Research Article

Hiba Hilal Hadi* and Ammar Ali Neamah

An image encryption method based on modified elliptic curve Diffie-Hellman key exchange protocol and Hill Cipher

https://doi.org/10.1515/eng-2022-0552
received September 01, 2023; accepted November 01, 2023

Abstract: Digital image protection is crucial since images often contain private and sensitive information in business, medical, and military. One of the best techniques for securing the content of these images is encryption. This article introduces a cryptosystem known as the elliptic curve Diffie-Hellman Hill Cipher (ECDHHC) that uses the modified eliptic curve Diffie-Hellman (ECDH) key exchange protocol to generate the shared secret key integrated with the Hill Cipher. An elliptic curve point-based secret shared key matrix using the ECDHHC, which will be used for encryption and decryption, is generated. Thereafter, the input image is split into a set of 8 × 8 submatrices and then changes the values of these matrices by multiplying each block with the secret shared key matrix. The encrypted image is constructed by merging all encrypted blocks. With this combination, the correlation between adjacent pixels in the ciphered image is effectively removed, and the level of unpredictability and uncertainty for the ciphered image is also enhanced. The suggested approach used the key space, entropy, histogram, antinoise attack, differential attack, and correlation coefficient to evaluate the performance of the encryption method. According to simulation findings, the proposed method offers a high level of security and efficiency, and resists attackers.

Keywords: image encryption, elliptic curve Diffie-Hellman key exchange protocol, Hill Cipher, security

1 Introduction

One mathematical technique employed to secure images from attacks and raise communication security is cryptography. Cryptography is divided into two main categories: symmetric key encryption and asymmetric key encryption. Symmetric one uses a shared key between the sender and the receiver for encryption and decryption. In contrast, asymmetric cryptography uses sender's public key to encrypt the plaintext and recipient's private key for decryption. Several approaches for multimedia data encryption, particularly digital images, were suggested and developed by researchers, such as DNA based [1–3], cellular automata based [4,5], chaotic based [6–8], and elliptic curves based [9–14]. Elliptic curve cryptography (ECC) is a robust public key cryptography algorithm introduced independently by Miller [15] and Koblitz [16]. One of the ECC benefits is the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP) from the attackers. Compared to other systems, such as RSA, ECC uses a tiny key size, a slight amount of memory, and low-power consumption [17].

Zhang and Wang [18] proposed an asymmetric image encryption method based on ECC. To produce the secret key, the Diffie-Hellman protocol is used. Permutation and diffusion are also performed using the chaotic system in conjunction with ECC. Hayat and Azam [19] introduced a method for encrypting an image using an elliptic curve (EC) based on pseudorandom integers and substitution boxes. Díaz et al. [20] suggested an effective encryption technique based on chaos and EC to encrypt BMP images without compression. Obaid and Alsaffar [21] proposed an image encryption technique based on ECC combined with Hilbert matrices of dimensions 2 × 2 and 4 × 4. Liang et al. [22] suggested a public key image encryption technique based on ECC, in which the hash value produced from the plain image was ciphered using ECC. Abbas et al. [23] suggested a chaotic image encryption method using an addition operator over the EC points to produce a discrete chaotic sequence that can then be used to build the

* **Corresponding author: Hiba Hilal Hadi,** Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq,
e-mail: alsalamyheba@gmail.com
Ammar Ali Neamah: Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq,
e-mail: ammara.meamah@uokufa.edu.iq

encryption scheme. Hayat et al. [24] suggested an efficient cryptographic system depending on an EC over finite rings and S-boxes. Castro et al. [25] introduced a hybrid asymmetric encryption technique based on ECC and AES employed to encrypt the medical image and fingerprint feature vector.

The Hill Cipher (HC), a traditional substitution cipher devised by Lester Hill in 1929, provides efficiency with a simple structure, is fast, and can shuffle plaintext [26]. The technique is classified as a security system due to the ability to encrypt and decode in a minimal period; however, it has security limitations since it is a symmetric cryptographic system. Many scholars sought to improve the security of the technique by proposing a new HC. Dawahdeh et al. [9] introduced an image encryption method integrating HC techniques and elliptic curves (ECCHC). The aim was to convert the HC approach from symmetric to asymmetric using ECC parameters to generate the private key. However, the method has a severe loophole in the form of private keys, which renders it subject to brute-force attacks. Ismail and Misro [14] suggested a modified ECCHC technique that combines a cubic Bézier coefficient matrix and HC to raise the image encryption/decryption level of security. It has over 2^{100} key search, making it incredibly difficult for attackers to predict the keys required to break their scheme. However, they focused on encryption and decryption operations while overlooking computational complexity.

Related studies have some limitations such as small key space and the incapacity to fend against statistical and differential assaults. These drawbacks motivated us to suggest an efficient method based on the modified elliptic curve Diffie-Hellman (ECDH) key exchange algorithm to encrypt images. Thus, the study introduces the elliptic curve Diffie-Hellman Hill Cipher (ECDHHC) cryptosystem, which generates the shared secret key integrated with the HC using the ECDH key exchange algorithm to protect images from unauthorized usage. Furthermore, if the $M_{4\times2}(E(F_p))$ matrix group is used, our approach can attain the best results when compared to the existing methodologies.

The contributions of this study are outlined below:

- The first use of the ECDH key exchange protocol using an 8 × 8 self-invertible matrix.
- The self-invertible matrix K_m of size 8 × 8 is suggested to be combined with the modified ECDH key exchange protocol.
- The fact that matrix's inverse does not always exist, which has an impact on the decryption procedure, is one of the fundamental shortcomings of the matrix produced by the modified ECDH key exchange protocol. To get around this issue, we used a self-invertible matrix, which increases the unpredictable nature of the pixel distribution during encryption while also lowering the computational operations during decryption.
- The proposed key matrix K_m increases the size of the key space to 2^{768}, which prevents brute-force attacks.
- A high level of security is provided by combining the HC with the matrix K_m.
- The performance of the proposed approach in terms of robustness, efficacy, and resistance to cryptanalysis is evaluated against all common attacks.

The article's remaining sections are organized as follows: Section 2 introduces the preliminaries of this study. The encryption structure of Dawahdeh's system [9] is highlighted in Section 3. Section 4 presents the proposed image encryption technique with the implementation example given in Section 5. In Section 6, simulation results and security evaluation are introduced. Finally, the conclusion is presented in Section 7.

2 Preliminaries

2.1 Elliptic curve cryptography (ECC)

Let $E_p(a, b)$ be an equation for an elliptic curve in a finite field F_p, where $E_p(a, b)$ is written as follows:

$$y^2 = x^3 + ax + b \bmod p, \tag{1}$$

where a and b are two constants that fulfill $4a^3 + 27b^2 \bmod p \neq 0$, and p is a prime number. The set of points (x, y) satisfying equation (1) and the point at infinity O_∞ constitute the elliptic curve group $E(F_p)$ [27].

2.2 Elliptic curve arithmetic operations

The elliptic curve scalar multiplication, which takes up the most time in encryption and decryption processes, is one of the primary operations connected to the elliptic curve function. Calculating the elliptic curve scalar multiplication requires two operations: point addition and point doubling [28].

Point addition and doubling: Assume $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ belong to $E(F_p)$. Then $P_1 \oplus P_2 = (x_3, y_3)$, which is also a point on $E(F_p)$, can be defined as follows:

$$x_3 = \lambda^2 - x_1 - x_2,\ y_3 = \lambda(x_1 - x_3) - y_1,$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \dfrac{3x_1^2}{2y_1} & \text{if } P_1 = P_2. \end{cases} \tag{2}$$

If $P_2 = \ominus P_1$, then $P_1 \oplus P_2 = O_\infty$, where $\ominus P_1 = (x_1, -y_1)$ is the inverse of the point P_1.

