modulo $f(z)$. One block is processed, reconstructing the resulting cipher image (C) for transmission to another party. The following steps are repeated for every block in the image:

$$\text{Let } P_1 = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \end{bmatrix} \text{ then } C_1 \equiv K_3 \cdot P_1 \equiv \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \end{bmatrix} \equiv \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \\ c_{41} \end{bmatrix} (\mathrm{mod}\ f(z)).$$

Upon receipt of the cipher image, the decryption process begins by partitioning the pixel data into blocks of four. Each block is formatted as a 4×1 column vector. The original plaintext vector $(P_1, P_2, P_3, \ldots)$ is recovered by multiplying the cipher vector by the decryption key under modulo $f(z)$ operation. Reassembling these vectors yields the original image. This proposed hybrid approach, which integrates ECC over a binary field and Hill Cipher, is designated as ECCBHC:

Algorithm 2. Key Generation Procedure by User A

User A, as the sender of the message must

(1) select a private key, n_A satisfying the condition that $\deg(n_A) < \deg(f(z))$.

(2) calculate the public key $P_A = n_A \cdot G$

(3) calculate initial keys $K_I = n_A \cdot P_B = (x, y)$

(4) calculate $K_1 = x \cdot G = (k_{11}, k_{12})$ and $K_2 = y \cdot G = (k_{21}, k_{22})$

(5) generate a SI key K_3 using Algorithm 1.

Algorithm 3. Key Generation Procedure by User B

User B, as the recipient of the message has to

(1) select a private key, n_B satisfying the condition that $deg(n_B) < deg\big(f(z)\big)$

(2) calculate the public key $P_B = n_B \cdot G$

(3) calculate initial keys $K_I = n_B \cdot P_A = (x, y)$

(4) calculate $K_1 = x \cdot G = (k_{11}, k_{12})$ and $K_2 = y \cdot G = (k_{21}, k_{22})$
(5) generate a SI key K_3 using Algorithm 1

Algorithm 4. Encryption by User A

Input: Read Green Blue (RGB) image with 256×256 pixel, self-invertible key matrix K_3 with size 4×4, irreducible polynomial $f(z)$
Output: Encrypted grayscale image and its entropy value
Computation:

(1) Convert the input RGB image to grayscale using the formula:

$$P = 0.299R + 0.587G + 0.114B \tag{3.1}$$

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 & 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 $P = [p_{i,j}]$ where $p_{i,j} = 0.299r_{i,j} + 0.587g_{i,j} + 0.114b_{i,j}$

(2) Convert $p_{i,j}$ to the polynomial over binary field F_{2^m}.
(3) Reduce each of those polynomials in modulo $f(z)$.
(4) Split the pixel value of P into four-sized blocks:

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

(5) Multiply the SI key K_3 by each vector P_i for $i = 1,2, \dots, 16384$ to get a value

$$C_i \equiv K_3 \cdot P_i \pmod{f(z)}.$$

(6) Convert C_i for $i = 1,2, \dots, 16384$ to numerical value.
(7) Concatenate all C_i vectors to reconstruct the encrypted image values:

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

(8) Assign each entry element of C_{ECCBHC} to $c_{i,j}$ for $i, j = 1,2, \dots, 256$.

(9) C_{ECCBHC} is flattened into a one-dimensional byte.

$$C_{ECCBHC} = [c_{1,1}, c_{2,1}, \dots, c_{256,1}, c_{2,1}, c_{2,2}, \dots, c_{2,256}, \dots, c_{256,1}, c_{256,2}, \dots, c_{256,256}].$$

(10) Find entropy value of C_{ECCBHC} from Step 9 using Equation (2.1).

(11) Return to encrypted grayscale image and its entropy.

Algorithm 5. Decryption by User B
Input: Encrypted grayscale image
Output: Original Image
Computation:

(1) Split the pixel value of the original image into four-sized blocks
(2) Arrange each block into column vectors with four rows (4×1)
(3) Multiply the SI key K_3 by each vector ($C_1, C_2, C_3, \dots$) to get a value $P_i \equiv K_3 \cdot C_i \ (mod\ f(z))$
(4) Building a cipher image C from the corresponding values in the cipher vector $(P_1, P_2, P_3, \dots)$

The step change listed above occurs in Algorithm 2 (Step 1), Algorithm 3 (Step 1), Algorithm 4 and Algorithm 5 (Step 3) when compared to Dawahdeh et al. (2018). This is due to the replacement of the curve E_p with E. Matrix K_{11} that generates a square matrix K_3 is the result of several steps of scalar multiplication via the curve E. Previous studies have proven that secrecy n_A and n_B are very difficult to disassemble due to discrete log problems in the execution of ECC. Hence, confidentiality K_{11} is also guaranteed. Like previous researchers, the ECCBHC system still retains its main advantages during the implementation of encryption and decryption with the Hill Cipher technique, which does not require an inverse for K_3.

