International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 2, April2021, pp. 1293~1302
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i2.pp1293-1302

Image encryption based on elliptic curve cryptosystem

Zahraa Kadhim Obaidand, Najlae Falah Hameed Al Saffar
Department of Mathematics, Faculty of Computer Science and Mathematics, University of Kufa, Iraq

Article Info

Article history:

Received Feb 16, 2020
Revised Aug 5, 2020
Accepted Sep 23, 2020

Keywords:

Elliptic curve cryptosystem
Hilbert matrix
Image encryption
Self invertible matrix

ABSTRACT

Image encryption based on elliptic curve cryptosystem and reducing its complexity is still being actively researched. Generating matrix for encryption algorithm secret key together with Hilbert matrix will be involved in this study. For a first case we will need not to compute the inverse matrix for the decryption processing cause the matrix that be generated in encryption step was self invertible matrix. While for the second case, computing the inverse matrix will be required. Peak signal to noise ratio (PSNR), and unified average changing intensity (UACI) will be used to assess which case is more efficiency to encryption the grayscale image.

Corresponding Author:

Najlae Falah Hameed Al Saffar
Department of Mathematics
Faculty of Computer Science and Mathematics
University of Kufa, Iraq
Email: najlaa.hameed@uokufa.edu.iq

1. INTRODUCTION

Cryptography is a strategy for putting away and transmitting information in a verified structure with the goal that lone proposed client can peruse and process it [1]. It includes encryption and decoding of messages.Encryption is the way toward changing over plain information into cipher text and decoding is the way toward getting back the first message from the encoded content. There are many uses for Cryptography such like: protect e-mail information, credit card information. Indeed it provides confidentiality, authentication, Integrity and non-repudiation [2].

Elliptic curves are arithmeticcurves which have been studied by numerous mathematicians for quite a while [3]. One of its most important applications appeared in 1985, where Neal [4] and Miller [5] independently proposed the public key cryptosystems based on elliptic curve they named elliptic curve cryptosystem (ECC). From that point forward, numerous scientists have gone through years concentrating the quality of ECC and improving procedures for its execution. In ECC a 160-bit key provides the same security as compared to the traditional cryptosystem RSA [6] with a 1024-bit key, in this way it can reduced computational cost or processing cost [7]. The security of ECC depends on the difficulty of solving elliptic curve discrete logarithm problem (ECDLP) [8].

Digital images are an appealing information type that offers a far reaching scope of utilization. Any clients are keen on actualizing content security strategies to their pictures [9]. Recently, many image encryption techniques have been proposed to verify interactive media data before transmission over insecure channels such like [10, 11]. Elliptic curve cryptosystem is a superior strategy to transmit the picture safely [12].

The matrix that appears to have an endless interest for mathematicians, is symmetric and definite, it is The Hilbert matrix, with $(\mathrm{i,j})$ element $\frac{1}{(\mathrm{i+j-1})}$. It is too special matrix that is because symmetric positive definite and totally positive [13]. For the purposes of the work in this study, we will use this matrix for a specific modulo, so no denominator will appear in any element of this matrix.

The security of images is exceptional compelling in this paper. Indeed new image encryption techniques have been proposed in this paper to combine ECC with Hilbert matrix. The new approaches use ECC to generate the private and public keys, and then both sender and receiver have the ability to produce the secret key with no need to share it through the internet or unsecured communication channel. One of the most important things that these technologies will focus on is that the matrices must be invertible. So, if the key matrix is not invertible, the decryption process cannot be done, and the receiver cannot get the original data.

Many researchrs were working on this subject such like Singh [14], in 2015 used image encryption using elliptic curve cryptography. They implement the elliptic curve cryptography to encrypt, decrypt and digitally sign the cipher image to provide authenticity and integrity. Ahmed [15], in 2013 used a hybrid chaotic system and cyclic elliptic curve for image encryption. The new scheme generates an initial key stream based on chaotic system and an external secret key of 256-bit in a feedback manner. Then, the generated key stream are mixed with key sequences derived from the cyclic elliptic curve points. Thorough encryption performance and security analysis ascertains efficacy of the proposed encryption scheme. Nagaraj [16] they propose in 2015 a new encryption technique using elliptic curve cryptography with a magic matrix operations for securing images that transmits over a public unsecured channel. There are two most important groups of image encryption algorithms: some are non chaos-based selective methods and chaos- based selective methods. The majority of these algorithms is planned for a specific image format, compressed or uncompressed.