Scalar multiplication: Let P_1 be any point on the elliptic curve $E_p(a, b)$. Then the repeated addition of the point P_1 to itself k times defines the scalar multiplication operation over P_1. It is expressed as usual by

$$[k]\underbrace{P_1 = P_1 \oplus P_1 \oplus P_1 \oplus \ldots \oplus P_1}_{k \text{ times}}. \tag{3}$$

2.3 Proposed elliptic curve Diffie Hellman (ECDH) key exchange protocol

Two parties A and B can create a shared secret key using the ECDH key exchange protocol across an insecure channel. This protocol is based on the original Diffie-Hellman (DH) agreement, which was established by Diffie and Hellman in the mid-1970s [29]. The protocol's security is dependent on the difficulty of computing discrete logarithms on elliptic curves (ECDLP), which is currently regarded to be an intractable issue.

In 2023, Hadi and Neamah [30] introduced a development of the ECDH protocol by combining it with the matrix concept. The domain parameters of the protocol are $(M_{m\times n}(E(F_p)), \beta)$, where $M_{m\times n}(E(F_p))$ is a matrix-group with m rows and n columns whose entries are points from $E(F_p)$. $E(F_p)$ is an elliptic curve group with parameters a and b, F_p is a prime field, and $\beta = [P_{ij}]$ is a base matrix such that the number of points of the elliptic curve is prime (*i.e.*, all P_{ij} are generators of $E(F_p)$, for $i = 1, \ldots, m, j = 1, \ldots, n$). The description of the proposed protocol is as follows:

- Party A randomly selects his private matrix D_A whose elements are integers and calculates his public key $P_A = D_A \odot \beta$ and sends it to Party B.
- Party B also selects his private matrix D_B such that D_B is the same as the size of the matrix β and calculates his public key $P_B = D_B \odot \beta$ and sends it to Party A.
- Then, both parties (A and B) secretly compute the shared secret key, K, as follows:

$$K = D_A \odot P_B = D_B \odot P_A, \tag{4}$$

where $\odot$ represents elementwise multiplication operation.

2.4 HC

HC, a polyalphabetic substitution cipher based on linear algebra, was invented by Lester Hill in 1929 [26]. The key matrix used in this method, $K_{(m\times m)}$, will be the same by all parties engaged in encryption and decryption. The ordinary readable text is split into m-blocks that fulfill the key matrix size, K, by allocating a numerical value to each letter so that $a = 0$, $b = 1$, and so on until $z = 25$. For instance, if the ordinary readable text O has a block size of 8×1, then the invertible key matrix, which will be used for the encryption and decryption process, will be $K_{8\times 8}$. Here, the encryption procedure will result in an encrypted text block of size 8×1 as follows:

$$C = K \times O \pmod{26},$$

$$C = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & k_{37} & k_{38} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & k_{47} & k_{48} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & k_{57} & k_{58} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & k_{67} & k_{68} \\ k_{71} & k_{72} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} & k_{78} \\ k_{81} & k_{82} & k_{83} & k_{84} & k_{85} & k_{86} & k_{87} & k_{88} \end{bmatrix} \tag{5}$$

$$\times \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \end{bmatrix} (\text{mod}\, 26) = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}.$$

Once the receiver obtains C, they can decode the encrypted text by finding K^{-1} so that $O = K^{-1} \times C (\text{mod} 26)$ to obtain the original message, O.

3 Elliptic curve cryptosystem Hill Cipher (ECCHC)

This system was established by transforming the HC from secret-key cryptography to a public-key cryptography technique for improved security and effectiveness [10]. Consider a scenario in which the sender (Party A) wishes to communicate with the recipient (Party B) using ECCHC over an unsecured channel. They must first agree on the

elliptic curve $E(F_p)$ with parameters a and b, and F_p is a finite field such that p is a large prime, and G is the generator point. Then, each party must specify their secret key, d_A and d_B, as $d_A, d_B \in [1, 1-p]$, which will be employed to compute their public keys, $P_A = [d_A]G$, $P_B = [d_B]G$. Then, by utilizing their respective secret keys along with their public keys, both parties (A and B) will secretly compute the first symmetric key, K_i, as follows:

$$K_i = [d_A]P_B = [d_B]P_A = (x, y).$$

After that, both of them must compute K_1 and K_2 as follows:

$$K_1 = [x]G = (k_{11}, k_{12}),$$
$$K_2 = [y]G = (k_{21}, k_{22}),$$

K_1 and K_2 will then be utilized to produce the secret key matrix, K_m for encryption, and K_m^{-1} for decryption. However, since K_m's inverse is not always present, the problem can be solved by using a self-invertible matrix, K_m, as the key matrix [31]. Therefore, since $K_m = K_m^{-1}$, the same K_m will be utilized for the encryption/decryption process.

Let $K_m = \left[\begin{array}{cc|cc} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ \hline k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{array}\right]$ be a self-invertible matrix, which can be divided as $K_m = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$. The suggested method makes that $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$, and then the values of the remaining partitions of the secret matrix key K_m are determined by solving $K_{12} = I - K_{11}$, $K_{21} = I + K_{11}$, and $K_{11} + K_{22} = 0$, where I is the identity matrix.

4 The proposed scheme

This study offers a modification of ECCHC to give an additional degree of security in the encryption/decryption of images by using the Diffie-Hellman key exchange protocol based on block matrices combined with elliptic curves in ECCHC. The proposed improved protocol was recently introduced by Hadi and Neamah [30]. This improvement is intended to increase the efficiency and the key space. Suppose Party A and Party B agree upon an $E(F_p)$ with parameters $a$and$b$, and a base matrix $\beta \in M_{4\times 2}(E(F_p))$ such that the number of points of the curve E is a prime number. Then each party must specify their secret key, D_A and D_B whose elements are integers with the same size of the matrix β, which will be employed to compute their public keys, $P_A = D_A \odot \beta$, $P_B = D_B \odot \beta$. Then, by utilizing their respective secret keys along with their public keys, both parties (A and B) will secretly compute the secret shared key, K, as follows:

$$K = D_A \odot P_B = D_B \odot P_A \pmod{256} = \begin{bmatrix} (k_{11}, k_{12}) & (k_{13}, k_{14}) \\ (k_{21}, k_{22}) & (k_{23}, k_{24}) \\ (k_{31}, k_{32}) & (k_{33}, k_{34}) \\ (k_{41}, k_{42}) & (k_{43}, k_{44}) \end{bmatrix}. \quad (6)$$

The key K will then be utilized to produce the secret key matrix K_m for encryption and K_m's inverse for decryption. However, since K_m's inverse is not always present, the problem can be solved by using a self-invertible matrix, K_m, as the key matrix [31]. Therefore, since $K_m = K_m^{-1}$, the same matrix K_m will be utilized for both processes (encryption and decryption). Since the definition of an image's pixels ranges from 0 to 255, the suggested technique uses a finite modular field of size 256.

Let

$$K_m(\mathrm{mod}256) = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & k_{37} & k_{38} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & k_{47} & k_{48} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & k_{57} & k_{58} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & k_{67} & k_{68} \\ k_{71} & k_{72} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} & k_{78} \\ k_{81} & k_{82} & k_{83} & k_{84} & k_{85} & k_{86} & k_{87} & k_{88} \end{bmatrix}$$

be a self-invertible matrix, which can be divided as $K_m = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$. The suggested method makes that $K_{11}(\mathrm{mod}256) = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix}$, then the values of the remaining partitions of the secret matrix key K_m are determined by solving $K_{12}(\mathrm{mod}256) = I - K_{11}$, $K_{21}(\mathrm{mod}256) = I + K_{11}$, and $K_{11}(\mathrm{mod}256) + K_{22}(\mathrm{mod}256) = 0$, where I is the identity matrix.

4.1 The process of encryption

4.1.1 Encryption (Party A)

Party A must first divide image's pixels of M into blocks of 8×8 submatrices called $(O_1, O_2, O_3, \ldots)$ to cipher an input image employing the suggested approach. Then, Party A needs to multiply each block by the secret matrix K_m of modulo 256 to obtain all the encrypted blocks $(C_1, C_2, C_3, \ldots)$. After that, the encrypted blocks will be rebuilt into input image's dimensions. Thus, Party B will receive C as an encrypted image. The flowchart for the encryption procedure is shown in Figure 1.

