Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040629
Article

Asymmetric Key Cryptosystem for Image Encryption by Elliptic Curve over Galois Field *GF* (2^n)

Mohammad Mazyad Hazzazi[1], Hafeez Ur Rehman[2,*], Tariq Shah[2] and Hajra Younas[2]

[1]Department of Mathematics, College of Science, King Khalid University, Abha, 61421, Saudi Arabia
[2]Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan
*Corresponding Author: Hafeez Ur Rehman. Email: hrehman@math.qau.edu.pk
Received: 26 March 2023; Accepted: 29 May 2023; Published: 30 August 2023

Abstract: Protecting the integrity and secrecy of digital data transmitted through the internet is a growing problem. In this paper, we introduce an asymmetric key algorithm for specifically processing images with larger bit values. To overcome the separate flaws of elliptic curve cryptography (ECC) and the Hill cipher (HC), we present an approach to picture encryption by combining these two encryption approaches. In addition, to strengthen our scheme, the group laws are defined over the rational points of a given elliptic curve (EC) over a Galois field (GF). The exclusive-or (XOR) function is used instead of matrix multiplication to encrypt and decrypt the data which also refutes the need for the inverse of the key matrix. By integrating the inverse function on the pixels of the image, we have improved system security and have a wider key space. Furthermore, through comprehensive analysis of the proposed scheme with different available analyses and standard attacks, it is confirmed that our proposed scheme provides improved speed, security, and efficiency.

Keywords: Elliptic curve; Galois field; group law; hill cipher

1 Introduction

Communication technology, multimedia data technology, and Internet protocol communication via wireless networks have experienced rapid growth in this decade. However, the transmission of sensitive information over an open wireless network like the internet poses a security risk. Therefore, it is crucial to develop new methods that can guarantee the confidentiality of sensitive information transmitted over such networks. It takes a variety of encryption methods, including substitution box (S-box) and chaotic maps in all three dimensions, to protect the security of data being transmitted via communication channels. The three main techniques for securing information—steganography, watermarking, and encryption—are used to hide data from unauthorized readers. Additionally, there are two categories of key cryptosystems in cryptography: symmetric (also known as a private key) and asymmetric (also known as the public key). One key is used in a symmetric key cryptosystem to perform both encryption and decryption. The private key used by each user in an asymmetric

cryptosystem, in contrast, is distinct from the private key used by the other user. In recent years, digital information technology and multimedia data have rapidly evolved, and security has become a crucial factor in sharing confidential information. Utilising standard-based cryptosystems and using photos as a base is one trustworthy way to accomplish this. Images are frequently employed for ordinary or unique actions in personal, institutional, military, medical, and other contexts where they are required. Hence, to prevent cyber-attacks, the images should be protected from attackers [1]. Several algorithms have been developed to guarantee secure image transmission, including the image encryption scheme based on the two-dimensional (2D) Salomon map, a two-dimensional hyperchaotic system using optimization benchmark functions, and the 2D eπ-map which is a 2D chaotic map based on Euler and Pi numbers [2–5] by Erkan et al. Moreover, various algorithms have been proposed for encrypting and decrypting digital audios and images [6–9] by Gao et al. In these algorithms, the sender converts the original data to an enciphered data before transmitting it to the other user over the internet (receiver) and the receiver decrypts the ciphered data and restores it to its original state. Also, ECC was suggested separately by [10,11] as an example of an effective public-key cryptography technique [12] because ECC has the advantages of small key size, fast computation, and better security [13]. Furthermore, ECC is promoted because the same security level is possible with shorter keys, less computation, and less memory usage. ECC's 160-bit key is as a 1024-bit Rivest-Shamir-Adleman (RSA) key but provides ten times better speed when using a 128-bit key. Furthermore, ECC uses mathematical problems that are harder to solve than those used by RSA, making it more difficult for attackers to break the encryption. ECC is also simpler and uses smaller keys, which require less storage without sacrificing security. Binary extension fields have been suggested as an alternative to prime fields for ECC, which can reduce the amount of computation needed [14].

1.1 Related Work

The HC technique is one of the symmetric algorithms used to encrypt a picture in numerous researches because of its straightforward structure and speedy computations. In order to address the problem with the inverse key matrices, which are typically not available in HC algorithms, Acharya et al. [15] introduced the picture encryption utilising an advanced HC approach, obviating the need for the receiver to compute the inverse key. With the aid of interlacing and iterations, Acharya et al. [16] enhanced the original HC by adding an unconscious key. However, because the solution was limited to one grayscale image, the analysis was faulty and the outcome was unreliable. In order to increase the security of the original HC technique, Hamissa et al. [17] proposed a unique encoder-decoder approach for picture encryption employing logistic map due functions. To increase the entropy of the encrypted image, researchers developed a system with three phases, including the HC. First, the two original images' pixel values are transformed to eight binary bits each, after which some fixed k bits are rotated and inverted. The lower nibbles of the image's pixels are then switched. The pixel values are then subjected to the HC method [18]. The ciphertext's numerical values are converted into points on the ECC via scalar multiplication using the HC algorithm, which is also utilised to produce another innovative encryption method. This strategy increases security while lengthening computation time [19]. The HC was further improved by Mahmoud et al. [20] to defend against assaults utilising statistics, brute force, and plaintext ciphertext. Later, Sun et al. [21] combined the contourlet-based steganography technique with the HC in their proposed picture encryption technology. Simultaneously, Naveenkumar et al. [22] offered a hybrid of Chaos and HC-based picture encryption that incorporates permutation and diffusion techniques as a different option to the traditional HC algorithm. Additionally, Sazaki et al. [23] integrated the advanced HC with the affine transform. The method used in [23] is slightly updated in [24] by Goutham et al. with regard to

the 128-bit key utilised. Due to the fact that the HC uses the same key for both encoding and decoding, it offers a low level of security. To fix this weakness, Hamissa et al. [17] introduced a revolutionary image encryption algorithm called Elliptic Curve Cryptosystem with Hill Cipher (ECCHC). This combination method results in the asymmetry of the ECCHC method. A binary extension field-based ECC system was subsequently proposed by Rabah [14] as a result. In this article, the fundamental EC and Diffie-Hellman design concepts are discussed. According to Farwa et al. [25], the group rule stated over the rational points of an EC over the GF also provides remarkable benefits when applied to block ciphers for the byte replacement process. A specific EC over the $GF\left(2^4\right)$ that has the same order as that of $\mathbb{F}_2^4$ is chosen for this. GF is used to handle large primes, which increases computational effort.

1.2 Motivation

The following are the primary justifications for suggesting this approach to boost the HC's speed and effectiveness by combining it with the EC over the GF.

- The fundamental flaw in the original HC technique was that it occasionally failed to recover plaintext since there was not a key matrix inverse because not all matrices could be inverted.
- Because the encryption and decryption operations used the same key, the HC also had the drawback of offering insufficient security.
- Agrawal et al. [19] suggested employing scalar multiplication to initially produce the ciphertext using numerical values before translating them into points on the ECC. Although the computing time was increased by this concept, the security level was increased.
- Although ECC was a novel and effective method, using big primes to achieve the required results increases computation time and complicates the algorithm [25].

1.3 Our Contribution

In order to increase security and create a new approach that follows the idea presented in [12], we offered a novel idea for image encryption that combines ECC over GF and HC with a modification in the key matrix utilised for the encoding and decoding process. According to [25], it stands to reason that the issue of handling large primes in ECC should be approached from a different angle. By using prime power fields, particularly binary extension fields, and a few certain elliptic curves, we can boost complexity while requiring little additional computation, avoiding this problem. This motive is the driving force for the method presented in this paper, which produces the private and public keys using ECC over the binary extension field as opposed to a prime field. The secret key can then be generated by both the sender and the receiver without being shared online or over an unsecured communication route. One of the critical problems in the HC method is that the inverse of the key matrix does not always exist. The decryption procedure will therefore fail and the receiver will not be able to recover the original data if the key matrix is not invertible. This method solves the issue of locating the inverse of the key matrix for decryption even though it uses the XOR operation because the same key matrix is utilised for both encryption and decryption. Additionally, the algorithm is strengthened against different cryptographic attacks by using the inverse function of the appropriate GF. The proposed scheme is presented in Section 3 of the remaining work, which also discusses the building of the Galois field, EC over the GF, and the original HC algorithm. We assessed the suggested algorithm performance indices in Section 4 and contrasted them to other existing S-boxes. Moreover, Section 4 provides an illustration of the suggested approach. The final segment will cover the conclusion.

2 Preliminaries

This section will discuss some basic definitions and concepts regarding EC and HC.

2.1 Elliptic Curve over a Finite Field

An EC over a field $\mathbb{F}_p$ (which is a prime field, and here p is a large prime), is defined by the Weierstrass equation

$$E: \; y^2 = x^3 + Ax + B, \; (mod\ p) \tag{1}$$

where $A, B \in \mathbb{F}_p$ and $4A^3 + 27B^2 \neq 0\ (mod\ p)$. Points of an EC over a finite field make a finite additive group that satisfies the property of an abelian group. All of the points that satisfy the EC and the infinity point O make an elliptic curve group [26,27].

2.1.1 Point Addition

Let $P'(x_1, y_1)$ and $Q'(x_2, y_2)$ be two points of the above EC satisfying its equation, then the addition of these two points $P' + Q' = R'(x_3, y_3)$ can be defined as

$$x_3 = t^2 - x_1 - x_2 \ \ (mod\ p)$$

$$y_3 = t(x_1 - x_3) - y_1 \ \ (mod\ p)$$

$$\text{where} \ \ t = \frac{y_2 - y_1}{x_2 - x_1} \ \ (mod\ p).$$

2.1.2 Point Doubling

Let $P'(x_1, y_1)$ be a point on E $\left(\mathbb{F}_p\right)$ satisfying its equation, then $P' + P' = 2P'(x_2, y_2)$ can be calculated as follows:

$$x_2 = t^2 - 2x_1 \ \ (mod\ p)$$

$$y_2 = t(x_1 - x_2) - y_1 \ \ (mod\ p)$$

$$\text{where} \ \ t = \frac{3x_1 + C}{2y_1} \ \ (mod\ p).$$

2.2 Galois Field

The binary field $\mathbb{F}_2^n$ is described as $\frac{\mathbb{F}_2[X]}{\langle f(X)\rangle}$ where $\langle f(X)\rangle$ would be a maximal ideal of $\mathbb{F}_2[X]$ such that an irreducible polynomial $f(X)$ having degree n generates it. As for GF $\mathbb{F}_2^4$, the irreducible primitive polynomial is given as $f(X) = x^4 + x + 1$. Elements of $\mathbb{F}_2^4$ form a cyclic group G under multiplication with order 15. Each element of this group is represented as a power of α that is a primitive element, just like $\alpha^2 = 0100$.