EXAMPLE OF IMPLEMENTATION

Assume that User A wants to send image M to User B and agrees to use an elliptical curve $E: y^2 + xy \equiv x^3 + z^3x^2 + (z^3 + 1) \bmod (z^4 + z + 1)$ over binary field $\mathbb{F}_{2^4}$. Let $a = z^3$ and $b = z^3 + 1$. One of the selected points from E is $G = (z^3, 1)$ is a generator point or a base point (Hankerson et al., 2006). Therefore, the domain $\{a, b, f(z), G\}$ for E is $\{z^3, z^3 + 1, z^4 + z + 1, (z^3, 1)\}$. Before User A sends a grayscale image, the first thing that needs to do is to get a numerical value that matches the red, green, blue (RGB) colors for each pixel for color images, for example, Angry Birds with size 256×256 pixel in Figure 1. The objects we chose for this example are unrealistic/real compared to the images of Lena, Baboon, Cameraman, Einstein and others that were commonly used as comparisons by previous researchers.

Figure 1. Color Image of Angry Birds for 256×256 pixel

The following are some of the numerical values in RGB corresponds to Figure 1. Each matrix represents one color from the red (R), green (G), and blue (B), respectively.

$$R = \begin{bmatrix} 184 & 184 & 196 & \dots \\ 184 & 184 & 208 & \dots \\ 184 & 190 & 213 & \dots \\ 184 & 201 & 213 & \dots \\ 185 & 211 & 213 & \dots \\ 193 & 213 & 213 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}, \quad B = \begin{bmatrix} 27 & 27 & 34 & \dots \\ 27 & 27 & 41 & \dots \\ 27 & 31 & 44 & \dots \\ 27 & 37 & 44 & \dots \\ 27 & 43 & 44 & \dots \\ 32 & 44 & 44 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.1}$$

Key Generation

User A as a message sender

(1) chooses a secret key, $n_A = z^2 + 1$

(2) calculate the public key $P_A = n_A \cdot G = (z^3 + z + 1, z)$

(3) calculate the initial key $K_I = n_A \cdot P_B = (z^3 + z^2, z^3 + z^2) = (x, y)$

(4) calculate $K_1 = x \cdot G = z^3(z^3, 1) + z^2(z^3, 1) = (z^3, 1) = (k_{11}, k_{12})$ and

$$K_2 = y \cdot G = z^3(z^3, 1) + z^2(z^3, 1) = (z^3, 1) = (k_{21}, k_{22})$$

(5) use $K_{11} = \begin{bmatrix} z^3 & 1 \\ z^3 & 1 \end{bmatrix}$ to generate a SI key

$$K_3 = \begin{bmatrix} z^3 & 1 & z^3 + 1 & 1 \\ z^3 & 1 & z^3 & 0 \\ z^3 + 1 & 1 & z^3 & 1 \\ z^3 & 0 & z^3 & 1 \end{bmatrix}$$

User B as the recipient of the message

(1) choose a secret key, $n_B = z^2 + z + 1$.

(2) calculate the public key $P_B = n_B \cdot G = (z^3 + z + 1, z^3 + 1)$

(3) calculate the initial key $K_I = n_B \cdot P_A = (z^3 + z^2, z^3 + z^2) = (x, y)$

(4) calculate $K_1 = x \cdot G = z^3(z^3, 1) + z^2(z^3, 1) = (z^3, 1) = (k_{11}, k_{12})$ and

$$K_2 = y \cdot G = z^3(z^3, 1) + z^2(z^3, 1) = (z^3, 1) = (k_{21}, k_{22})$$

(5) use $K_{11} = \begin{bmatrix} z^3 & 1 \\ z^3 & 1 \end{bmatrix}$ to generate SI key

$$K_3 = \begin{bmatrix} z^3 & 1 & z^3+1 & 1 \\ z^3 & 1 & z^3 & 0 \\ z^3+1 & 1 & z^3 & 1 \\ z^3 & 0 & z^3 & 1 \end{bmatrix}$$

Encryption by User A

(1) Map the entries elements in matrix (4.1) to corresponding number of grayscale image through formula $P = 0.299R + 0.587G + 0.114B$.