4.2 The process of decryption

4.2.1 Decryption (Party B)

The decryption process flowchart for Party B is shown in Figure 2. When Party B receives the encrypted image, C, he must divide it into blocks of 8×8 submatrices $C_i = (C_1, C_2, C_3, \ldots)$. Then, to decode the encrypted image, Party B must multiply all submatrices, E_i, by K_m^{-1} with a finite modular field of size 256 since $K_m = K_m^{-1}$. This will result in blocks $(O_1, O_2, O_3, \ldots)$, which are used to reconstruct the original image M.

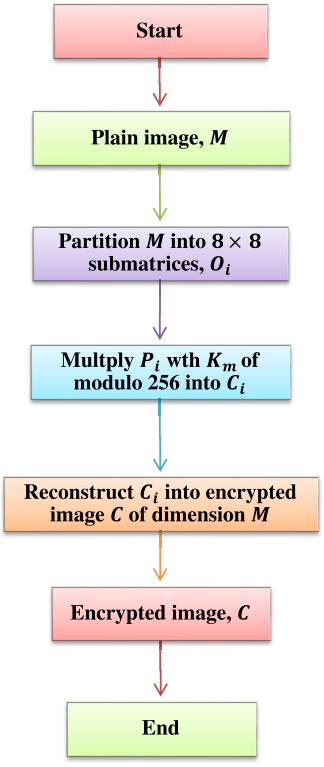

Figure 1: The suggested approach's encryption process.

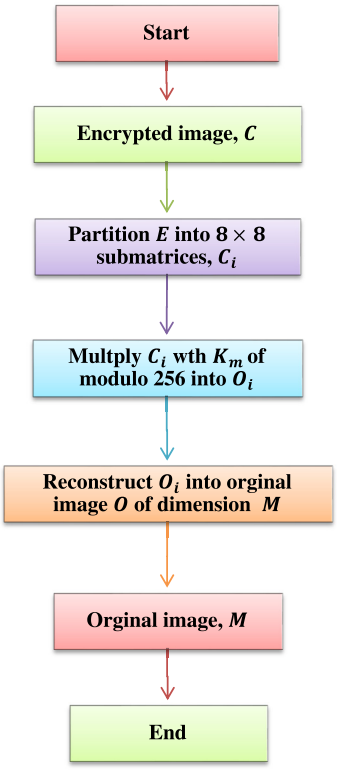

Figure 2: The suggested approach's decryption process.

5 Implementation example

Assume that two parties (A and B) decide to utilize the suggested approach for airplane image sending with size of 512×512. They can reach an agreement to utilize the elliptic curve $E : y^2 = x^3 + x + 15 \bmod 5003$, where $4a^3 + 27b^2 (\bmod p) = 6079 \bmod 5003 = 1076 \neq 0$. Here, all points are a generator for the group $E(F_{5,003})$, since the number of points of the elliptic curve is 5081. Suppose they select

$$\beta = \begin{bmatrix} (5000, 534) & (4864, 3353) \\ (4999, 1221) & (4987, 4842) \\ (4997, 3843) & (4996, 1497) \\ (4995, 4672) & (4991, 91) \end{bmatrix}$$

to be the base matrix.

5.1 Key Generation

Party A (sender): Party A will compute his public key, key $P_A = D_A \odot \beta = \begin{bmatrix} (5000, 534) & (3980, 4365) \\ (3904, 3751) & (4987, 4842) \\ (1402, 2809) & (4501, 1847) \\ (2467, 3574) & (4991, 91) \end{bmatrix}$, by selecting $D_A = \begin{bmatrix} 1 & 4 \\ 3 & 1 \\ 3 & 5 \\ 4 & 1 \end{bmatrix}$ as the private key, and submit it to Party B.

Party B (receiver): Party B will compute his public key, key $P_B = D_B \odot \beta = \begin{bmatrix} (660, 2709) & (2867, 292) \\ (3904, 3751) & (2182, 4483) \\ (4316, 4517) & (4996, 1497) \\ (4563, 1363) & (1197, 332) \end{bmatrix}$, by selecting $D_B = \begin{bmatrix} 2 & 5 \\ 3 & 4 \\ 5 & 1 \\ 3 & 6 \end{bmatrix}$ as the private key, and submit it to Party A.

Shared private keys used by both parties:

Both parties will privately compute the shared private key K using their shared public keys, P_A, P_B and their own secret keys, D_A and D_B as follows:

$$K = D_A \odot P_B = D_B \odot P_A = \begin{bmatrix} (660, 2709) & (4278, 2969) \\ (149, 1479) & (2182, 4483) \\ (235, 368) & (4501, 1847) \\ (3569, 2534) & (1197, 332) \end{bmatrix}.$$

The matrix, K_m, is then produced by finding the following equations so that

$$K_{11} = K \mod 256 = \begin{bmatrix} 660 & 2709 & 4278 & 2969 \\ 149 & 1479 & 2182 & 4483 \\ 235 & 368 & 4501 & 1847 \\ 3569 & 2534 & 1197 & 332 \end{bmatrix}$$

$$\mod 256 = \begin{bmatrix} 148 & 149 & 182 & 153 \\ 149 & 199 & 134 & 131 \\ 235 & 112 & 149 & 55 \\ 241 & 230 & 173 & 76 \end{bmatrix},$$

$$K_{12} = (I - K_{11}) \mod 256 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$- \begin{bmatrix} 148 & 149 & 182 & 153 \\ 149 & 198 & 134 & 131 \\ 235 & 112 & 149 & 55 \\ 241 & 230 & 173 & 76 \end{bmatrix} = \begin{bmatrix} 109 & 107 & 74 & 103 \\ 107 & 58 & 122 & 125 \\ 21 & 144 & 108 & 201 \\ 15 & 26 & 83 & 181 \end{bmatrix},$$

$$K_{21} = (I + K_{11}) \mod 256 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 135 & 251 & 134 & 41 \\ 150 & 25 & 88 & 104 \\ 228 & 247 & 153 & 23 \\ 8 & 237 & 54 & 106 \end{bmatrix} = \begin{bmatrix} 149 & 149 & 182 & 153 \\ 149 & 200 & 134 & 131 \\ 235 & 112 & 150 & 55 \\ 241 & 230 & 173 & 77 \end{bmatrix},$$

$$K_{22} = -K_{11} \mod 256 = \begin{bmatrix} 108 & 107 & 74 & 103 \\ 107 & 57 & 122 & 125 \\ 21 & 144 & 107 & 201 \\ 15 & 26 & 83 & 180 \end{bmatrix},$$

$$K_m = \left[\begin{array}{cccc|cccc} 148 & 149 & 182 & 153 & 109 & 107 & 74 & 103 \\ 149 & 199 & 134 & 131 & 107 & 58 & 122 & 125 \\ 235 & 112 & 149 & 55 & 21 & 144 & 108 & 201 \\ 241 & 230 & 173 & 76 & 15 & 26 & 83 & 181 \\ \hline 149 & 149 & 182 & 153 & 108 & 107 & 74 & 103 \\ 149 & 200 & 134 & 131 & 107 & 57 & 122 & 125 \\ 235 & 112 & 150 & 55 & 21 & 144 & 107 & 201 \\ 241 & 230 & 173 & 77 & 15 & 26 & 83 & 180 \end{array}\right].$$