2.3 Elliptic Curve over $\mathbb{F}_2^4$

EC over the binary extension field is of the form

$$y^2 + xy = x^3 + Ax^2 + B \tag{2}$$

where $A, B \in \mathbb{F}_2^n$, and $B \neq 0$. Here, we take a special curve

$$E: \; y^2 + xy = x^3 + \alpha^4 x^2 + 1 \tag{3}$$

This curve has unique characteristics because of its rational points. The above curve has total 15 points including a point at infinity.

2.4 Group Law

An EC makes an additive group that adds points using group law [28]. Here, we discuss only the point doubling and the point addition laws for the EC over the GF.

2.4.1 Point Doubling

Let $P'(x_{P'}, y_{P'})$ be the point lying on the curve $y^2 + xy = x^3 + Ax^2 + B$ over $\mathbb{F}_2^n$, then by point doubling operation, we get 2P with coordinates.

$$x_{2P'} = s^2 + s + A$$

$$y_{2P'} = x_{P'}^2 + (s+1)\, x_{2P'}$$

where $s = x_{P'} + \dfrac{y_{P'}}{x_{P'}}$ is the slope of a line which is a tangent over P'. This operation is very useful in point multiplication as $9P' = 2\,(2\,(P' + P')) + P'$.

2.4.2 Point Addition

Let $P'(x_{P'}, y_{P'})$ and $Q'(x_{Q'}, y_{Q'})$ be two distinct points. The coordinates of $P' + Q'$ are given by

$$x_{P'+Q'} = S^2 + S + x_{P'} + x_{Q'} + A$$

$$y_{P'+Q'} = S\left(x_{P'} + x_{P'+Q'}\right) + x_{P'+Q'} + y_{P'}$$

where $S = \dfrac{y_{P'} + y_{Q'}}{x_{P'} + x_{Q'}}$, is the slope of a line that passes through the given two points.

2.5 Hill Cipher

The symmetric block cipher algorithm known as the HC was created by Lester Hill in 1929 [29]. Both the sender and the receiver use the same key matrix for ciphering and decoding. The fundamental idea is to give each letter a numerical value, such as a = 0, b = 1, ..., z = 25. Depending on the size of the key matrix having order $m \times m$, the plaintext should then be divided into blocks of the same size m. If $m = 2$ then the plaintext block $(P_{2\times1})$ will have a size 2 and the key matrix $(k_{2\times2})$ will have order 2×2. The encryption process ends by producing a ciphertext block with the following two values $(C_{2\times1})$ [11].

2.5.1 Encryption

If $P = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ and $K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$, then

$$C = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \; mod\; 26$$

$$= \begin{bmatrix} (k_{11}p_1 + k_{12}p_2)\, mod\ 26 \\ (k_{21}p_1 + k_{22}p_2)\, mod 26 \end{bmatrix}$$

2.5.2 Decryption

For decryption, the receiver must find K^{-1} such that $K^{-1} \cdot K = I$ where I is the 2×2 identity matrix [27]. So, decryption can be done as given

$$P = K^{-1} \cdot C\ mod\ 26$$

3 Proposed Algorithm

This section provides a novel proposed technique that combines EC over the GF with HC, which is more efficient and secure than the original HC method. This technique has the important benefit of avoiding the difficult calculations involved in matrix multiplication using XOR which speeds up decryption computations by removing the need to compute the inverse of the key matrix. Additionally, employing GF operations expedites the procedure and strengthens the suggested scheme. Let us assume that the sender (a) chooses to use this approach to send an image to the recipient (b) across an insecure channel. First, they should agree on the EC function E, and then the domain parameters $\{A, B, n, G\}$ should be shared by the two users where A, B are the EC coefficients and n is the power of binary extension field $GF\ (2^n)$ such that n cannot be greater than the number of bits of the highest image pixel value, and G is the generator point. Then, each partner must choose his private key randomly from a given interval $[1,\ 2^n - 1]$, say n_a the private key of the sender (a) and n_b the recipient (b), after which they compute their public keys in the way shown below.

3.1 Key Generation

The Public key for a is $P_a = n_a \cdot G$ and $P_b = n_b \cdot G$ for b. To get the initial key K_m, each user gets a product of their private key with the public key of the other user.

$$K_m = n_a \cdot P_b = n_b \cdot P_a = n_a \cdot n_b \cdot G = (x, y)$$

Then it evaluates

$$K_1 = x \cdot G = \left(k'_{11}, k'_{12}\right)$$

$$K_2 = y \cdot G = \left(k'_{21}, k'_{22}\right)$$

The above values generate the initial key matrix that is given as

$$K = \begin{bmatrix} k'_{11} & k'_{12} \\ k'_{21} & k'_{22} \end{bmatrix}$$

The sender and the receiver then work together to construct the key matrix. Subsequently, an encrypted message cannot be decoded by the recipient because it is not always possible to create an invertible key matrix. However, the inverse key matrix will not need to be evaluated because this suggested method uses the XOR for encryption and decryption. In this technique, the image will be divided into sixteen pixel-sized chunks, which are then converted into a matrix of size 4×4 whose entries are then transferred into the elements of $GF\ (2^n)$. Then it is computed with a key matrix K_M of

the same size which can be constructed using the initial key matrix as follows:

$$K_M = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix}$$

over $GF(2^n)$ which is consist of 4 matrices of size 2×2 such that $K_M = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$. If $K_{11} = K = \begin{bmatrix} k'_{11} & k'_{12} \\ k'_{21} & k'_{22} \end{bmatrix}$, then the others can be calculated as $K_{12} = K \cdot K = K^2$, $K_{21} = K^{-1}$, and $K_{22} = (K^2)^{-1}$. Hence K_M is given by

$$K_M = \begin{bmatrix} K & K^2 \\ K^{-1} & (K^2)^{-1} \end{bmatrix}$$

over $GF(2^n)$.

3.2 Encryption

After dividing the values of image pixels into 16 size blocks that are transformed into the size of a 4×4 matrix which is $(P_1, P_2, P_2, \ldots)$ with entries converted into the elements of $GF(2^n)$, each 4×4 matrix of plaintext (Image) is then multiplied by K_M using XOR, producing a 4×4 matrix of ciphertext. This procedure may be described as

$C_i = K_M \oplus P_i$ where $i = 1, 2, 3, \ldots$

For $i = 1$

$C_1 = K_M \oplus P_1$

$$C_1 = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} \oplus \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

An inverse map is applied after assessing each 4×4 matrix of the plain picture,

$\varphi : C_i \rightarrow \mathrm{C_i}$

such that

$\varphi(x) = x^{-1}$ if $x \in GF(2^n)/\{0\}$
$= x$ if $x = 0$

Here, n is the power of binary extension field $GF(2^n)$ such that n cannot be greater than the number of bits of the highest image pixel value. Moreover, the collected data is integrated into a single matrix that contains the ciphered information.

3.3 Decryption

The decryption procedure can start after the encrypted image has been received. Because we utilise XOR instead of matrix multiplication, we can skip computing the inverse key matrix. First, we arrange

all of the 4 × 4 matrices of ciphered image ($C_1, C_2, C_3, \ldots$) and then use an inverse map

$\varphi : C_i \to \mathrm{C_i}$

such that

$$\varphi^{-1}(x) = x^{-1} \quad \text{if} \quad x \in GF(2^{\mathrm{n}})/\{0\}$$
$$= x \quad \text{if} \quad x = 0$$

where n is the power of binary extension field $GF(2^n)$ such that n cannot be greater than the number of bits of the highest image pixel value. Following that, we apply the XOR function to each matrix C_i with the key matrix K_M to produce the 4 × 4 matrix below that contains the original image's pixel values.

$P_i = K_M \oplus C_i$ where $i = 1, 2, 3, \ldots$

For $i = 1$

$P_1 = K_M \oplus C_1$

$$P_1 = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} \oplus \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

After evaluating P_i, they are combined into a single matrix to get the original image.

3.4 Proposed Technique Using ECC

3.4.1 Key Generation

User (a)

1. Choose a private key $n_a \in [1, 2^n - 1]$
2. Calculate the public key $P_a = n_a \cdot G$
3. Evaluate initial key $K_m = n_a \cdot P_b = n_a \cdot n_b \cdot G = (x, y)$.
4. Compute $K_1 = x \cdot G = \left(k'_{11}, k'_{12}\right)$ *and* $K_2 = y \cdot G = \left(k'_{21}, k'_{22}\right)$.
5. Here $K = K_{11} = \begin{bmatrix} k'_{11} & k'_{12} \\ k'_{21} & k'_{22} \end{bmatrix}$ is the initial key matrix, then $K_{12} = K^2 = \begin{bmatrix} k'_{13} & k'_{14} \\ k'_{23} & k'_{24} \end{bmatrix}$, $K_{21} = K^{-1} = \begin{bmatrix} k'_{31} & k'_{32} \\ k'_{41} & k'_{42} \end{bmatrix}$, *and* $K_{22} = \left(K^2\right)^{-1} = \begin{bmatrix} k'_{33} & k'_{34} \\ k'_{43} & k'_{44} \end{bmatrix}$.
6. Then finally, the key matrix of size 4 × 4 is generated as follows:

$$K_M = \begin{bmatrix} K & K^2 \\ K^{-1} & (K^2)^{-1} \end{bmatrix} = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix}$$

over $GF(2^n)$.

User (b).

1. Select a private key $n_b \in [1, 2^n - 1]$
2. Calculate the public key $P_b = n_b \cdot G =$

3. Evaluate initial key $K_m = n_b \cdot P_a = n_b \cdot n_a \cdot G = (x, y)$.
4. Compute $K_1 = x \cdot G = \left(k'_{11}, k'_{12}\right)$ *and* $K_2 = y \cdot G = \left(k'_{21}, k'_{22}\right)$.
5. Here $K = K_{11} = \begin{bmatrix} k'_{11} & k'_{12} \\ k'_{21} & k'_{22} \end{bmatrix}$ is the initial key matrix, then $K_{12} = K^2 = \begin{bmatrix} k'_{13} & k'_{14} \\ k'_{23} & k'_{24} \end{bmatrix}$, $K_{21} = K^{-1} = \begin{bmatrix} k'_{31} & k'_{32} \\ k'_{41} & k'_{42} \end{bmatrix}$, *and* $K_{22} = \left(K^2\right)^{-1} = \begin{bmatrix} k'_{33} & k'_{34} \\ k'_{43} & k'_{44} \end{bmatrix}$
6. Then finally, the key matrix of size 4×4 is generated as given below

$$K_M = \begin{bmatrix} K & K^2 \\ K^{-1} & (K^2)^{-1} \end{bmatrix} = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix}$$

over $GF(2^n)$.

3.4.2 Encryption

1. Divide 256×256 pixel values of the image into blocks of size 16 and convert them into the matrix of size 4×4 (P_i). Entries of that matrix are then converted into elements of $GF(2^n)$ as

$$P_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

2. Apply the operation of XOR to each P_i with K_M one by one to get the data in the form of

$$C_1 = K_M \oplus P_1$$

$$C_1 = \begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} \oplus \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$C_1 = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

3. After obtaining C_1, apply an inverse function under the $GF(2^n)$ to each entry of that matrix to get the matrix in the form of

$$C'_1 = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}^{-1}$$

4. Similarly, apply the same technique to all of the 4×4 matrices of the plain image P_i to get the resultant 4×4 matrices C'_i and combine them to get the ciphered image.