$$P = \begin{bmatrix} 58 & 58 & 62 & \dots \\ 58 & 58 & 67 & \dots \\ 58 & 60 & 69 & \dots \\ 58 & 64 & 69 & \dots \\ 58 & 68 & 69 & \dots \\ 61 & 69 & 69 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.4}$$

(2) Followed by changing the matrix (4.4) to the polynomial form as follows:

$$\begin{bmatrix} z^5+z^4+z^3+z & z^5+z^4+z^3+z & z^5+z^4+z^3+z^2+z & \dots \\ z^5+z^4+z^3+z & z^5+z^4+z^3+z & z^6+z+1 & \dots \\ z^5+z^4+z^3+z & z^5+z^4+x^3+z^z & z^6+z^2+1 & \dots \\ z^5+z^4+z^3+z & z^6 & z^6+z^2+1 & \dots \\ z^5+z^4+z^3+z & z^6+z^2 & z^6+z^2+1 & \dots \\ z^5+z^4+z^3+z^2+1 & z^6+z^2+1 & z^6+z^2+1 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.5}$$

(3) Next, reduce each of those polynomials in modulo $(z^4 + z + 1)$ i.e.

$$\begin{bmatrix} z^3+z^2+z+1 & z^3+z^2+z+1 & z^3+z+1 & \dots \\ z^3+z^2+z+1 & z^3+z^2+z+1 & z^3+z^2+z+1 & \dots \\ z^3+z^2+z+1 & z^3+1 & z^3+1 & \dots \\ z^3+z^2+z+1 & z^3+z^2 & z^3+1 & \dots \\ z^3+z^2+z+1 & z^3 & z^3+1 & \dots \\ z^3 & z^3+1 & z^3+1 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.6}$$

(4) Split the matrix (4.6) be a block $P_1, P_2, P_3, \dots$ with size 4×1 for each block:

$$P_1 = \begin{bmatrix} z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \end{bmatrix}, P_2 = \begin{bmatrix} z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+1 \\ z^3+z^2 \end{bmatrix}, P_3 = \begin{bmatrix} z^3+z+1 \\ z^3+z^2+z+1 \\ z^3+1 \\ z^3+1 \end{bmatrix}, \dots$$

(5) Multiplication K_3 to vector P_1 will produce a cipher vector C_1:

$$C_1 \equiv K_3 \cdot P_1 \equiv \begin{bmatrix} z^3 & 1 & z^3+1 & 1 \\ z^3 & 1 & z^3 & 0 \\ z^3+1 & 1 & z^3 & 1 \\ z^3 & 0 & z^3 & 1 \end{bmatrix} \begin{bmatrix} z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \end{bmatrix} \equiv \begin{bmatrix} z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \\ z^3+z^2+z+1 \end{bmatrix} \bmod (z^4+z+1)$$

and the same process is repeated to acquire the other blocks. $C_2, C_3, \dots$.

(6) Convert C_i for $i = 1,2, \dots, 16384$ to numerical value.

(7) Concatenate all C_i vectors to reconstruct the encrypted image values:

$$C_{ECCBHC} = \begin{bmatrix} 5 & 15 & 12 & \dots \\ 15 & 10 & 12 & \dots \\ 15 & 9 & 14 & \dots \\ 15 & 9 & 10 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.7}$$

(8) Assign each entry element of C_{ECCBHC} to $c_{i,j}$ for $i, j = 1,2, \dots, 256$.

(9) C_{ECCBHC} is flattened into a one-dimensional byte.

$$C_{ECCBHC} = [5{,}15{,}15, 15, \dots, c_{256,1}, 15{,}10{,}9, 9, \dots, c_{2,256}, \dots, c_{256,1}, c_{256,2}, \dots, c_{256,256}].$$

(10) Find entropy value of C_{ECCBHC} from Step 9 using Equation (2.1).

(11) Return to encrypted grayscale image with entropy 2.7920.