5.2 Encryption (Party A)

Party A will divide the pixel value of the airplane image into blocks of size eight as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
1	**65**	**199**	**193**	**185**	**179**	**185**	**191**	**189**	*191*	*193*	*200*	*193*	*197*	*199*	*200*	*202*	196	...
2	**54**	**196**	**192**	**195**	**195**	**192**	**191**	**192**	*188*	*193*	*195*	*193*	*190*	*189*	*193*	*191*	190	...
3	**54**	**197**	**194**	**197**	**195**	**191**	**191**	**188**	*186*	*197*	*191*	*189*	*189*	*186*	*188*	*187*	188	...
4	**55**	**187**	**187**	**188**	**186**	**180**	**180**	**182**	*183*	*187*	*183*	*184*	*185*	*189*	*190*	*192*	194	...
5	**62**	**181**	**180**	**180**	**180**	**183**	**182**	**183**	*181*	*181*	*181*	*185*	*185*	*188*	*190*	*187*	190	...
6	**69**	**174**	**174**	**178**	**179**	**179**	**181**	**186**	*182*	*179*	*181*	*178*	*180*	*185*	*189*	*185*	185	...
7	**72**	**180**	**179**	**174**	**174**	**181**	**179**	**180**	*168*	*171*	*175*	*172*	*172*	*180*	*175*	*175*	171	...
8	**77**	**176**	**171**	**168**	**170**	**175**	**169**	**174**	*168*	*173*	*172*	*164*	*163*	*172*	*164*	*165*	163	...
9	85	169	171	166	171	167	161	164	165	160	165	159	153	157	158	162	160	...
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	...

$$O_1 = \begin{bmatrix} 65 & 199 & 193 & 185 & 179 & 185 & 191 & 189 \\ 54 & 196 & 192 & 195 & 195 & 192 & 191 & 192 \\ 54 & 197 & 194 & 197 & 195 & 191 & 191 & 188 \\ 55 & 187 & 187 & 188 & 186 & 180 & 180 & 182 \\ 62 & 181 & 180 & 180 & 180 & 183 & 182 & 183 \\ 69 & 174 & 174 & 178 & 179 & 179 & 181 & 186 \\ 72 & 180 & 179 & 174 & 174 & 181 & 179 & 180 \\ 77 & 176 & 171 & 168 & 170 & 175 & 169 & 174 \end{bmatrix}, O_2 = \begin{bmatrix} 191 & 193 & 200 & 193 & 197 & 199 & 200 & 202 \\ 188 & 193 & 195 & 193 & 190 & 189 & 193 & 191 \\ 186 & 197 & 191 & 189 & 189 & 186 & 188 & 187 \\ 183 & 187 & 183 & 184 & 185 & 189 & 190 & 192 \\ 181 & 181 & 181 & 185 & 185 & 188 & 190 & 187 \\ 182 & 179 & 181 & 178 & 180 & 185 & 189 & 185 \\ 168 & 171 & 175 & 172 & 172 & 180 & 175 & 175 \\ 168 & 173 & 172 & 164 & 163 & 172 & 164 & 165 \end{bmatrix}.$$

Then, the first block will be multiplied by the key K_m of modulo 256. The process will be carried out once again for the other blocks that produce (C_1, C_2, C_3, . . .), so that C_i becomes the encrypted image, and C as follows:

$$C_1 = K_m \times O_1 = \begin{bmatrix} 148 & 149 & 182 & 153 & 109 & 107 & 74 & 103 \\ 149 & 199 & 134 & 131 & 107 & 58 & 122 & 125 \\ 235 & 112 & 149 & 55 & 21 & 144 & 108 & 201 \\ 241 & 230 & 173 & 76 & 15 & 26 & 83 & 181 \\ 149 & 149 & 182 & 153 & 108 & 107 & 74 & 103 \\ 149 & 200 & 134 & 131 & 107 & 57 & 122 & 125 \\ 235 & 112 & 150 & 55 & 21 & 144 & 107 & 201 \\ 241 & 230 & 173 & 77 & 15 & 26 & 83 & 180 \end{bmatrix}$$

$$\times \begin{bmatrix} 65 & 199 & 193 & 185 & 179 & 185 & 191 & 189 \\ 54 & 196 & 192 & 195 & 195 & 192 & 191 & 192 \\ 54 & 197 & 194 & 197 & 195 & 191 & 191 & 188 \\ 55 & 187 & 187 & 188 & 186 & 180 & 180 & 182 \\ 62 & 181 & 180 & 180 & 180 & 183 & 182 & 183 \\ 69 & 174 & 174 & 178 & 179 & 179 & 181 & 186 \\ 72 & 180 & 179 & 174 & 174 & 181 & 179 & 180 \\ 77 & 176 & 171 & 168 & 170 & 175 & 169 & 174 \end{bmatrix}$$

$$= \begin{bmatrix} 217 & 65 & 117 & 153 & 27 & 63 & 11 & 221 \\ 115 & 139 & 79 & 74 & 134 & 249 & 48 & 57 \\ 103 & 158 & 31 & 28 & 146 & 230 & 98 & 101 \\ 224 & 62 & 201 & 61 & 17 & 49 & 166 & 53 \\ 129 & 7 & 75 & 39 & 113 & 101 & 31 & 68 \\ 21 & 202 & 102 & 234 & 227 & 151 & 253 & 130 \\ 231 & 46 & 37 & 119 & 15 & 59 & 65 & 235 \\ 211 & 79 & 255 & 126 & 36 & 16 & 85 & 216 \end{bmatrix}, \ldots$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	217	65	117	153	27	63	11	221	72	53	143	122	60	83	1	17	...
2	115	139	79	74	134	249	48	57	72	113	197	127	50	13	191	1	...
3	103	158	31	28	146	230	98	101	85	86	127	228	34	41	252	148	...
4	224	62	201	61	17	49	166	53	176	53	95	219	192	67	22	157	...
5	129	7	75	39	113	101	31	68	219	33	107	113	147	80	8	3	...
6	21	202	102	234	227	151	253	130	57	97	74	153	49	11	40	210	...
7	231	46	37	119	15	59	65	235	124	83	86	182	216	12	8	165	...
8	211	79	255	126	36	16	85	216	245	24	113	223	86	65	223	106	...
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	...

5.3 Decryption (Party B)

When Party B receives the encrypted image, C, he must divide it into blocks of 8 × 8 submatrices $C_i = (C_1, C_2, C_3, \ldots)$ as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	**217**	**65**	**117**	**153**	**27**	**63**	**11**	**221**	*72*	*53*	*143*	*122*	*60*	*83*	*1*	*17*	...
2	**115**	**139**	**79**	**74**	**134**	**249**	**48**	**57**	*72*	*113*	*197*	*127*	*50*	*13*	*191*	*1*	...
3	**103**	**158**	**31**	**28**	**146**	**230**	**98**	**101**	*85*	*86*	*127*	*228*	*34*	*41*	*252*	*148*	...
4	**224**	**62**	**201**	**61**	**17**	**49**	**166**	**53**	*176*	*53*	*95*	*219*	*192*	*67*	*22*	*157*	...
5	**129**	**7**	**75**	**39**	**113**	**101**	**31**	**68**	*219*	*33*	*107*	*113*	*147*	*80*	*8*	*3*	...
6	**21**	**202**	**102**	**234**	**227**	**151**	**253**	**130**	*57*	*97*	*74*	*153*	*49*	*11*	*40*	*210*	...
7	**231**	**46**	**37**	**119**	**15**	**59**	**65**	**235**	*124*	*83*	*86*	*182*	*216*	*12*	*8*	*165*	...
8	**211**	**79**	**255**	**126**	**36**	**16**	**85**	**216**	*245*	*24*	*113*	*223*	*86*	*65*	*223*	*106*	...
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	...