3.4.3 Decryption

1. Separate 256×256 pixel values of the ciphered image into blocks of size 16 and convert them into the matrix of size 4×4 $\left(C_i^{'}\right)$ whose entries are from $GF\left(2^n\right)$ as

$$C_1^{'} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}^{-1}$$

2. Apply an inverse function under $GF\left(2^n\right)$ to each entry of the matrix $C_1^{'}$ to get the matrix such that

$$C_1 = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

3. Similarly, take all matrices of size 4×4 $\left(C_i^{'}\right)$ and apply this technique to each matrix to get the data in the form of C_i.
4. Apply XOR operation to each C_i with the key matrix K_M one by one to get values of the form

$$P_1 = K_M \oplus C_1$$

$$P_1 = \begin{bmatrix} k_{11}^{'} & k_{12}^{'} & k_{13}^{'} & k_{14}^{'} \\ k_{21}^{'} & k_{22}^{'} & k_{23}^{'} & k_{24}^{'} \\ k_{31}^{'} & k_{32}^{'} & k_{33}^{'} & k_{34}^{'} \\ k_{41}^{'} & k_{42}^{'} & k_{43}^{'} & k_{44}^{'} \end{bmatrix} \oplus \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

$$P_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

5. Finally, combine all of the matrices P_i of size 4×4 to get the required ciphered image of 256×256.

3.5 Example

Let a person (a) want to deliver an image M to the other person (b), and they both agree on the EC function given by

$$E: \; y^2 + xy = x^3 + \alpha^4 x^2 + 1$$

over $GF\left(2^4\right)$, where $A = \alpha^4$ *and* $B = 1$. Let us define the 15 elements of this field that can be expressed as a power of primitive root as well as binary form, as shown in Table 1.

For the chosen curve, we get a generator point $G = \left(\alpha^3, \; \alpha^8\right)$, all other points can be found using group law and can be expressed as a multiple of $G \;\; (x_G, \; y_G)$. Let us calculate $2G \;\; (x_{2G}, \; x_{2G})$.

$$s = x_G + \frac{y_G}{x_G} = \alpha^3 + \frac{\alpha^8}{\alpha^3} = \alpha^3 + \alpha^5 = \alpha^{11},$$

$$x_{2G} = s^2 + s + A$$

$$x_{2G} = (\alpha^{11})^2 + \alpha^{11} + \alpha^4$$

$$x_{2G} = \alpha^5$$

$$y_{2G} = x_G^2 + (s+1)\, x_{2G}$$

$$y_{2G} = (\alpha^3)^2 + \left(\alpha^{11} + 1\right) \cdot \alpha^5$$

$$y_{2G} = \alpha^3$$

$$2G = G_6$$

Table 1: Elements of GF $\left(2^4\right)$

$\alpha^0 = 0001,$	$\alpha^4 = 0011,$	$\alpha^8 = 0101,$	$\alpha^{12} = 1111$
$\alpha^1 = 0010,$	$\alpha^5 = 0110,$	$\alpha^9 = 1010,$	$\alpha^{13} = 1101$
$\alpha^2 = 0100,$	$\alpha^6 = 1100,$	$\alpha^{10} = 0111,$	$\alpha^{14} = 1001$
$\alpha^3 = 1000,$	$\alpha^7 = 1011,$	$\alpha^{11} = 1110,$	$\alpha^{15} = 0001$

All other points are given in Table 2.

Table 2: Points lying on given EC

G_∞	$G = G_4\left(\alpha^3,\ \alpha^8\right)$	$5G = G_8\left(\alpha^6,\ \alpha^8\right)$	$6G = G_{12}\left(\alpha^{10},\ \alpha\right)$
$8G = G_1\,(0,1)$	$15G = G_5\left(\alpha^3,\ \alpha^{13}\right)$	$11G = G_9\left(\alpha^6,\ \alpha^{14}\right)$	$10G = G_{13}\left(\alpha^{10},\ \alpha^8\right)$
$12G = G_2\left(1, \alpha^6\right)$	$2G = G_6\left(\alpha^5,\ \alpha^3\right)$	$3G = G_{10}\left(\alpha^9,\ \alpha^{10}\right)$	$9G = G_{14}\left(\alpha^5,\ 0\right)$
$4G = G_3\left(1, \alpha^{13}\right)$	$14G = G_7\left(\alpha^5,\ \alpha^{11}\right)$	$13G = G_{11}\left(\alpha^9,\ \alpha^{13}\right)$	$7G = G_{15}\left(\alpha^{12},\ \alpha^{12}\right)$

As the generator point is $G\left(\alpha^3,\ \alpha^8\right)$, other domain parameters are $\left\{A, B,\ 2^4 - 1, G\right\} = \left\{\alpha^4, 1, 15, \left(\alpha^3,\ \alpha^8\right)\right\}$. If a person (a) desires to deliver an image of size 256 × 256 to person (b), they should use the proposed method.

3.5.1 Key Generation

User (a)

1. Select a random value as a private key $n_a = 6 \in [1, 15]$
2. Calculate the public key $P_a = n_a \cdot G = 6\left(\alpha^3,\ \alpha^8\right)$
3. Evaluate initial key $K_m = n_a \cdot P_b = n_a \cdot n_b \cdot G = 6.11\left(\alpha^3,\ \alpha^8\right) = 6\left(\alpha^3,\ \alpha^8\right) = \left(\alpha^{10},\ \alpha\right) = (7,2) = (x, y)$
4. Compute $K_1 = x \cdot G = 7\left(\alpha^3,\ \alpha^8\right) = \left(\alpha^{12},\ \alpha^{12}\right) = (k_{11}, k_{12})$ *and* $K_2 = y \cdot G = 2\left(\alpha^3,\ \alpha^8\right) = \left(\alpha^5,\ \alpha^3\right) = (k_{21}, k_{22})$
5. Here $K = K_{11} = \begin{bmatrix} \alpha^{12} & \alpha^{12} \\ \alpha^5 & \alpha^3 \end{bmatrix} = \begin{bmatrix} 15 & 15 \\ 6 & 8 \end{bmatrix}$ is the initial key matrix, then $K_{12} = K^2 = \begin{bmatrix} \alpha^{11} & \alpha^7 \\ \alpha^0 & \alpha^3 \end{bmatrix} = \begin{bmatrix} 14 & 11 \\ 1 & 8 \end{bmatrix}$, $K_{21} = K^{-1} = \begin{bmatrix} \alpha^{10} & \alpha^4 \\ \alpha^{12} & \alpha^4 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 15 & 3 \end{bmatrix}$ *and* $K_{22} = \left(K^2\right)^{-1} = \begin{bmatrix} \alpha^2 & \alpha^6 \\ \alpha^{14} & \alpha^{10} \end{bmatrix} = \begin{bmatrix} 4 & 12 \\ 9 & 7 \end{bmatrix}$

6. Then finally, the key matrix of size 4×4 is generated as follows:

$$K_M = \begin{bmatrix} \alpha^{12} & \alpha^{12} & \alpha^{11} & \alpha^{7} \\ \alpha^{5} & \alpha^{3} & \alpha^{0} & \alpha^{3} \\ \alpha^{10} & \alpha^{4} & \alpha^{2} & \alpha^{6} \\ \alpha^{12} & \alpha^{4} & \alpha^{14} & \alpha^{10} \end{bmatrix} = \begin{bmatrix} 15 & 15 & 14 & 11 \\ 6 & 8 & 1 & 8 \\ 7 & 3 & 4 & 12 \\ 15 & 3 & 9 & 7 \end{bmatrix}$$

User (b)

1. Select a private key $n_b = 11 \in [1, 15]$
2. Calculate the public key $P_b = n_b \cdot G = 6\left(\alpha^3,\ \alpha^8\right)$
3. Evaluate initial key $K_m = n_b \cdot P_a = n_b \cdot n_a \cdot G = 11.6\left(\alpha^3,\ \alpha^8\right) = 6\left(\alpha^3,\ \alpha^8\right) = (\alpha^{10},\ \alpha) = (7, 2) = (x, y)$
4. Compute $K_1 = x \cdot G = 7\left(\alpha^3,\ \alpha^8\right) = \left(\alpha^{12},\ \alpha^{12}\right) = (k_{11}, k_{12})$ *and* $K_2 = y \cdot G = 2\left(\alpha^3,\ \alpha^8\right) = \left(\alpha^5,\ \alpha^3\right) = (k_{21}, k_{22})$
5. Here $K = K_{11} = \begin{bmatrix} \alpha^{12} & \alpha^{12} \\ \alpha^{5} & \alpha^{3} \end{bmatrix} = \begin{bmatrix} 15 & 15 \\ 6 & 8 \end{bmatrix}$ is the initial key matrix, then $K_{12} = K^2 = \begin{bmatrix} \alpha^{11} & \alpha^{7} \\ \alpha^{0} & \alpha^{3} \end{bmatrix} = \begin{bmatrix} 14 & 11 \\ 1 & 8 \end{bmatrix}$, $K_{21} = K^{-1} = \begin{bmatrix} \alpha^{10} & \alpha^{4} \\ \alpha^{12} & \alpha^{4} \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 15 & 3 \end{bmatrix}$ *and* $K_{22} = \left(K^2\right)^{-1} = \begin{bmatrix} \alpha^{2} & \alpha^{6} \\ \alpha^{14} & \alpha^{10} \end{bmatrix} = \begin{bmatrix} 4 & 12 \\ 9 & 7 \end{bmatrix}$
6. Then finally, the key matrix of size 4×4 is generated as follows:

$$K_M = \begin{bmatrix} \alpha^{12} & \alpha^{12} & \alpha^{11} & \alpha^{7} \\ \alpha^{5} & \alpha^{3} & \alpha^{0} & \alpha^{3} \\ \alpha^{10} & \alpha^{4} & \alpha^{2} & \alpha^{6} \\ \alpha^{12} & \alpha^{4} & \alpha^{14} & \alpha^{10} \end{bmatrix} = \begin{bmatrix} 15 & 15 & 14 & 11 \\ 6 & 8 & 1 & 8 \\ 7 & 3 & 4 & 12 \\ 15 & 3 & 9 & 7 \end{bmatrix}$$

3.5.2 Encryption

1. Choose an image of size 256×256 and divide the pixel values into blocks of size 16 and convert them into the matrix of size 4×4 (P_i). Entries of that matrix then converted into elements of $GF\left(2^8\right)$ As

$$P_1 = \begin{bmatrix} 165 & 161 & 157 & 157 \\ 163 & 160 & 158 & 159 \\ 161 & 159 & 158 & 160 \\ 159 & 158 & 157 & 158 \end{bmatrix}$$