2. ELLIPTIC CURVE FUNCTION

Elliptic curve cryptosystem (ECC) it is a reasonable encryption strategy to be utilized in for example: embedded systems and mobile devices, that is because it can provide high security with smaller key size and fewer computations with less memory usage and lower power consumptions [17].

- Definition: An elliptic curve E over a prime field F_p is defined by $E_p(a,b)\colon y^2 = x^3 + ax + b\, modp$, where $p > 3$, $a,b \in F_p$ and satisfy the condition $4a^3 + 27b^2 modp \not\equiv 0$. The elliptic curve group $E(E_p)$ consists of all points (x,y) that satisfy the elliptic curve $E_p(a,b)$ and the point at the infinity O_∞ [18, 19].
- Elliptic curve operations: The primary operations related to elliptic curve function is the elliptic curve scalar multiplication which is the main operation on the elliptic curve that consumes more time in encryption and decryption operations. Two operations are involved in calculating the elliptic curve scalar multiplication, they are point addition and point doubling [20].
- Point addition: Suppose $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, where $P_1 \neq P_2$, are two points lie on an elliptic curve $E_p(a,b)$. Adding the two points P_1 and P_2giving a third point $P_3 = (x_3, y_3)$, as $x_3 \equiv (s^2 - x_1 - x_2)\, modp$, $y_3 \equiv (s(x_1 - x_3) - y_1))\, modp$ and $s = \frac{y_2 - y_1}{x_2 - x_1}\, modp$. P_3 should lie on the same curve $E_p(a,b)$.
- Point doubling: Suppose $P = (x_1, y_1)$is a point on an elliptic curve $E_p(a,b)$, the point $R = 2P = (x_2, y_2)$ that results from doubling the point P as $x_2 \equiv (s^2 - 2x_1)\, modp$, $y_2 \equiv (s(x_1 - x_2) - y_1))\, modp$ and $s = \frac{3x_1^2 + a}{2y_1}\, modp$. $R$is also point on an elliptic curve $E_p(a,b)$.
- Elliptic curve scalar multiplication: Let Pbe any point on the elliptic curve $E_p(a,b)$. Elliptic curve scalar multiplication operation over $P$is defined by the repeated addition

$$kP = \underbrace{P + P + \cdots +}_{k\text{times}} P.$$

3. HILBERT MATRIX

Hilbert [21] in 1894 presented a square matrix with entries being the unit fractions. That is mean each element of this matrix say h_{ij} will be written as $\frac{1}{i+j-1}$. David Hilbert named this type of matrices as Hilbert Matrix. For the purposes of the work in this study, we will use $n \times n$-Hilbert matrix with for modulo $2n - 1$, so no denominator will appear in any element of this matrix. So by computing the inverse of each denominator modulo $(i + j - 1)$ to create a new Hilbert matrix. For example, $4 \times 4 -$ Hilbert matrix:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \text{can be exchange to } 4 \times 4\text{-Hilbert matrix modulo 7, as } \begin{bmatrix} 1 & 4 & 5 & 2 \\ 4 & 5 & 2 & 3 \\ 5 & 2 & 3 & 6 \\ 2 & 3 & 6 & 7 \end{bmatrix}.$$

Involved Hilbert matrix with cryptography is new subject duscude recently by Roopaei [22]. In this work, we will try to use this matrix to high level of security to encrypte images. Indeed, there are another types of invertible matrix [23, 24] can be subject to future studies.

4. MATERIALS AND METHODS

4.1. Proposed algorithms

Hybrid the ECC and Hilbert matrix is approach to encrypt the image is a technique has been introduced in this section in two cases. These techniques increase the security and make the system more efficient, also it speeds up the decryption computations since it does not need the computation of the key matrix inverse for the first case.

Suppose the sender (User A) wants to send an image M to the other party (User B) using this technique over an insecure channel. Firstly, they should agree on the elliptic curve $E_p(a, b)$ and share the domain parameters $\{a, b, p, G\}$, where G is the generator point. Then each party needs to choose randomly his private key from the interval $[1, p - 1]$; n_A for User A and n_B for User B, and generates his public key as P_A=$n_A \cdot G$ and P_B=$n_B \cdot G$. Each user multiplies his private key by the public key of the other user to get the initial key as:

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

then computes;

$$K_1 = x.G = (k_{11}, k_{12})$$

$$K_2 = y.G = (k_{21}, k_{22})$$

The next step is generating the secret key matrix K_m by sender and receiver. The inverse of the key matrix does not always exist. So, if the key matrix is not invertible, the recipient cannot decrypt the encrypted data. To solve this problem, the self invertible key matrix [25] will be generated, and the same key will be used for encryption and decryption and no need to find the inverse key matrix in the first case. But the second case, we need compute inverse for the key matrix.