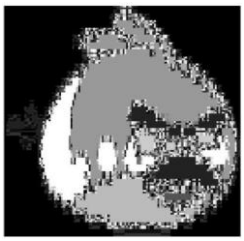


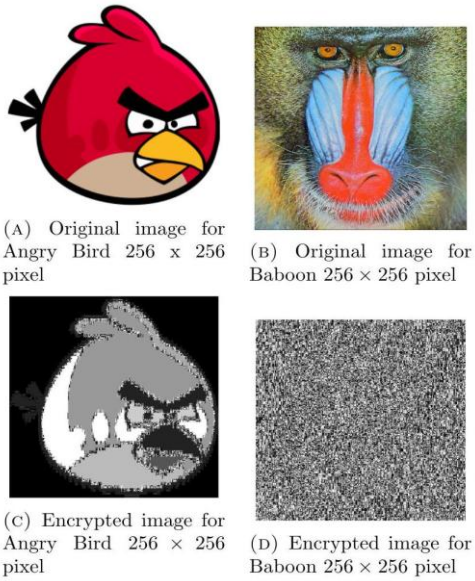
Figure 2. Encrypted Image of Angry Birds for 256×256 pixel

Decryption by User B

While the decryption process is a fundamental component of the proposed system, it was not the focus of this analysis, as the entropy values we calculated solely based on the generated encryption images.

EXPERIMENTAL RESULTS

The experimental results from encryption process are shown in Figure 3.



(A) Original image for Angry Bird 256 x 256 pixel

(B) Original image for Baboon 256×256 pixel

(C) Encrypted image for Angry Bird 256×256 pixel

(D) Encrypted image for Baboon 256×256 pixel

Figure 3. The original image and Encrypted image for Angry Bird and Baboon

A comparison between the proposed technique and some other techniques for the Baboon grayscale image is shown in Table 2. It is clear that the entropy value (3.9998) in the proposed technique is lower than the ECCHC techniques from Dawahdeh et al. (2018) and the MECCHC technique from Rajvir et al. (2020), that raises concerns about its robustness against attacks. The ECCBHC process applied to encrypted images of Angry Bird and Baboon deviates from the ideal entropy value of eight. The suboptimal entropy observed in the ECCBHC system likely stems from the use of a low-degree irreducible polynomial, specifically $z^4 + z + 1$, which restricts the variety of limits and output states. Furthermore, the algorithm's fails to comprehensively comprehensively disrupt spatial pixel correlations (the dependency relationships between adjacent pixel values), thereby preserving detectable residual patterns of the

original structure. Consequently, the encryption process falls short of attaining an optimal level of randomness.

Table 2. Comparison of ECCBHC with Existing Methods over Entropy Measures

Methods	Entropy
Dawadeh et al. (2018)	7.9968
Chintan et al. (2020)	7.9971
Proposed Method	3.9998

The proposed technique is tested on other images, and the results are summarized in Table 3.

Table 3. Entropy measures for certain images

Images	Entropy
Angry Bird	2.7920
Einstein	3.9981
Peppers	3.9995
Baboon	3.9998

This scenario raises a critical question regarding AI capability to what extent can an AI model, acting as a passive adversary, reconstruct original image without prior knowledge of the cryptographic system employed? The potential for AI to function as a passive attacker by identifying original images hidden within ciphertexts without access to the encryption keys is a significant concern. Figure 4 shows that AI Mode in Google can interpret images if the underlying shape (e.g., the 'Angry Bird') remains partially visible. In contrast, complex encryption involving high variation in color and tone, such as the 'Baboon' image currently successfully thwarts AI recognition. However, given the rapid advancement of AI, it is foreseeable that future models could be trained to bypass these cryptographic layers by learning to identify original images directly from ciphertext.