2. Apply the operation of XOR to P_1 with K_M to get the matrix of the form

$$C_1 = K_M \oplus P_1$$

$$C_1 = \begin{bmatrix} 15 & 15 & 14 & 11 \\ 6 & 8 & 1 & 8 \\ 7 & 3 & 4 & 12 \\ 15 & 3 & 9 & 7 \end{bmatrix} \oplus \begin{bmatrix} 165 & 161 & 157 & 157 \\ 163 & 160 & 158 & 159 \\ 161 & 159 & 158 & 160 \\ 159 & 158 & 157 & 158 \end{bmatrix} = \begin{bmatrix} 170 & 174 & 147 & 150 \\ 165 & 168 & 159 & 151 \\ 166 & 156 & 154 & 172 \\ 144 & 157 & 148 & 153 \end{bmatrix}$$

3. Likewise, apply the same procedure to each matrix P_i to get the matrices C_i.

4. Apply an inverse function under the $GF\left(2^8\right)$ to each element of the matrix C_1 such that

$$C_1' = \begin{bmatrix} 170 & 174 & 147 & 150 \\ 165 & 168 & 159 & 151 \\ 166 & 156 & 154 & 172 \\ 144 & 157 & 148 & 153 \end{bmatrix}^{-1}$$

$$C_1' = \begin{bmatrix} 13 & 194 & 170 & 124 \\ 190 & 206 & 172 & 9 \\ 79 & 162 & 189 & 155 \\ 15 & 92 & 156 & 220 \end{bmatrix}$$

5. Similarly, take all of the matrices C_i and apply an inverse function under the $GF\left(2^8\right)$ to get all the 64 matrices C_i' of size 4×4 and combined them to get the encrypted image of 256×256.

3.5.3 Decryption

1. Separate 256×256 pixel values of the encrypted image into a block of size 16 and convert them into a matrix of size 4×4 $\left(C_i'\right)$, whose entries are from $GF\left(2^8\right)$. Firstly, we take the matrix C_1' such that

$$C_1' = \begin{bmatrix} 13 & 194 & 170 & 124 \\ 190 & 206 & 172 & 9 \\ 79 & 162 & 189 & 155 \\ 15 & 92 & 156 & 220 \end{bmatrix}$$

2. Apply an inverse to each entry of the matrix C_1 under the $GF\left(2^8\right)$ to get the matrix C_1 given below

$$C_1 = \begin{bmatrix} 170 & 174 & 147 & 150 \\ 165 & 168 & 159 & 151 \\ 166 & 156 & 154 & 172 \\ 144 & 157 & 148 & 153 \end{bmatrix}$$

3. Similarly, apply an inverse function under $GF\left(2^8\right)$ to each matrix C_i' to get the matrices C_i of size 4×4.
4. After obtaining C_i, the operation of XOR is utilized for each matrix C_i with the key matrix K_M one by one such that

$$P_1 = K_M \oplus C_1$$

$$P_1 = \begin{bmatrix} 15 & 15 & 14 & 11 \\ 6 & 8 & 1 & 8 \\ 7 & 3 & 4 & 12 \\ 15 & 3 & 9 & 7 \end{bmatrix} \oplus \begin{bmatrix} 170 & 174 & 147 & 150 \\ 165 & 168 & 159 & 151 \\ 166 & 156 & 154 & 172 \\ 144 & 157 & 148 & 153 \end{bmatrix} = \begin{bmatrix} 165 & 161 & 157 & 157 \\ 163 & 160 & 158 & 159 \\ 161 & 159 & 158 & 160 \\ 159 & 158 & 157 & 158 \end{bmatrix}$$

5. Finally, combine all of the matrices P_i of size 4×4 to get the required original image of 256×256.

In this article, we encrypted images of Lena, Cameraman and Baboon and their respective encrypted images are depicted in Fig. 1. After analyzing them, we conclude that our approach is more robust against cryptographic attacks compared to the existing schemes mentioned in Section 5.

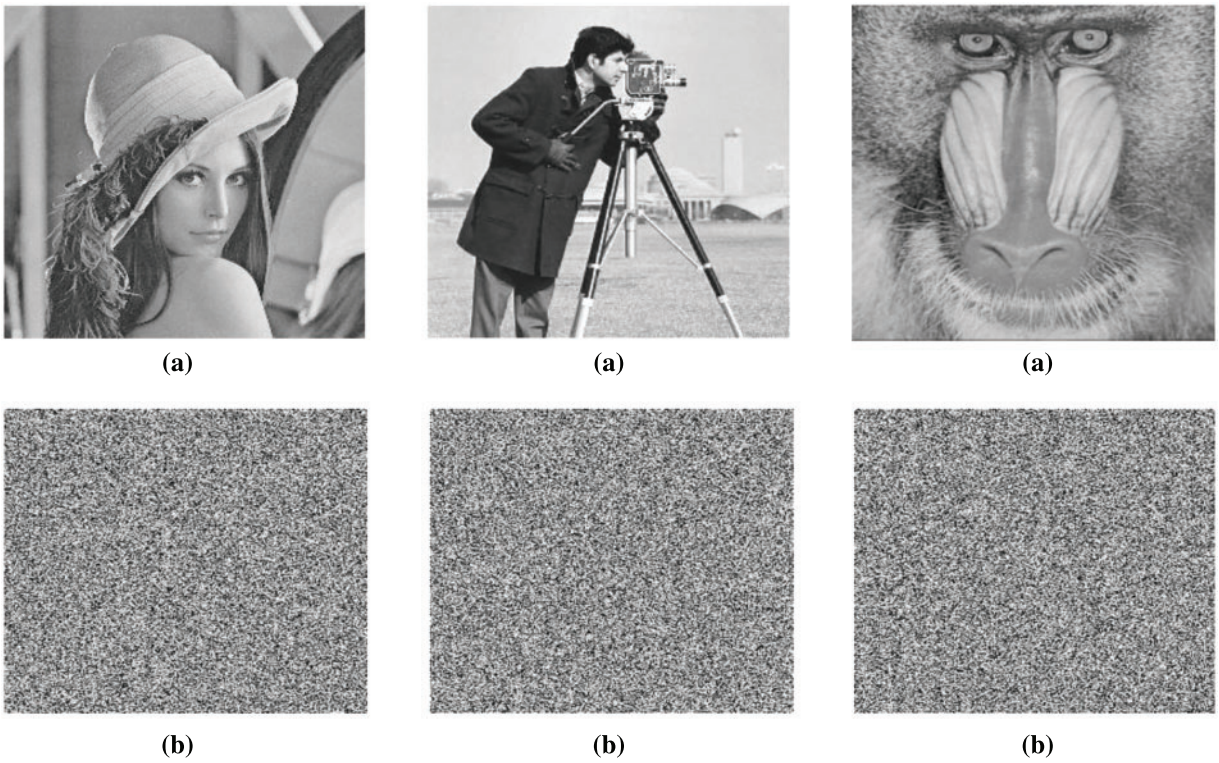

Figure 1: (a) Original Images: Lena, Cameraman, and Baboon. (b) Encrypted images. Lena, Cameraman, and the Baboon

4 Security Analysis

To check the performance of the proposed encryption technique, some parameters or measures are used to check the efficiency of the greyscale image and compare the original image with the encrypted image. The resulting encrypted images were also subjected to various performance tests, which will be discussed in the following subsections, to assess their security against different types of attacks. A comparison has also been made between the existing schemes and this new approach.

4.1 Entropy Analysis

Entropy is the measurement of randomness and statistical parameter used to evaluate image enciphering. It represents the trends that occur most frequently. The formula for calculating entropy is given below.

$$E = \sum_{a=0}^{255} \left[P(a) \times \log_2 \frac{1}{P(a)} \right] \tag{4}$$

where $P(a)$ is the probability of pixel value a calculated as

$$P(a) = \frac{The \quad frequency \quad of \quad pixel \quad value}{Total \quad number \quad of \quad image \quad pixels} \tag{5}$$

For the greyscale image of size 256×256, the ideal entropy value is 8. As closer the entropy value to eight, the encrypted image is efficient. The entropy value of the proposed scheme is almost equal

to 8, which is significantly better than the outcome of the existing methods mentioned in Table 3. Accordingly, the proposed scheme created optimum haphazardness in the ciphered images. In this manner, the proposed cryptosystem effectively endures entropy attacks.

Table 3: Entropy analysis

Test images	Image size	Cipher image
Lena	256 × 256	7.9973
Baboon	256 × 256	7.9973
Cameramen	256 × 256	7.9972
Ref [30]; Lena	256 × 256	7.9970
Ref [31]; Lena	256 × 256	7.9970
Ref [31]; Baboon	256 × 256	7.9969
Ref [12]; Lena	256 × 256	7.9970
Ref [12]; Cameramen	256 × 256	7.9848
Ref [32]; Lena	256 × 256	7.9962
Ref [32]; Baboon	256 × 256	7.9971

4.2 NPCR and UACI

To measure the difference between the original and encrypted images, we use the number of pixels changes rate (NPCR) and the unified average changing intensity (UACI) tests. They are used to test the strength of the encryption process. The NPCR is used to check the number of changing pixels between the original and ciphered images. In contrast, the UACI measures the average change in intensity between the original and ciphered image. Its value depends upon the size and the format of the image. These two tests are used to show the resistance of the algorithm to different attacks. The formula for calculating NPCR is given as

$$NPCR = \sum_{i=1}^{m} \sum_{j=1}^{n} K(i, j) \times \frac{100\%}{m \times n} \tag{6}$$

$$\text{where } K(i, j) = \begin{cases} 0, & if\ i = j \\ 1, & if\ i \neq j \end{cases} \tag{7}$$

UACI can be evaluated as

$$UACI = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|X(i, j) - Y(i, j)|}{255} \tag{8}$$

The value of UACI for the greyscale image of size 256 × 256 is 33.61, and the value of NPCR is close to 100%, which shows that the proposed algorithm is resistant to different attacks. Finally, the analyses also show that the suggested method's diffusion property is remarkably superior to the schemes shown in Table 4.

Table 4: NPCR and UACI analysis

Test images	Image size	NPCR	UACI
Lena	256 × 256	99.61	33.72
Baboon	256 × 256	99.63	33.70
Cameraman	256 × 256	99.60	33.61
Ref [26]; Lena	256 × 256	98.68	30.38
Ref [33]; Lena	256 × 256	97.74	38.33
Ref [34]; Cameraman	256 × 256	98.08	30.17
Ref [35]; Cameraman	256 × 256	99.44	31.12
Ref [36]; Baboon	256 × 256	98.68	32.62

4.3 Histogram Analysis

Histogram analysis is one of the most straightforward methods to illustrate image encryption quality. It is a graph that shows the number of pixels of an image at different intensity values found in the image. For good encryption, the graph of the histogram should be uniformly distributed. It is used to show how an algorithm is resistant to statistical attacks. The corresponding histogram of the original, encrypted, and decrypted image is given below. The original and decrypted image has the same graph, so there is no data loss. In contrast, the histogram of the encrypted image is flat, indicating that the encryption scheme is good. Fig. 2 shows the histograms of the original images and corresponding ciphered images of Lena, Cameraman, and Baboon.