The first case and the second case will be implemented on grayscale images. The image will be divided into blocks of size four pixel values. So, each party produces the 4×4 key matrix K_m, where K_m be a self invertible matrix partitioned as four square matrices: K_{11}, K_{12}, K_{21} and K_{22}. So, we can rewrite K_m as $\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$. Now, if we consider $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$ we can calculate the others square matrices as: $K_{12} = I - K_{11}$, $K_{21} = I + K_{11}$, $K_{11} + K_{22} = 0$, where I is the identity matrix.

4.2. First case

In this case generating K_m will be as follows: Suppose that $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} + H_{2\times2}$, where H is Hilbert matrix with dimension 2 × 2. So, if we consider $K_m = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$ be a self invertible matrix, then other partitions of the secret matrix key $K_m$is obtained by solving $K_{12} = I - K_{11}$, $K_{21} = I + K_{11}$, $K_{11} + K_{22} = 0$, where I is the identity matrix. Now, separate the image pixel values into blocks of size four, each block will be converted to a vector of size 4×1: $P_1, P_2, P_3, \ldots$. Next step will be calculating the ciphered vectors $C_1, C_2, C_3, \ldots$ as: $C_i = K_m.P_i \bmod 256,\ i = 1, 2, 3, \ldots$. The last step for the encryption algorithm is reconstruct the ciphered image from the values of C_i and send it to the other party B. For the decryption processes, party B will separate the ciphered image pixel values into blocks of 4×1: $C_1, C_2, C_3, \ldots$. The next step is computing $P_1, P_2, P_3, \ldots$ as: $P_i = K_m.C_i \bmod 256,\ i = 1, 2, 3, \ldots$. The last step for the decryption algorithm is reconstruct the plain image from the values of P_i.

4.3. Second case

In this case generating $\mathrm{K}_{\tilde{\mathrm{m}}}$ will be as follows: $\mathrm{K}_{\tilde{\mathrm{m}}} = \mathrm{K}_\mathrm{m} + \mathrm{H}_{4\times4}$,, where $\mathrm{K}_\mathrm{m} = \begin{bmatrix} \mathrm{K}_{11} & \mathrm{K}_{12} \\ \mathrm{K}_{21} & \mathrm{K}_{22} \end{bmatrix}$ and $\mathrm{H}_{4\times4}$, where H is Hilbert matrix with dimension 4×4. Now, separate the image pixel values into blocks of size four, each block will be converted to a vector of size 4×1: $\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots.$ Next step will be calculating the ciphered vectors $\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \ldots$ as: $\mathrm{C}_\mathrm{i} = \mathrm{K}_\mathrm{m}.\mathrm{P}_\mathrm{i} \bmod 256,\ \mathrm{i} = 1, 2, 3, \ldots.$ The last step for the encryption algorithm is reconstruct the ciphered image from the values of C_i and send it to the other party B. For the decryption processes, party B will separate the ciphered image pixel values into blocks of 4×1: $\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \ldots.$ The next step is computing $\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots$ as: $\mathrm{P}_\mathrm{i} = \mathrm{K}_{\tilde{\mathrm{M}}}^{-1}.\mathrm{C}_\mathrm{i} \bmod 256,\ \mathrm{i} = 1, 2, 3, \ldots.$ The last step for the decryption algorithm is reconstruct the plain image from the values of P_i.

5. IMPLEMENTATION OF THE PROPOSED ALGORITHMS MATERIALS AND METHODS

Assume that party A wants to send an image "CAT IMAGE" to party B using the proposed algorithm. They will agreed to use an elliptic curve say $\mathrm{E}_{37}(1, 3)\colon \mathrm{y}^2 = \mathrm{x}^3 + \mathrm{x} + 3 \bmod 37$, where $\mathrm{a}^3 + 27\mathrm{b}^2 \bmod \mathrm{p} = 247 \bmod 37 = 25 \neq 0$. The points that satisfying $\mathrm{E}_{37}(1, 3)$ are: (0,15), (0,22), (3,12), (3,25), (4,16), (4,21), (6,15), (6,22), (9,1), (9,36), (12,2), (12,35), (13,17), (13,20), (15,10), (15,27), (17,7), (17,30), (18,9), (18,28), (19,6), (19,31), (26,17), (26,20), (29,1), (29,36), (31,15), (31,22), (32,13), (32,24), (33,3), (33,34), (34,11), (34,26), (35,17), (35,20), (36,1) and (36,36). So $\#\,\mathrm{E}_{37}(1, 3) = 39$. So, if we choose $\mathrm{G} = (0, 15)$, the domain parameters for $\mathrm{E}_{37}(1, 3)$ are $\{\mathrm{a}, \mathrm{b}, \mathrm{P}, \mathrm{G}\} = \{1, 3, 37, (0, 15)\}$.