$$C_1 = \begin{bmatrix} 217 & 65 & 117 & 153 & 27 & 63 & 11 & 221 \\ 115 & 139 & 79 & 74 & 134 & 249 & 48 & 57 \\ 103 & 158 & 31 & 28 & 146 & 230 & 98 & 101 \\ 224 & 62 & 201 & 61 & 17 & 49 & 166 & 53 \\ 129 & 7 & 75 & 39 & 113 & 101 & 31 & 68 \\ 21 & 202 & 102 & 234 & 227 & 151 & 253 & 130 \\ 231 & 46 & 37 & 119 & 15 & 59 & 65 & 235 \\ 211 & 79 & 255 & 126 & 36 & 16 & 85 & 216 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 72 & 53 & 143 & 122 & 60 & 83 & 1 & 17 \\ 72 & 113 & 197 & 127 & 50 & 13 & 191 & 1 \\ 85 & 86 & 127 & 228 & 34 & 41 & 252 & 148 \\ 176 & 53 & 95 & 219 & 192 & 67 & 22 & 157 \\ 219 & 33 & 107 & 113 & 147 & 80 & 8 & 3 \\ 57 & 97 & 74 & 153 & 49 & 11 & 40 & 210 \\ 124 & 83 & 84 & 182 & 216 & 12 & 8 & 165 \\ 245 & 24 & 113 & 223 & 86 & 65 & 223 & 106 \end{bmatrix},$$

Then, Party B must multiply all submatrices, C_i, by K_m^{-1} with modulo 256 to obtain $(O_1, O_2, O_3 \ldots)$, which represent the values of the original pixel as follows:

$$O_1 = K_m \times C_1$$

$$= \begin{bmatrix} 148 & 149 & 182 & 153 & 109 & 107 & 74 & 103 \\ 149 & 199 & 134 & 131 & 107 & 58 & 122 & 125 \\ 235 & 112 & 149 & 55 & 21 & 144 & 108 & 201 \\ 241 & 230 & 173 & 76 & 15 & 26 & 83 & 181 \\ 149 & 149 & 182 & 153 & 108 & 107 & 74 & 103 \\ 149 & 200 & 134 & 131 & 107 & 57 & 122 & 125 \\ 235 & 112 & 150 & 55 & 21 & 144 & 107 & 201 \\ 241 & 230 & 173 & 77 & 15 & 26 & 83 & 180 \end{bmatrix}$$

$$\times \begin{bmatrix} 217 & 65 & 117 & 153 & 27 & 63 & 11 & 221 \\ 115 & 139 & 79 & 74 & 134 & 249 & 48 & 57 \\ 103 & 158 & 31 & 28 & 146 & 230 & 98 & 101 \\ 224 & 62 & 201 & 61 & 17 & 49 & 166 & 53 \\ 129 & 7 & 75 & 39 & 113 & 101 & 31 & 68 \\ 21 & 202 & 102 & 234 & 227 & 151 & 253 & 130 \\ 231 & 46 & 37 & 119 & 15 & 59 & 65 & 235 \\ 211 & 79 & 255 & 126 & 36 & 16 & 85 & 216 \end{bmatrix},$$

$$= \begin{bmatrix} 65 & 199 & 193 & 185 & 179 & 185 & 191 & 189 \\ 54 & 196 & 192 & 195 & 195 & 192 & 191 & 192 \\ 54 & 197 & 194 & 197 & 195 & 191 & 191 & 188 \\ 55 & 187 & 187 & 188 & 186 & 180 & 180 & 182 \\ 62 & 181 & 180 & 180 & 180 & 183 & 182 & 183 \\ 69 & 174 & 174 & 178 & 179 & 179 & 181 & 186 \\ 72 & 180 & 179 & 174 & 174 & 181 & 179 & 180 \\ 77 & 176 & 171 & 168 & 170 & 175 & 169 & 174 \end{bmatrix}.$$

Thus, Party B will receive the groups of pixels that make up the plain image that Party A submitted, as seen below before pixels' values are converted into bytes so that Party B can view the image.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	65	199	193	185	179	185	191	189	191	193	200	193	197	199	200	202	...
2	54	196	192	195	195	192	191	192	188	193	195	193	190	189	193	191	...
3	54	197	194	197	195	191	191	188	186	197	191	189	189	186	188	187	...
4	55	187	187	188	186	180	180	182	183	187	183	184	185	189	190	192	...
5	62	181	180	180	180	183	182	183	181	181	181	185	185	188	190	187	...
6	69	174	174	178	179	179	181	186	182	179	181	178	180	185	189	185	...
7	72	180	179	174	174	181	179	180	168	171	175	172	172	180	175	175	...
8	77	176	171	168	170	175	169	174	168	173	172	164	163	172	164	165	...
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	...

6 Experimental results and security analysis

To evaluate the performance of the encryption approach, various metrics (parameters) are employed to examine the gray scale image encryption effectiveness and compare the encrypted image with the original image.

6.1 Experimental platform

The implementation is performed on an i7 CPU 1.30 GHz HP laptop with 8 GB RAM. The technique mentioned earlier was implemented in MATLAB (R2018A).

6.2 Experimental results

The efficiency of the proposed approach was tested in these studies utilizing a variety of standard grayscale images (Baboon, Boat, Lena, Peppers, Clock, and House) with sizes of 256 × 256 and 512 × 512. Datasets from SIPI contain these images [32]. Figure 3 displays the original images of size 256 × 256, ciphered images, and decoded images. According to Figure 3, encrypted images are unexpected, with no apparent data from the plain images in the encrypted images.

6.3 Key space analysis

An exhaustive key search needs a 2^{κ} operation to successfully break a method, where κ is the size of the key in bits. In the proposed scheme, the significant keys utilized to decode the encrypted image are D_A, D_B, K, and K_m. The key size of D_A, D_B, K, and K_m is dependent on the value of p in equation (6), where D_A and D_B have a t key size, respectively, while K has a $16t$ key size and t being the size of the key of p in bits for per element in the matrices. Since K_m is an 8 × 8 matrix of the prime field modulo 256 with 8 bits assigned to each element, the key size is 512 bits. Consequently, the key space analysis of the suggested technique may be determined as follows:

$$2^{8t} \times 2^{8t} \times 2^{16t} \times 2^{512} = 2^{32t+512}$$

Attackers who select an 8-bit value for p will need to carry out $2^{32(8)+512} = 2^{768}$ operations to defeat the method, which is robust enough to withstand brute-force attacks. When compared to other systems, such as those in Table 1, the key size is significantly large. As a result, the suggested approach offers a robust defense against brute-force attacks.

6.4 The entropy

One of the statistical scalar features utilized to evaluate image encryption is entropy. It displays the patterns that appear the most frequently. It quantifies the degree of randomness based on the likelihood of the pixel values. The optimal entropy value for an encrypted image is eight,