4.4 Correlation Coefficient Analysis

Correlation measures the degree of relationship between two adjacent pixels in an image and the degree of association between two adjacent pixels. Standard images we see daily have a high correlation of pixel values with their neighbors. There will be a very low correlation of pixel values with their neighbors for good image encryption. Generally, if the correlation coefficient is equal to zero or is about zero, then the plain image and its encrypted image are different. This means the encrypted image is highly independent of the plain image. A correlation coefficient of less than 0.1 between the plain and ciphered images is preferable. It is calculated as

$$CC = \frac{cov\,(x, z)}{\sigma_y \times \sigma_z} \tag{9}$$

where $\sigma_y = \sqrt{var\,(y)}$ and $\sigma_z = \sqrt{var\,(z)}$

$$var\,(y) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - E\,(y)\right)^2 \tag{10}$$

$$cov\,(y, z) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - E\,(y)\right)\left(z_i - E\,(z)\right) \tag{11}$$

Table 5 displays the experimental outcomes of the correlation test of the original and cipher images.

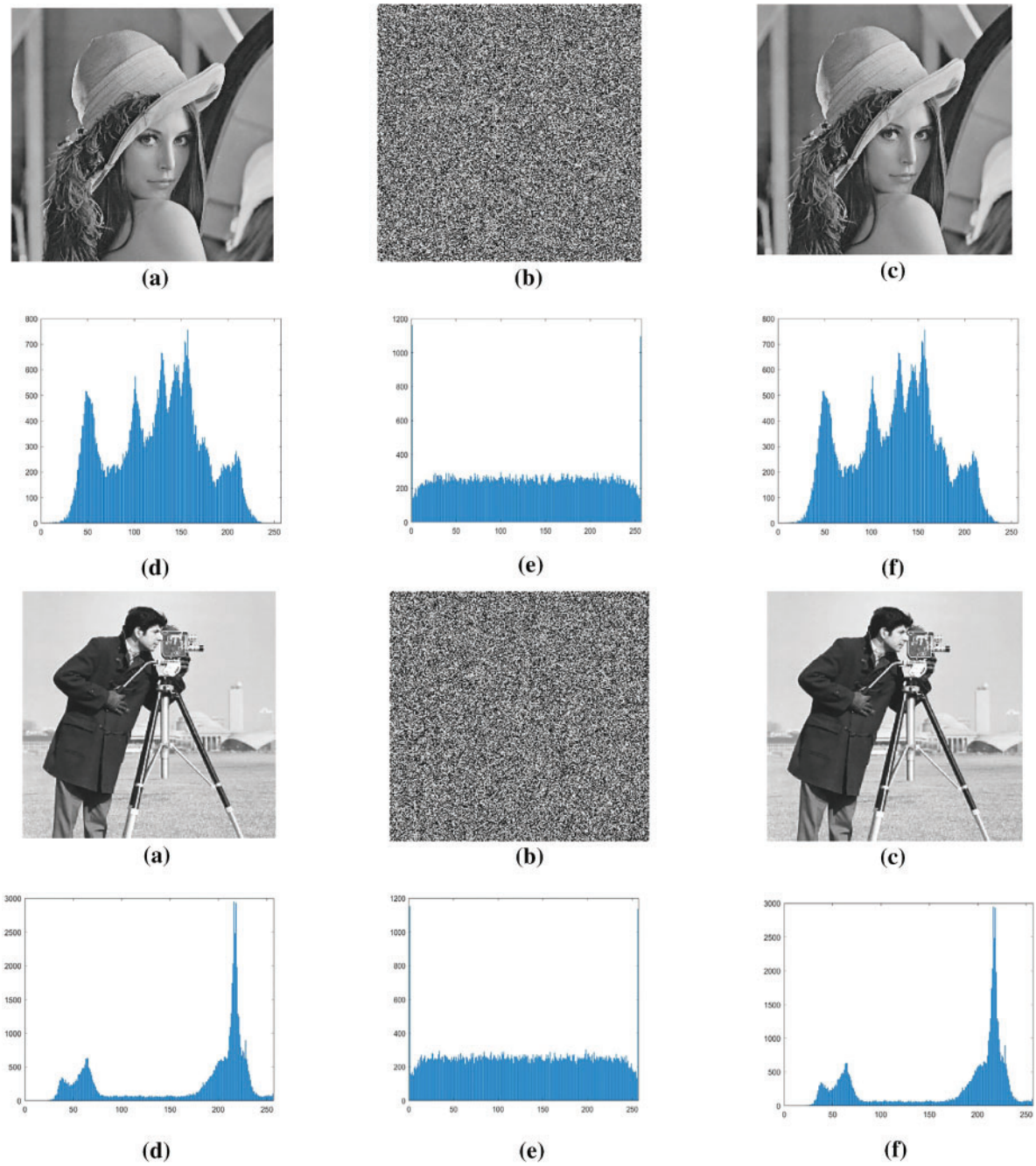

Figure 2: (Continued)

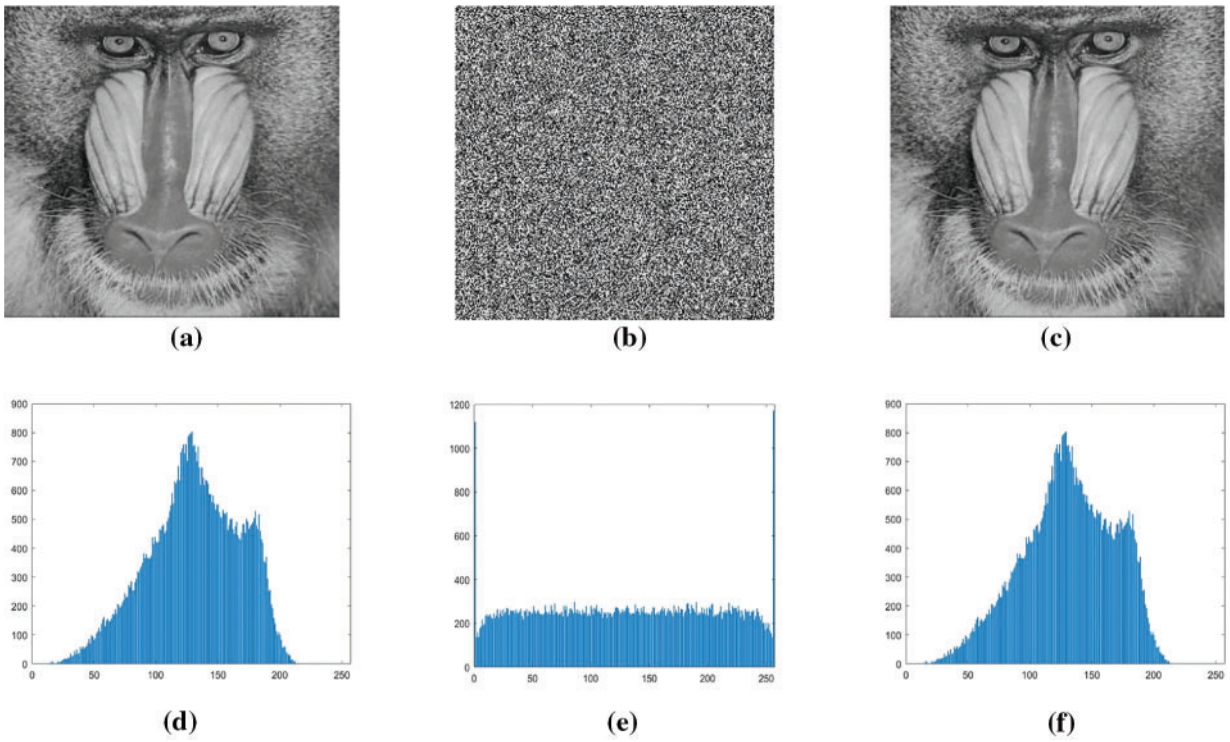

Figure 2: (a) Original Images: Lena, Cameraman, and Baboon. (b) Encrypted Images. Lena, Cameraman, and Baboon. (c) Decrypted images. Lena, Cameraman, and Baboon. (d) Histogram of original images. Lena, Cameraman, and Baboon. (e) Histogram of encrypted images. Lena, Cameraman, and Baboon. (f) Histogram of decrypted images. Lena, Cameraman, and Baboon

Table 5: Correlation analysis and their comparison with existing schemes

Test images	Image-size	Diagonal	Horizontal	Vertical
Lena	256 × 256	0.8941	0.9038	0.9451
Enc-Lena	256 × 256	−0.0093	0.0168	0.0081
Cameraman	256 × 256	0.9044	0.9372	0.9595
Enc-Cameraman	256 × 256	0.0092	0.0101	−0.0013
Baboon	256 × 256	0.8282	0.9028	0.8718
Enc-Baboon	256 × 256	0.0330	0.0086	−0.0032
Ref [33]; Lena	256 × 256	0.9376	0.8714	0.8359
Enc-Lena	256 × 256	0.0047	−0.0016	−0.0069
Ref [35]; Cameraman	256 × 256	0.9463	0.9187	0.9423
Enc-Cameraman	256 × 256	0.1856	0.0529	0.0257

Additionally, compared to some existing relevant literature, the correlation coefficient results demonstrate that the suggested encryption system is significantly more effective and immune to statistical attacks (Figs. 3–5).