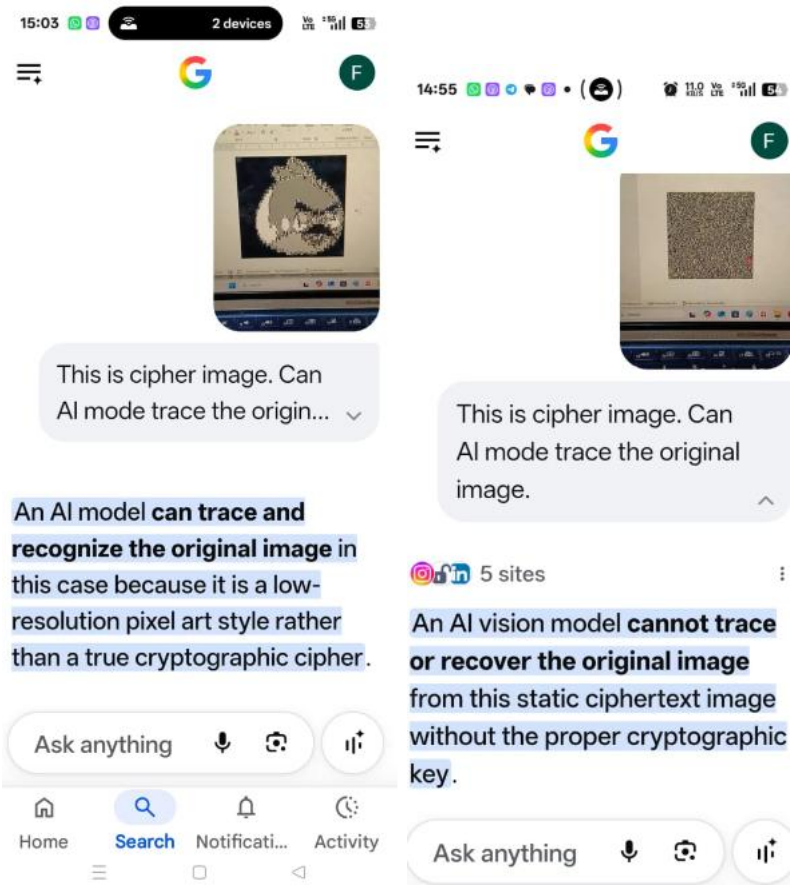


Figure 4. AI Mode capability in recognizing encrypted content across different cipher images

CONCLUSION

In conclusion, we have successfully examined the entropy values of grayscale images encrypted using ECCBHC. Our findings indicate that ECCBHC utilizing a lower-degree irreducible polynomial may not provide the expected level of security and randomness preservation, as it deviates significantly from the theoretical entropy value of eight. This constraint stems from remaining spatial pixel correlations and residual patterns, which potentially expose the cipher image to the risk of passive pattern recognition by AI models. Consequently, future research should explore the utilization of higher-degree irreducible polynomials to eliminate structural pixel dependencies and enhance cryptographic randomness. Furthermore, the evaluation framework should be expanded to incorporate comprehensive auditing metrics, such as the Histogram analysis, Number of Pixels Change Rate (NPCR) and Peak Signal-to-Noise Ratio (PSNR) and running time tracking to optimize real-time deployment efficiency.

REFERENCES

Rathidevi, M., Yaminipriya, R., & Sudha, S. V. (2017). Trends of cryptography stepping from ancient to modern. In *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)* (pp. 1-9). IEEE.

Davies, D. (1997). A brief history of cryptography. *Information Security Technical Report*, *2*(2), 14-17.

Hill, L. S. (1929). Cryptography in an algebraic alphabet. *The American Mathematical Monthly*, 36(6), 306-312. https://doi.org/10.1080/00029890.1929.11986963

Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer.

Christensen, C. (2006). Cryptography of the Vigenère cipher. In *Proceedings of Computer Sciences Corporation* (pp. 1–18).

Acharya, B., Rath, G., Patra, S., & 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., & 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., & Minz, V. K. (2010). Privacy protection of biometric traits using modified Hill cipher with involutory key and robust cryptosystem. *Procedia Computer Science, 2*, 242-247. https://doi.org/10.1016/j.procs.2010.11.031

Dey, S. (2012). SD-AEI: An advanced encryption technique for images. In *2012 Second International Conference on Digital Information Processing and Communications (ICDIPC)* (pp. 68-73). IEEE.

Naveenkumar, S. K., & Panduranga, H. T. (2013). Partial image encryption for smart camera. In *Proceedings of the 2013 IEEE International Conference on Recent Trends in Information Technology (ICRTIT)* (pp. 126-132). IEEE.

Dawahdeh, Z. E., Yaakob, S. N., & 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.

Rajvir, C., Satapathy, S., Rajkumar, S., & Ramanathan, L. (2020). Image encryption using modified elliptic curve cryptography and Hill cipher. In *Smart intelligent computing and applications* (pp. 675-683). Springer.

Sastry, V. U. K., & Shankar, N. R. (2008). Modified Hill cipher for a large block of plaintext with interlacing and iteration. *Journal of Computer Science, 4*(1), 15-20.

Yunos, F., & Buhari, M. N. A. (2022). Gabungan kriptografi lengkuk eliptik dan saifer Hill dalam perutusan imej berskala samar. *Jurnal Asas Ilmu Matematik, 1*(4), 68–81.

Panduranga, H. T., & Naveenkumar, S. K. (2012). Advanced partial image encryption using two-stage Hill cipher technique. *International Journal of Computer Applications, 60*(16), 14–19.

Yunos, F., Buhari, M. N. A., & Muslim, N. (2023). ECCBHC system while sending grayscale image. In *Proceedings of the 9th International MARDIN ARTUKLU Scientific Research Conference* (pp. 164–173).

Nguyen, R. M. H., & 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 (CVPR)* (pp. 6750–6758).