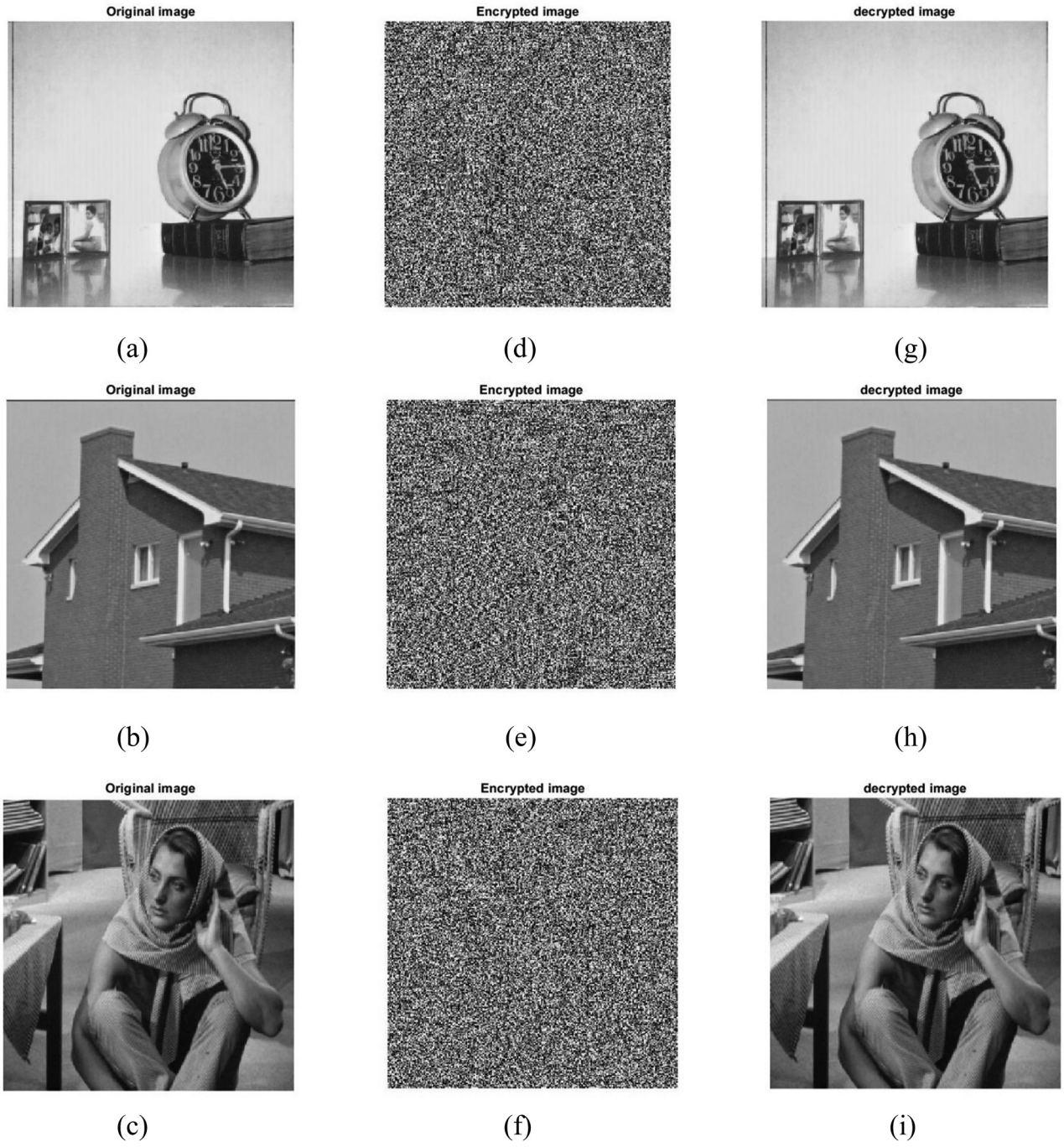

Figure 3: (a)–(c) Original images; (d)–(f) corresponding ciphered images; and (g)–(i) are deciphered images.

Table 1: Key space analysis

Scheme	Dawahdeh et al. [9]	Lone et al. [13]	Ismail and Misro [14]	Proposed algorithm
Key space size	2^{144}	2^{272}	2^{301}	2^{768}

Table 2: The Entropy of the proposed method compared with other methods

Images	Size	Proposed algorithm	Dawahdeh et al. [9]	Rajvir et al. [10]	Ye et al. [11]	Lone et al. [13]	Ismail and Misro [14]
Baboon	512 × 512	7.9994	—	—	—	—	7.9993
Boat	512 × 512	7.9993	—	—	—	—	7.9992
Lena	256 × 256	7.9974	7.9970	7.9969	—	—	—
Peppers	256 × 256	7.9970	—	7.9968	7.9976	7.9983	—
Baboon	256 × 256	7.9974	—	7.9971	—	7.9979	—
Barbara	256 × 256	7.9976	—	—	7.9974	7.9979	—
Cameraman	256 × 256	7.9969	7.9848	—	—	—	—
Clock	256 × 256	7.9958	—	—	—	—	7.9916
House	256 × 256	7.9984	—	—	7.9976	7.9982	7.9968

and if the entropy value is close to eight, the encrypted image efficiency improves. Generally speaking, the higher the entropy, the more difficult it is to crack the cryptographic system. Entropy is calculated using the following formula: size of

$$H(x) = \sum_{x=0}^{2^N-1} P(x) \times \log_2\left(\frac{1}{p(x)}\right), \tag{7}$$

where N is the number of bits in pixel value x. $p(x)$ denotes the probability of the pixel value x.

Table 2 shows the examined images' entropy, which is quite close to number eight, representing the optimum value. As a result, the proposed technique may withstand entropy attacks. When compared to previous strategies [9–11,13,14], our approach outperforms them in terms of information entropy. Hence, the suggested technique is immune to statistical attacks.

6.5 Histogram analysis

A histogram analysis is a graphical representation of the frequency distribution information between intensity values and pixel values of data. The equally distributed data of the cipher images is produced via a secure and good encryption algorithm. Figure 4 shows the outcomes of the test of grayscale images. The uniform distribution of the cipher image shows that the ciphered information is secure and that this method cannot reveal information to outsiders. Equation (8) can be used to establish data consistency using the chi-square test:

$$\chi^2 = \sum_{K=0}^{2^n-1} \frac{(\mu_K - \varepsilon_K)^2}{\varepsilon_K}, \tag{8}$$

where μ_K is the observed frequency and $\varepsilon_K = \frac{mn}{256}$ is the expected frequency of an image with size mn. At significance level a = 5% with 255 degrees of freedom, the value of chi-square passes the hypothesis uniformity such that $\chi^2_{(0.05,255)}$ = 293.2478. Table 3 shows that the chi-square values on a set of images are less than 293, which demonstrates that the histogram of the encrypted image is uniformly distributed.

6.6 Anti-noise attack analysis

When ciphered images are sent across physical communications channels, they are sensitive to noise or interference. Thus, the encryption method has to be sufficiently resilient to decrypt the encrypted images even while noise accumulates. The peak signal-to-noise ratio (PSNR) is used to determine how much the encrypted image and the plain image differ from one another. PSNR is calculated using the following formula:

$$\text{PSNR} = 20 \times \log_{10}\left(\frac{\max}{\text{MSE}}\right), \tag{9}$$

where max is the highest possible grayscale (8-bit) value. MSE can be defined as follows:

$$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - C_{ij})^2, \tag{10}$$

where A_{ij} is the plain image's pixel value and C_{ij} is the encrypted image's pixel value. The PSNR values for the 256 × 256 decrypted images are displayed in Table 4. The suggested method can withstand “salt and peppers” noise with an average PSNR of 37.461 db and a density of 0.0001. The average decreased to 26.46 and 16.94 dB when the noise intensity was increased to 0.001 and 0.01, respectively.