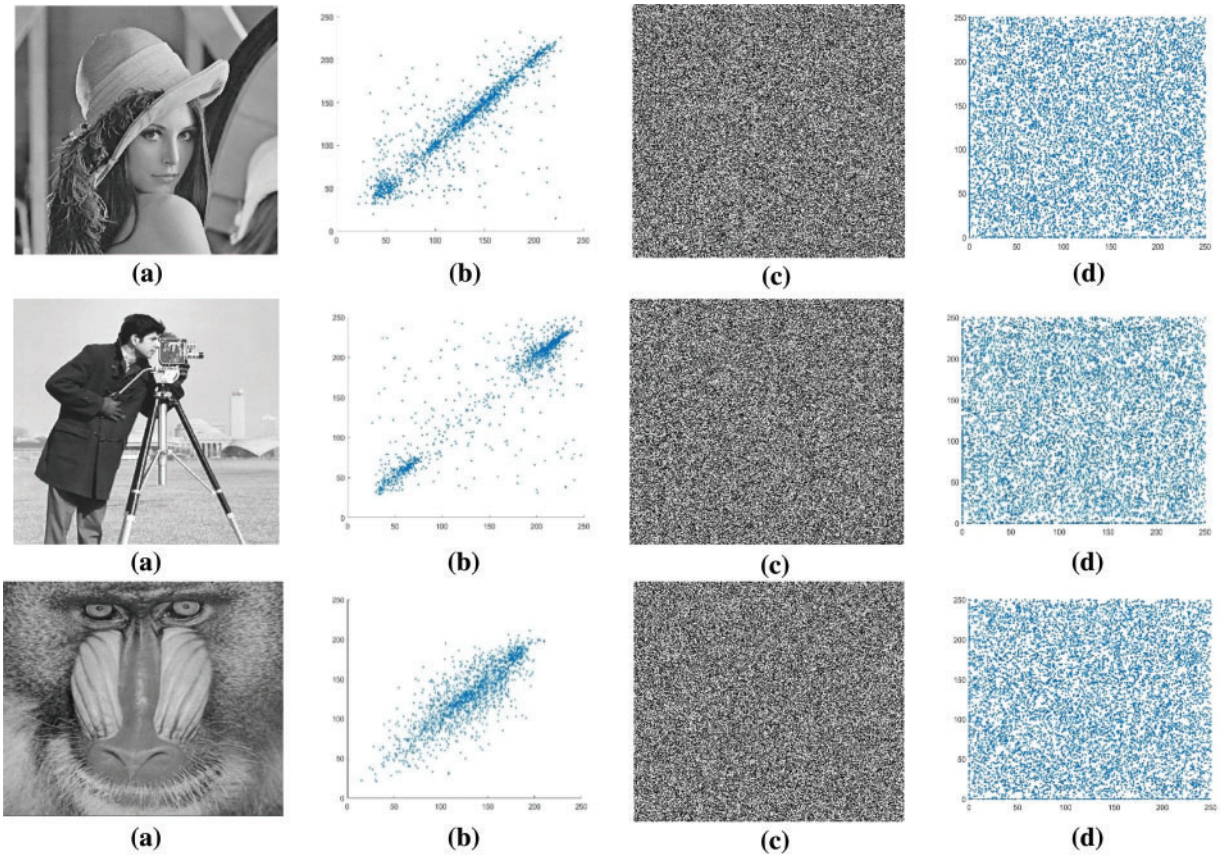

Figure 3: (a) Original Images. Lena, Cameraman, and Baboon. (b) Diagonal correlation of original images. Lena, Cameraman, and Baboon. (c) Encrypted images. Lena, Cameraman, and Baboon. (d) Diagonal correlation of encrypted images. Lena, Cameraman, and Baboon

4.5 Contrast

The contrast ratio is an essential feature of picture quality because it allows the viewer to identify the object in the image. Contrast analysis is a technique for determining the intensity level of contrast about pixels in an image. The encryption system is said to pass the contrast test if the contrast ratio in the ciphered image is high. The following is the mathematical expression of the contrast coefficient:

$$C^* = \sum_{i,j} \frac{f\,(i,j)}{1 + |i - j|} \tag{12}$$

where $f\,(i,j)$ represents the number of gray level co-occurrence matrices (GLCM) of the image. The constant image has 0 contrast value while the contrast value of the ciphered image is around 10.52, given in Table 6, indicating the presence of a significant change in the intensity of a pixel and its neighbor across the entire ciphered image.

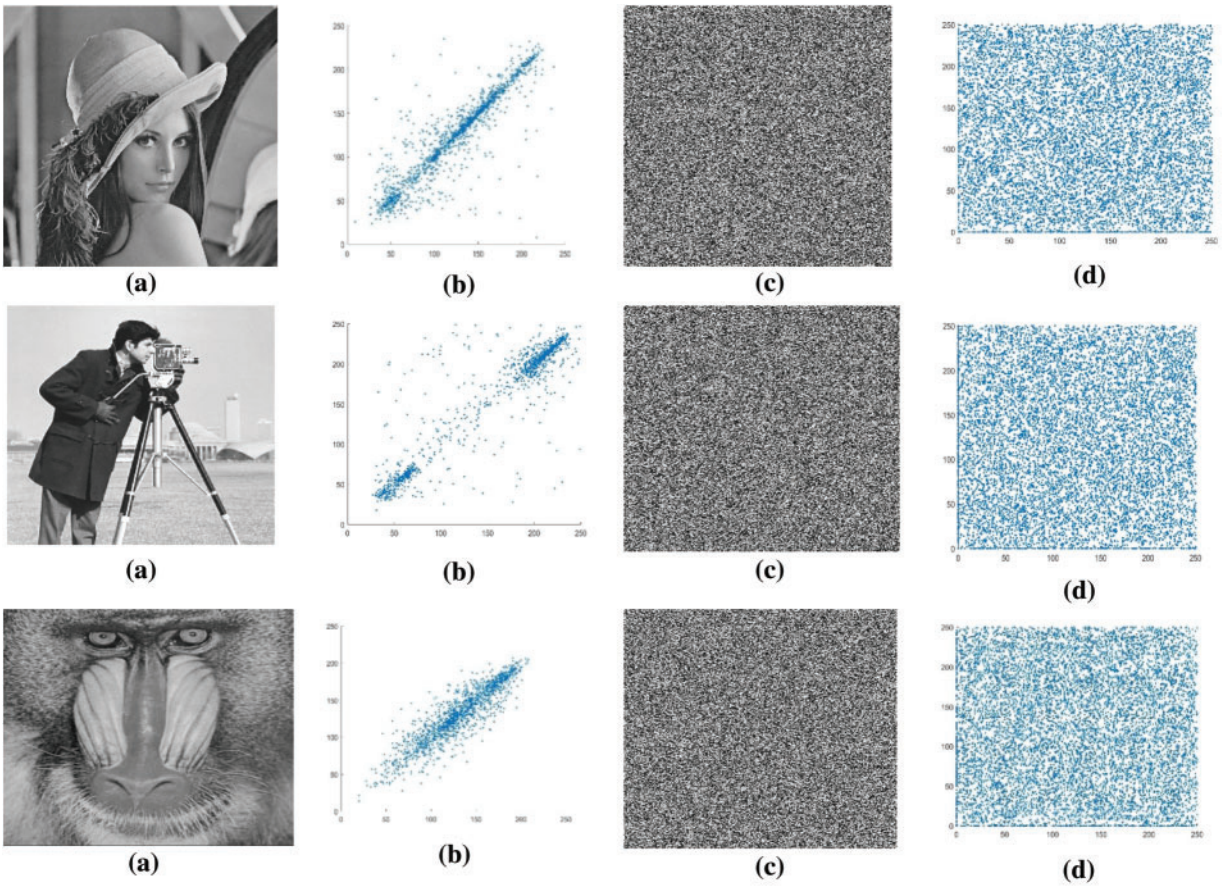

Figure 4: (a) Original images. Lena, Cameraman, and Baboon. (b) Horizontal correlation of original images. Lena, Cameraman, and Baboon. (c) Encrypted images. Lena, Cameraman, and Baboon. (d) Horizontal correlation of encrypted images. Lena, Cameraman, and Baboon

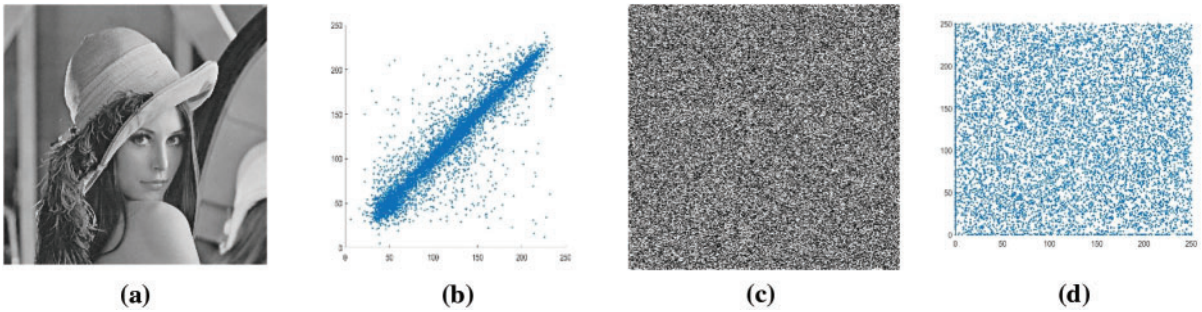

Figure 5: (Continued)

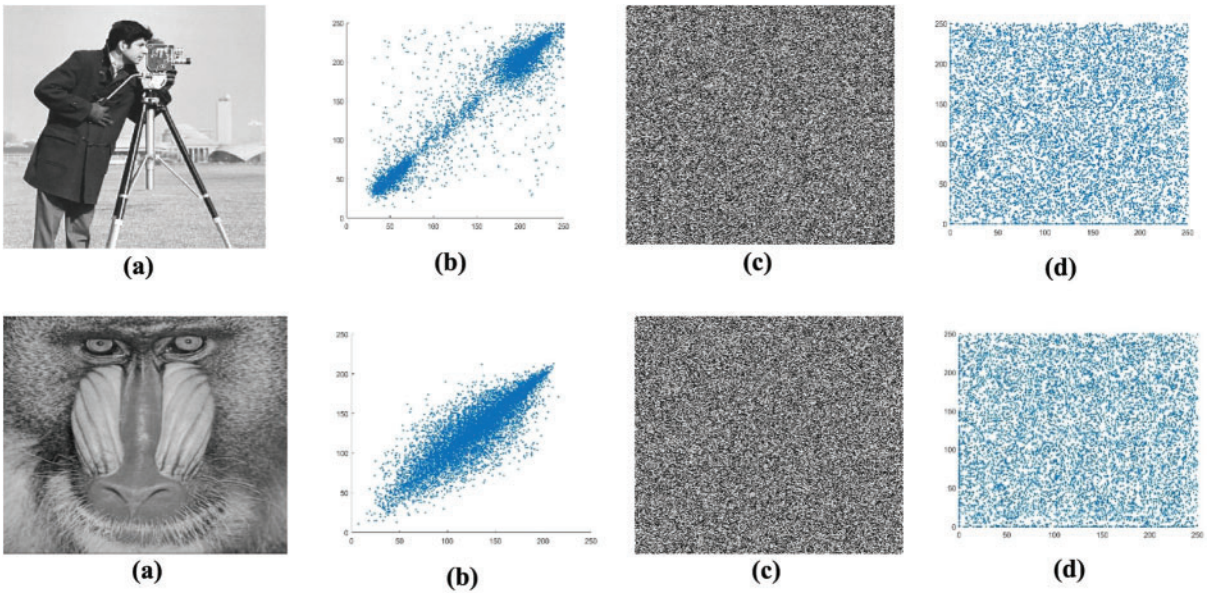

Figure 5: (a) Original images. Lena, Cameraman, and Baboon. (b) Vertical correlation of original images. Lena, Cameraman, and Baboon. (c) Encrypted images. Lena, Cameraman, and Baboon. (d) Vertical correlation of encrypted images. Lena, Cameraman, and Baboon

Table 6: Contrast, energy, and homogeneity analysis and their comparison with existing schemes

Test images	Contrast	Energy	Homogeneity
Lena	10.38	0.0156	0.3813
Baboon	10.48	0.0156	0.3823
Cameraman	10.52	0.156	0.3901
Ref [37]	9.9955	0.0158	0.3948
Ref [38]	9.9764	0.0161	0.4171
Ref [39]	10.3986	0.0158	0.4214
Ref [40]	9.8198	0.0163	0.4228
Ref [41]	9.99240	0.0156	0.3887

4.6 Energy

The GLCMs of the encrypted image are used in the energy analysis of an image. The energy test calculates the square root of the angular second moment to determine pixel intensity uniformity. To calculate energy, apply the following mathematical equation.

$$E^* = \sum_{i,j} f\,(i,j)^2 \tag{13}$$

where $f\ (i,j)$ represents the number of GLCM of the image. The energy score of proposed cipher images is comparable to many existing schemes given in Table 6, which describes the worth of the proposed encryption scheme.

4.7 Homogeneity

Images have intrinsically dispersed contents when seized. GLCM dispersed elements are compared to the GLCM diagonal using this methodology to see how closely they are related. Additionally, a gray-tone spatial dependence matrix of it has been recorded. The following equation represents the search for homogeneity mathematically

$$H^* = \sum_i \sum_j \frac{g\ (i,j)}{|i-j|} \tag{14}$$

The homogeneity score for Lena's ciphered image is deficient, as depicted in Table 6. As a result, it implies that the GLCM difference is more considerable.

4.8 Key Space

A crucial element of a cryptosystem's security is the size of its key-space. A higher cardinality in the set of keys used in the encryption algorithm makes it more resistant to brute-force attacks, also known as exhaustive key search. This new scheme employs the order, base fields of elliptic curves, coefficients of the elliptic curve, and inverse function as secret keys. The article provides a broad concept that relies mainly on the GF and the corresponding bit values of the image. A larger key space can be achieved by increasing the order of the binary extension and using images with higher bit values. Furthermore, the use of XOR and inverse functions in the encryption process makes it more difficult for attackers to guess the key. The algorithm also uses ECC over GF, which offers a high level of security and makes it resistant to attacks based on prime factorization. Overall, the key space analysis indicates that the proposed image encryption algorithm is highly secure and offers strong protection against various types of attacks by taking higher order of the binary extension and images with higher bit values.

4.9 Resistant to Cryptographic Attacks

In the proposed scheme, an algorithm for image encryption is designed combining an elliptic curve, a hill cipher, and an invertible function under the Galois field. This algorithm has been designed to resist various types of attacks, including ciphertext-only, known-plaintext, chosen-plaintext, and chosen-ciphertext attacks. The EC provides a secure key exchange mechanism, while the hill cipher permutes the plaintext and confuses the encryption process. The 2-time invertible function under the Galois field provides diffusion to ensure that any changes made to the ciphertext will significantly impact the decrypted image. Using these three cryptographic primitives in combination provides a high level of security and makes it difficult for an attacker to break the encryption. Additionally, using an EC in the key exchange process ensures that the algorithm resists attacks. Overall, this proposed algorithm is a robust and effective method for image encryption that provides resistance to various attacks. With the increasing importance of secure image transmission in today's digital world, this algorithm is an essential contribution to the field of cryptography.

4.10 Noise Analysis

In this section, we examine the effectiveness of the encryption-decryption algorithm in the presence of noise. When multimedia data is transmitted through a communication channel, various types of noise can cause distortion or errors. Consequently, to evaluate the decryption efficiency of the

proposed scheme, we intentionally added some noise to the encrypted image before sending it through the communication channel. The different types of noise added for this purpose are discussed briefly below.

Salt and pepper noise, also known as impulsive or fat-tail distribution, causes sudden and sharp disturbances in an image's dark and bright regions, resulting in randomly scattered dark and white pixels. Bit errors typically cause this noise during signal transmission or analogue-to-digital signal conversion. Thus, several algorithms and techniques like non-local means, block-matching 3D filtering (BM3D), dark frame subtraction, and interpolation are used to remove salt and pepper noise. In this study, we added salt and pepper noise, Speckle noise to the encrypted image of Apple and then decrypted the noisy encrypted images, as shown in Figs. 6 and 7. Figs. (a–c) depict the noisy encrypted images, while Figs. (d–f) display the corresponding decrypted images. Hence, despite the presence of noise in the encrypted images, the decrypted images are still recognizable, as seen from the figures.

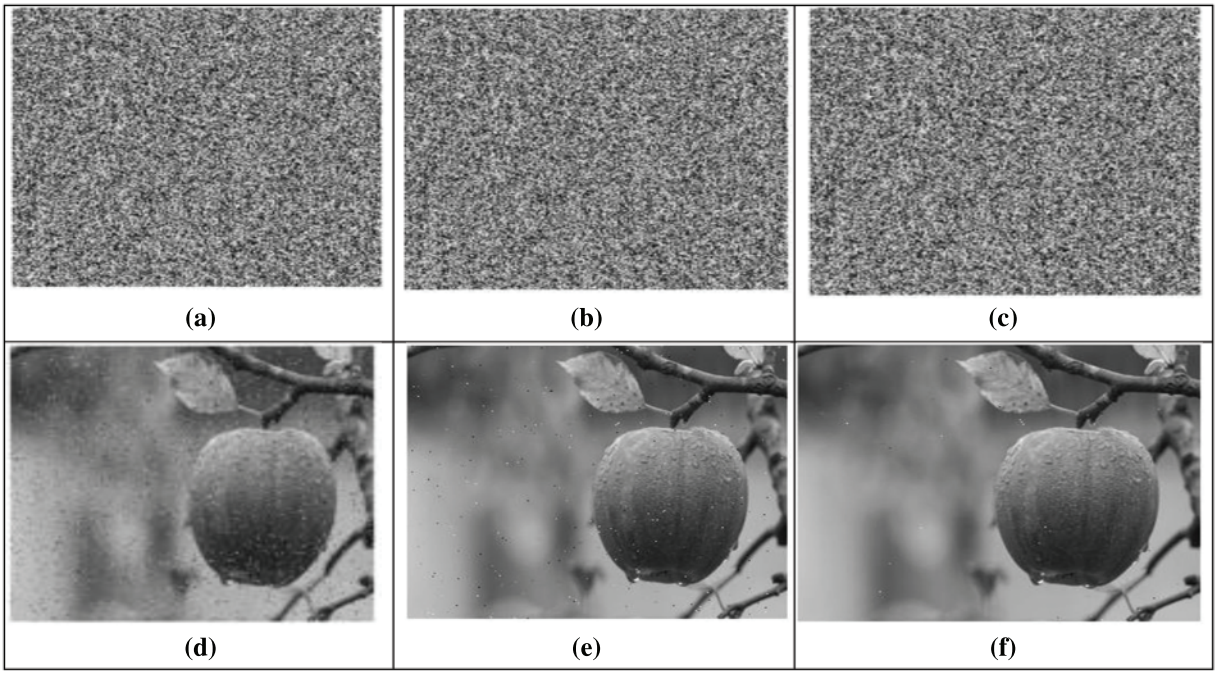

Figure 6: Slat and Peppers analysis of Apple image: (a–c) Apple ciphered image with salt and pepper variance 0.05, 0.005, 0.0005 and 0.5. (d–f) Corresponding deciphered images

4.11 National Institute of Standard and Technology (NIST)

The NIST testing suite comprises of various tests used to determine the randomness of sequences, which can be generated using different techniques such as cryptographic algorithms. In 2001, collaborative efforts between the NIST statistics and computer security departments led to the publication of the NIST testing suite. Table 7 shows the simulation outcomes of the testing tool applied to the pixels of the encrypted image. In this research study, our pseudo random number sequences (ES-PRNS) are evaluated by setting $\beta = 0.01$, which imply that a sequence is accepted as random with confidence 0.99 unless its P value is greater than 0.01.

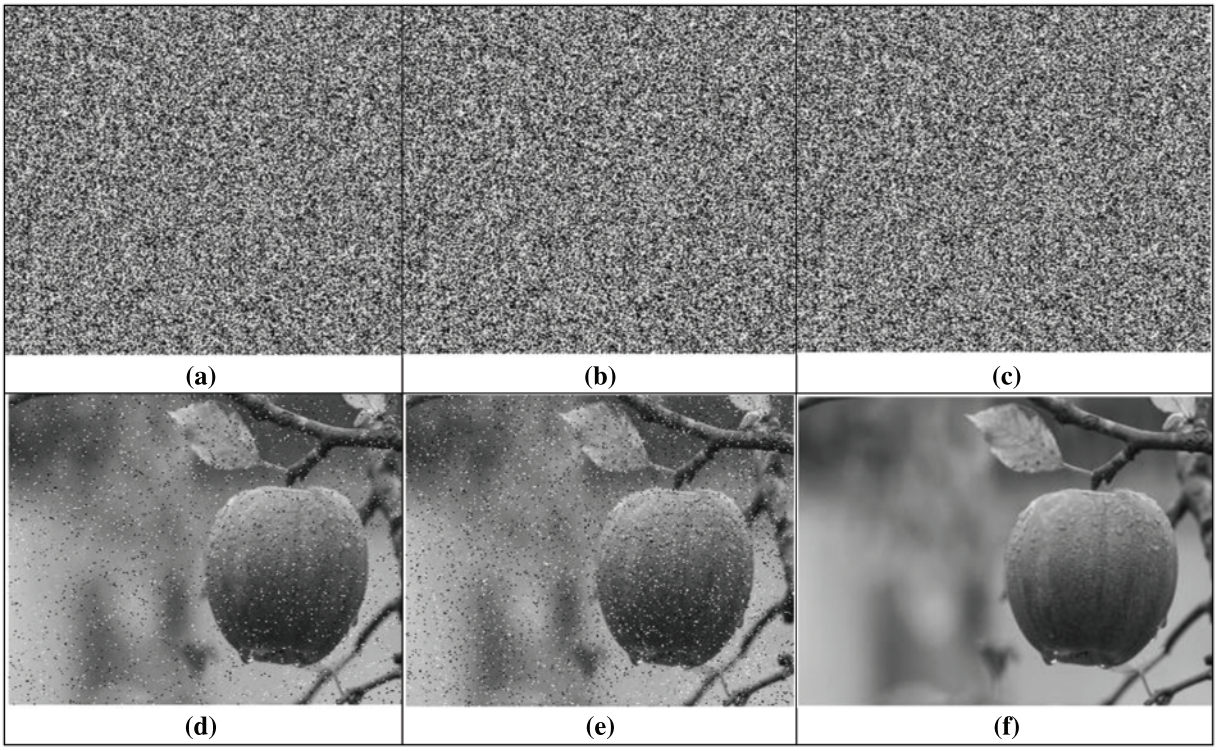

Figure 7: Speckle analysis of Apple image: (a–c) Apple ciphered image with noise addition 0.003, 0.0003, 0.00003 and 0.5. (d–f) Corresponding deciphered images