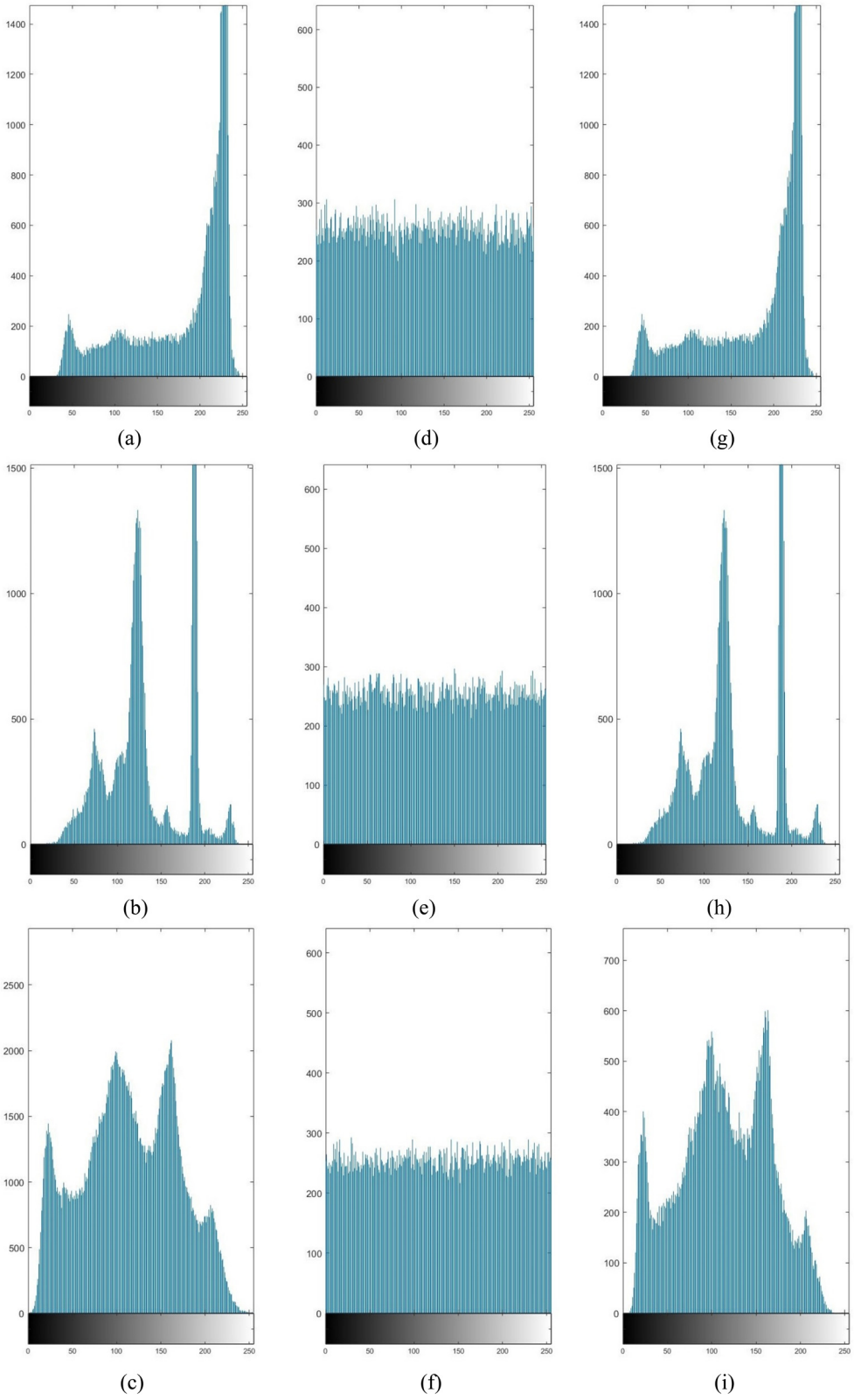

Figure 4: (a)–(c) Histograms of original images shown in Figure 3a–c; (d)–(f) histograms of the ciphered images shown in Figure 3d–f; and (g)–(i) histograms of the deciphered images shown in Figure 3c–i.

Table 3: Chi-square test values

Size	Barbara	Lena	Airplane	Peppers	House
256 × 256	220.8516	255.9844	268.5547	275.1172	277.6016

Table 4: The values of PSNR for noise attack

Images	Clock	House	Baboon
Salt and Pepper noise (0.0001)	37.5650	39.7241	35.0959
Salt and Pepper noise (0.001)	25.4511	26.7390	27.2014
Salt and Pepper noise (0.01)	15.5780	17.5441	17.7068

6.7 Differential attack analysis

To evaluate the ability to access the differential attack, the number of pixel change rate (NPCR) and unified average changing intensity (UACI) are utilized. UACI calculates the difference between the original and ciphered images. The highest UACI indicates that the suggested approach is immune to differential attacks. The following equations are used to calculate NPCR and UACI for a grayscale image:

$$\text{NPCR} = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} h(i,j) \times 100\%, \tag{11}$$

$$\text{UACI} = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%, \tag{12}$$

where $C_1(i,j)$ and $C_2(i,j)$ are the encrypted images created by two plain images with a one-pixel difference, and $h(i,j)$ is defined as follows:

$$h(i,j) = \begin{cases} 0, & \text{if} \quad C_2(i,j) = C_1(i,j) \\ 1, & \text{if} \quad C_2(i,j) \neq C_1(i,j). \end{cases} \tag{13}$$

The value of NPCR has to be greater than 99%, and the value of UACI of the image encryption methods has to be bigger than 33% to guarantee the method's security. Table 5 presents the average UACI and NPCR values of images examined for cryptographic method validation. The average NPCR and UACI values of the tested image are 99.62 and 33.45, respectively. Therefore, the cyphered images' average NPCR and UACI values beat the methods [9–11]. The outcomes show that NPCR and UACI average values are ideal, meaning that the suggested strategy will successfully withstand differential assaults.

6.8 Correlation analysis

The relationship between adjacent pixels is referred to as correlation. Thus, in plain images, a dense correlation graph is used to find a strong correlation among the pixels, whereas, in cipher images, an evenly distributed graph is used to find a low correlation among the adjacent pixels. Table 6 depicts the correlation coefficient value for cipher images in the horizontal (H), vertical (V), and diagonal (D) axes and compares results with the existing methods. The suggested technique outperforms methods presented in studies by Liu et al. [11] and Mohammed and Adamu [13], as reported in Table 6. The formula used to compute the correlation coefficients is as follows:

$$r_{x,y} = \frac{\text{cov}(x,y)}{\frac{1}{N}\sqrt{\sum_{i=1}^{N}[x_i - E(x)]^2 \times \sum_{i=1}^{N}[y_i - E(x)]^2}}, \tag{14}$$

where $\text{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)][y_i - E(y)]$ and $E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i$ and N is the total number of image pixels. As shown in Table 6, the correlation values of the different grayscale images are almost zero, which indicates that the suggested strategy successfully broke the strong relationship between nearby pixels in all tested images.

7 Conclusion

This work developed a novel image encryption scheme that combines the ECCHC established by Dawahdeh et al. [9]

Table 5: NPCR and UACI values for the chosen images of size 256 × 256

Images	Proposed algorithm		Dawahdeh et al. [9]		Rajvir et al. [10]		Lone et al. [13]	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
Lena	99.58	33.41	—	30.38	—	33.58	—	—
Peppers	99.65	33.43	—	—	—	34.64	99.63	33.31
Baboon	99.63	33.46	—	—	—	27.36	99.62	33.31
Barbara	99.64	33.51	—	—	—	—	99.63	33.34
House	99.61	33.43	—	—	—	—	99.62	33.35

Table 6: Comparison correlation results for the plain and cipher images with existing techniques

Method	Images	Size	Plain			Cipher		
			H	V	D	H	V	D
Proposed	Airplane	512 × 512	0.9663	0.9641	0.937	0.0019	−0.0041	0.0001
	Lena	512 × 512	0.9719	0.985	0.9593	−0.0007	0.0119	−0.001
	Peppers	512 × 512	0.9768	0.9792	0.9639	−0.0011	0.0051	0.0029
	Baboon	512 × 512	0.8665	0.7587	0.7262	−0.0029	0.0026	−0.001
	Boat	512 × 512	0.9381	0.9713	0.9222	0.0022	−0.0007	0.0014
	Airplane	256 × 256	0.9364	0.9302	0.8819	0.0003	0.0027	0.0047
	Lena	256 × 256	0.9456	0.9727	0.9213	0.0025	0.009	0.0014
	Peppers	256 × 256	0.9634	0.9704	0.9363	−0.0017	−0.0034	0.004
	Baboon	256 × 256	0.8736	0.8261	0.7843	−0.0013	0.0019	−0.0023
	Boat	256 × 256	0.9268	0.9452	0.8833	0.0023	−0.0042	−0.0059
	Clock	256 × 256	0.9565	0.9741	0.9389	0.0015	0.0042	−0.0015
Ye et al. [11]	Peppers	256 × 256	0.9719	0.9687	0.9488	0.004	−0.0044	−0.0012
	Barbara	256 × 256	0.9693	0.8971	0.8487	0.007	−0.0193	0.0031
	House	256 × 256	0.9664	0.978	0.9484	0.004	−0.0044	−0.0012
Lone et al. [13]	Peppers	256 × 256	0.8548	0.8791	0.9399	0.0004	0.0019	0.0003
	Baboon	256 × 256	0.8469	0.8456	0.8989	0.0021	0.0011	0.0011
	Barbara	256 × 256	0.9568	0.9214	0.8745	0.0017	−0.002	0.0047
	House	256 × 256	0.9654	0.9452	0.9624	−0.0019	0.0001	0.0029

with the ECDH key exchange protocol to improve the image cryptosystem's security. The investigation also shows that Dawahdeh's system, and others are insecure and can be broken by a brute-force attack since its key space is small. To overcome these drawbacks, a modified and enhanced version of the method is provided, which employs the modification of the ECDH key exchange protocol in conjunction with HC. We first utilized the ECDHHC to generate a secret shared key matrix K that consisted of elliptic curve points. We then employed K to produce the secret key matrix, K_m of size 8×8, which will be used for the encryption and decryption processes. In the encryption stage, we divided the input image into a set of 8×8 sub-matrices, and we utilized the matrix K_m to modify the pixels' values for all submatrices. According to the findings of the security analysis, the proposed method has robust encryption since encrypted images' histograms are uniformly distributed. Security and robustness testing of the suggested approach also indicated excellent sensitivity to every pixel and resilience in the face of all common assaults. We applied the technique to grayscale images in this paper, and the recommended technology will be investigated in future work to be used for color images.