Table 7: NIST statistical analysis

No.	Test type	*p*-value	Random/Non-random
1	Frequency test (Monobit)	0.003747845769769	Non-random
2	Frequency test	0.906478765333257	Random
3	Discrete Fourier	0.313236745633209	Random
4	Longest Run T	0.713281498657333	Random
5	Run T	0.653287524174435	Random
6	Non-overlapping	0.822929875335104	Random
7	Overlapping	0.078167876305817	Random
8	Maurer's universal	0.735308753493466	Random
9	Binary rank T	0.845488796572692	Random
10	Linear complexity	0.891251897344084	Random
11	Approximate entropy	0.554646472097093	Random
12	Cumulative sums (Forward)	0.003877855744688	Non-random
13	Cumulative Sums (Reverse)	0.003	Non-Random

(Continued)

Table 7 (continued)

No.	Test type		*p*-value	Random/Non-random
14	Serial		0.206874411	Random
			0.515833023	Random
15	Random excursions test:			
	State	Chi-Squared	*P*-value	
	–4	1.05	0.917792279583921	Random
	–3	1.93	0.913223579583945	Random
	–2	0.88	0.575957235883537	Random
	–1	3.70	0.051954758584633	Random
	1	11.02	0.096329744538283	Random
	2	9.3	0.066554763775373	Random
	3	10.26	0.627889784392978	Random
	4	3.50	0.434564333243565	Random
16	Random excursions variant test:			
	State	Counts	*P*-value	
	–9	292	0.352413954635032	Random
	–8	306	0.474512344665062	Random
	–7	325	0.446723774673953	Random
	–6	328	0.619434236569726	Random
	–5	376	0.515447152837526	Random
	–4	354	0.382955453828235	Random
	–3	320	0.619387544538283	Random
	–2	332	0.763276523887282	Random
	–1	324	0.256476487646874	Random
	1	340	0.067858958949856	Random
	2	296	0.043875382591195	Random
	3	340	0.118247128295985	Random
	4	387	0.929383873673396	Random
	5	385	0.728768939838630	Random
	6	385	0.827639823929865	Random
	7	343	0.829849823982988	Random
	8	348	0.729502363702783	Random
	9	269	0.5992566438783839	Random

5 Conclusion

In this article, we combined EC over GF with a traditional HC algorithm. The ECC approach creates a new encryption and decryption key which offers increased security because it is not shared over the internet. As XOR is used for encryption and decryption, which reduces calculation time and complexity with parallel increased speed, it eliminates the requirement to determine the invertible key matrix during decryption. It also reduces the computational work by employing the GF instead of large primes. Moreover, the use of inverse functions in place of substitution techniques makes the encryption algorithm more resistant to cryptographic attacks because inverse functions are more complex and

challenging to reverse-engineer. From a futuristic point of view, XOR and inverse functions can be easily combined with other encryption techniques to create new and more robust encryption algorithms, making them a flexible and adaptable option for image encryption. In addition, this idea can also be extended to generate multiple initial matrices and use them in different cryptographic protocols.

Funding Statement: The authors extend their gratitude to the deanship of Scientific research at King Khalid University for funding this work through the research group's program under Grant Number R. G. P. 2/5/44.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Thein, H. A. Nugroho, T. B. Adji and I. M. Mustika, "Comparative performance study on ordinary and chaos image encryption schemes," in *2017 Int. Conf. on Advanced Computing and Applications (ACOMP)*, Ho Chi Minh, Vietnam, IEEE, pp. 122–126, 2017.

[2] U. Erkan, A. Toktas and Q. Lai, "2D hyperchaotic system based on Schaffer function for image encryption," *Expert Systems with Applications*, vol. 213, pp. 1–12, 2023.

[3] Q. Lai, G. Hu, U. Erkan and A. Toktas, "A novel pixel-split image encryption scheme based on 2D Salomon map," *Expert Systems with Applications*, vol. 213, pp. 118845, 2023.

[4] U. Erkan, A. Toktas and Q. Lai, "Design of two-dimensional hyperchaotic system through optimization benchmark function," *Chaos, Solitons & Fractals*, vol. 167, pp. 1–13, 2023.

[5] U. Erkan, A. Toktas, F. Toktas and F. Alenezi, "2D eπ-map for image encryption," *Information Sciences*, vol. 589, pp. 770–789, 2022.

[6] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li *et al.,* "Asynchronous updating Boolean network encryption algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 1, 2023.

[7] S. Gao, R. Wu, X. Wang, J. Wang, Q. Li *et al.,* "A 3D model encryption scheme based on a cascaded chaotic system," *Signal Processing*, vol. 202, pp. 1–13, 2023.

[8] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li *et al.,* "Encryption for face recognition based on the chaos and semi-tensor product theory," *Information Sciences*, vol. 621, pp. 766–781, 2023.

[9] W. Rui, S. Gao, X. Wang, S. Liu, Q. Li *et al.,* "AEA-NCS: An audio encryption algorithm based on a nested chaotic system," *Chaos, Solitons & Fractals*, vol. 165, pp. 112770, 2022.

[10] V. S. Miller, "Use of elliptic curves in cryptography," in *Conf. on the Theory and Application of Cryptographic Techniques*, Berlin, Germany, Springer, pp. 417–426, 1986.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[12] Z. E. Dawahdeh, S. N. Yaakob and R. R. B. 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.

[13] X. Zhang, G. Zhu, W. Wang and M. Wang, "Design and realization of elliptic curve cryptosystem," in *2012 Int. Symp. on Instrumentation & Measurement, Sensor Network and Automation (IMSNA)*, Sanya, China, IEEE, pp. 302–305, 2012.

[14] K. Rabah, "Elliptic curve cryptography over binary finite field GF (2^m)," *Information Technology Journal*, vol. 5, no. 1, pp. 204–229, 2006.

[15] 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, no. 1, pp. 14–21, 2007.

[16] 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, no. 1, pp. 242–247, 2010.
[17] G. Hamissa, A. Sarhan, H. Abdelkader and M. Fahmy, "Securing JPEG architecture based on enhanced chaotic hill cipher algorithm," in *The 2011 Int. Conf. on Computer Engineering & Systems*, Cairo, Egypt, IEEE, pp. 260–266, 2011.
[18] H. T. Panduranga, "Advanced partial image encryption using two-stage hill cipher technique," *International Journal of Computer Applications*, vol. 60, no. 16, pp. 14–19, 2012.
[19] K. Agrawal and A. Gera, "Elliptic curve cryptography with hill cipher generation for secure text cryptosystem," *International Journal of Computer Applications*, vol. 106, no. 1, pp. 18–24, 2014.
[20] A. Mahmoud and A. Chefranov, "Hill cipher modification based on pseudo-random eigenvalues," *Applied Mathematics & Information Sciences*, vol. 8, no. 2, pp. 505–516, 2014.
[21] S. Sun and Y. Guo, "A novel image steganography based on contourlet transform and hill cipher," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 5, pp. 889–897, 2015.
[22] S. K. Naveenkumar and H. T. Panduranga, "Chaos and hill cipher-based image encryption for mammography images," in *2015 Int. Conf. on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, IEEE, pp. 1–5, 2015.
[23] Y. Sazaki and R. S. Putra, "Implementation of affine transform method and advanced hill cipher for securing digital images," in *2016 10th Int. Conf. on Telecommunication Systems Services and Applications (TSSA)*, Denpasar, Indonesia, IEEE, pp. 1–5, 2016.
[24] L. Goutham, M. S. Mahendra, A. P. Manasa and S. N. Prajwalasimha, "Modified hill cipher based image encryption technique," *International Journal for Research in Applied Science & Engineering Technology*, vol. 5, no. 4, pp. 342–345, 2017.
[25] S. Farwa, A. Sohail and N. Muhammad, "A novel application of elliptic curves in the dynamical components of block ciphers," *Wireless Personal Communications*, vol. 115, no. 2, pp. 1309–1316, 2020.
[26] D. Hankerson, A. J. Menezes and S. Vanstone, "Elliptic curve arithmetic," in *Guide to Elliptic Curve Cryptography*. New York, USA: Springer Science & Business Media, no. 3, pp. 75–112, 2006.
[27] J. Hoffstein, J. Pipher and J. H. Silverman, "Elliptic curves and cryptography," in *An Introduction to Mathematical Cryptography*. New York, USA: Springer, no. 5, pp. 299–371, 2014.
[28] J. H. Silverman, "The arithmetic of elliptic curves," in *Graduate Texts in Mathematics*, New York, USA: Springer, vol. 106, no. 2, 2009.
[29] L. S. Hill, "Cryptography in an Algebraic Alphabet," *The American Mathematical Monthly*, vol. 36, no. 6, pp. 306–312, 1929.
[30] C. K. Volos, I. M. Kyprianidisand and I. N. Stouboulos, "Image encryption process based on chaotic synchronization phenomena," *Signal Processing*, vol. 93, no. 5, pp. 1328–1340, 2013.
[31] X. Wei, L. Guo, Q. Zhang, J. Zhang and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Journal of Systems and Software*, vol. 85, no. 2, pp. 290–299, 2012.
[32] X. Wang, C. Liu and D. Xu, "Image encryption scheme using chaos and simulated annealing algorithm," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1417–1429, 2016.
[33] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography*. New York, USA: CRC Press, 2008. [Online]. Available: https://doi.org/10.1201/9781420071474
[34] K. Kedarisetti, R. Gamini and V. Thanikaiselvan, "Elliptical curve cryptography for images using fractal based multiple key kill cipher," in *2018 Second Int. Conf. on Electronics, Communication and Aerospace (ICECA)*, Coimbatore, India, IEEE, pp. 643–649, 2018.
[35] M. A. Bakr, M. A. Mokhtar and A. E. S. Takieldeen, "Elliptic curve cryptography modified Hill Cipher dependent on circulant matrix," *International Journal of Industrial Electronics and Electrical Engineering*, vol. 6, no. 1, pp. 24–29, 2018.
[36] X. Wang, X. Zhu and Y. Zhang, "An image encryption algorithm based on Josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23733–23746, 2018.

[37] H. U. Rehman, T. Shah, A. Aljaedi, M. M. Hazzazi and A. R. Alharbi, "Design of nonlinear components over a mordell elliptic curve on Galois fields," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1313–1329, 2022.
[38] S. Hussain, S. S. Jamal, T. Shah and I. Hussain, "A power associative loop structure for the construction of non-linear components of block cipher," *IEEE Access*, vol. 8, pp. 123492–123506, 2020.
[39] M. I. Haider, A. Ali, D. Shah and T. Shah, "Block cipher's nonlinear component design by elliptic curves, an image encryption application," *Multimedia Tools and Applications*, vol. 1, no. 80, pp. 4693–4718, 2021.
[40] Y. Naseer, T. Shah, D. Shah and S. Hussain, "A novel algorithm of constructing highly nonlinear Sp-boxes," *Cryptography*, vol. 3, no. 1, pp. 1–13, 2019.
[41] H. U. Rehman, T. Shah, M. M. Hazzazi, A. Alshehri and B. Zaid, "Mordell elliptic curve-based design of nonlinear component of block cipher," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 2913–2930, 2022.