Acknowledgments: The authors thank the University of Kufa for its support.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: The most datasets generated and/or analysed in this study are comprised in this submitted manuscript. The other datasets are available on reasonable request from the corresponding author with the attached information.

References

[1] Zou C, Wang X, Zhou C, Xu S, Huang C. A novel image encryption algorithm based on DNA strand exchange and diffusion. Appl Math Comput. 2022;430:127291. doi: 10.1016/j.amc.2022.127291.

[2] Zhang T, Zhu B, Ma Y, Zhou X. A novel image encryption algorithm based on multiple random DNA coding and annealing. Electronics. 2023;12(3):501. doi: 10.3390/electronics12030501.

[3] Cun Q, Tong X, Wang Z, Zhang M. A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing. Vis Comput. 2023;39(12):1–20. doi: 10.1007/s00371-022-02750-5.

[4] Kumar A, Raghava NS. An efficient image encryption scheme using elementary cellular automata with novel permutation box. Multimed Tools Appl. 2021;80:21727–50. doi: 10.1007/s11042-021-10750-1.

[5] Ma X, Wang C. Hyper-chaotic image encryption system based on N+2 ring Joseph algorithm and reversible cellular automata. Multimed Tools Appl. 2023;82(25):1–26. doi: 10.1007/s11042-023-15119-0.

[6] Neamah AA, Shukur AA. A novel conservative chaotic system involved in hyperbolic functions and its application to design an efficient colour image encryption scheme. Symmetry. 2023;15(8):1511. doi: 10.3390/sym15081511.

[7] Xu J, Zhao B, Wu Z. Research on color image encryption algorithm based on bit-plane and Chen Chaotic System. Entropy. 2022;24(2):186. doi: 10.3390/e24020186.

[8] Neamah AA. An image encryption scheme based on a seven-dimensional hyperchaotic system and Pascal's matrix. J King Saud Univ-Comput Inf Sci. 2023;35(3):238–48. doi: 10.1016/j.jksuci.2023.02.014.
[9] Dawahdeh ZE, Yaakob SN, bin Othman RR. A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher. J King Saud Univ-Comput Inf Sci. 2018;30(3):349–55. doi: 10.1016/j.jksuci.2017.06.004.
[10] Rajvir C, Satapathy S, Rajkumar S, Ramanathan L. Image encryption using modified elliptic curve cryptography and Hill cipher. In Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics. Vol. 1. Singapore: Springer; 2020. p. 675–83. doi: 10.1007/978-981-13-9282-5_64.
[11] Ye G, Liu M, Wu M. Double image encryption algorithm based on compressive sensing and elliptic curve. Alex Eng J. 2022;61(9):6785–95. doi: 10.1016/j.aej.2021.12.023.
[12] Parida P, Pradhan C, Alzubi JA, Javadpour A, Gheisari M, Liu Y, et al. Elliptic curve cryptographic image encryption using Henon map and Hopfield chaotic neural network. Multimed Tools Appl. 2023;82(22):1–26. doi: 10.1007/s11042-023-14607-7.
[13] Lone PN, Singh D, Stoffová V, Mishra DC, Mir UH, Kumar N. Cryptanalysis and improved image encryption scheme using elliptic curve and affine hill cipher. Mathematics. 2022;10(20):3878. doi: 10.3390/math10203878.
[14] Ismail NMH, Misro MY. An improved image encryption algorithm based on Bézier coefficients matrix. J King Saud Univ-Comput Inf Sci. 2022;34(10):10056–67. doi: 10.1016/j.jksuci.2022.10.005.
[15] Miller V. Uses of elliptic curves in cryptography. In Advances in Cryptology-CRYPTO. Vol. 85. Berlin/Heidelberg, Germany: Springer; 1985. p. 417–26.
[16] Koblitz N. Elliptic curve cryptosystems. Math Comput. 1987;48(177):203–9.
[17] Li L, Abd El-Latif AA, Niu X. Elliptic curve ElGamal based homomorphic image encryption scheme for sharing secret images. Signal Process. 2012;92(4):1069–78. doi: 10.1016/j.sigpro.2011.10.020.
[18] Zhang X, Wang X. Digital image encryption algorithm based on elliptic curve public cryptosystem. IEEE Access. 2018;6:70025–34. doi: 10.1109/ACCESS.2018.2879844.
[19] Hayat U, Azam NA. A novel image encryption scheme based on an elliptic curve. Signal Process. 2019;155:391–402. doi: 10.1016/j.sigpro.2018.10.011.
[20] Díaz EAH, Meana HMP, García VMS. Encryption of RGB images by means of a novel cryptosystem using elliptic curves and chaos,. IEEE Lat Am Trans. 2020;18(8):1407–15. doi: 10.1109/TLA.2020.9111676.
[21] Obaid ZK, AlSaffar NFH. Image encryption based on elliptic curve cryptosystem. Int J Electr Comput Eng. 2021;11(2):1293. doi: 10.11591/ijece.v11i2.pp1293-1302.
[22] Liang H, Zhang G, Hou W, Huang P, Liu B, Lim S. A novel asymmetric hyperchaotic image encryption scheme based on elliptic curve cryptography. Appl Sci. 2021;11(12):5691. doi: 10.3390/app11125691.
[23] Abbas AM, Alharbi AA, Ibrahim S. A novel parallelizable chaotic image encryption scheme based on elliptic curves,. IEEE Access. 2021;9:54978–91. doi: 10.1109/ACCESS.2021.3068931.
[24] Hayat U, Ullah I, Azam NA, Azhar S. A novel image encryption scheme based on elliptic curves over finite rings. Entropy. 2022;24(5):571. doi: 10.3390/e24050571.
[25] Castro F, Impedovo D, Pirlo G. A medical image encryption scheme for secure fingerprint-based authenticated transmission. Appl Sci. 2023;13(10):6099. doi: 10.3390/app13106099.
[26] Hill LS. Cryptography in an algebraic alphabet. Am Math Monthly. 1929;36(6):306–12. doi: 10.1080/00029890.1929.11986963.
[27] Neamah AA. New collisions to improve pollard's rho method of solving the discrete logarithm problem on elliptic curves. J Comput Sci. 2015;11(9):971–5. doi: 10.3844/jcssp.2015.971.975.
[28] Hankerson D, Menezes AJ, Vanstone SA. Guide to Elliptic Curve Cryptography. Springer Professional Computing. New York, USA: Springer; 2004.
[29] Diffie W, Hellman ME. Multiuser cryptographic techniques. In Proceedings of the June 7-10, 1976, National Computer Conference and Exposition; 1976. p. 109–12. doi: 10.1145/1499799.1499815.
[30] Hadi HH, Neamah AA. Diffie-hellman key exchange based on block matrices combined with elliptic curves. Int J Intell Syst Appl Eng. 2023;11(5s):353–60.
[31] Panigrahy SK, Acharya B, Jena D. Image encryption using self-invertible key matrix of hill cipher algorithm. Int J Secur. 2007;1(1):14–27.
[32] USC-SIPI Image Database. http://sipi.usc.edu/database/database.php (accessed on 5 June 2023; 2023.