Figures 1 and 2 show the original image, ciphered image, and deciphered image for the proposed algorithm first case and second case respectively. MATLAB R2014a, 64-bit software on Core i5 computer with CPU @1.80 GHz 2.30 GHz and RAM 6 GB is used for encryption and decryption processes.

Now, to apply the proposed algorithm first case we will do the following steps:

Step 1: “Generating of keys”

- Party A Chooses the private key $n_A = 11 \in [1,36]$
- He computes the public key $P_A = n_A \cdot G = 11(0,15) = (3,25)$
- He computes the$K = n_A.P_B = 11(26,20) = (26,17) = (x, y)$
- He computes $K_1 = x.G = 26(0,15) = (26,17) = (k_{11}, k_{12})$

and $K_2 = y.G = 17(0,15) = (19,31) = (k_{21}, k_{22})$

- He constructs $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} + H_{2\times2} = \begin{bmatrix} 26 & 17 \\ 19 & 31 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 27 & 19 \\ 21 & 34 \end{bmatrix}$
- He calculates $K_{12} = (I - K_{11}) mod 256 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 27 & 19 \\ 21 & 34 \end{bmatrix}\right) mod 256 = \begin{bmatrix} 230 & 237 \\ 235 & 223 \end{bmatrix}$
- He calculates $K_{21} = (I + K_{11}) mod 256 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 27 & 19 \\ 21 & 34 \end{bmatrix}\right) mod 256 = \begin{bmatrix} 28 & 19 \\ 21 & 35 \end{bmatrix}$
- He solves $K_{11} + K_{22} = 0$, so $K_{22} = (-K_{11}) mod\ 256 = \begin{bmatrix} 229 & 237 \\ 235 & 222 \end{bmatrix}$
- Finally, he constructs the self invertible key matrix

$$K_m = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix}$$

In the other side:

- Party BChooses the private key $n_B = 13 \in [1,36]$
- He computes the public key $P_B = n_B.G = 13(0,15) = (26,20)$
- Hecomputes$K = n_B.P_A = 13(3,25) = (26,17) = (x, y)$
- He computes $K_1 = x.G = 26(0,15) = (26,17) = (k_{11}, k_{12})$ and $K_2 = y.G = 17(0,15) = (19,31) = (k_{21}, k_{22})$

He will do the other step same like party A to construct the self invertible key matrix

$$K_m = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix}$$

Step 2: “Encryption by party A”

- He separates the pixel values of “CAT IMAGE” into blocks of size four as:

	1	2	3	4	5	6	7	8	9
1	249	246	244	245	247	244	237	231	…
2	249	246	244	245	246	244	237	230	…
3	249	246	243	244	246	244	237	230	…
4	248	245	243	244	246	243	236	230	…
5	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

$$\text{So, } P_1 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix}, P_2 = \begin{bmatrix} 247 \\ 244 \\ 237 \\ 231 \end{bmatrix}, P_3 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix}, \dots$$

- He computes the values of $C_1, C_2, C_3, \ldots$ as:

$$C_1 = K_m \cdot P_1 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix} mod\ 256 = \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix},$$

$$C_2 = K_m \cdot P_2 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 247 \\ 244 \\ 237 \\ 231 \end{bmatrix} mod\ 256 = \begin{bmatrix} 242 \\ 115 \\ 252 \\ 128 \end{bmatrix},$$

$$C_3 = K_m \cdot P_3 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix} mod\ 256 = \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix}, \dots$$

- He reconstructs the ciphered image from the values of C_i as:

	1	2	3	4	5	6	7	8	9
1	142	128	147	129	242	115	252	128	…
2	142	128	147	129	234	127	243	141	…
3	187	182	193	184	234	127	243	141	…
4	141	127	146	128	241	114	251	127	…
5	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

- He sends it to the other party B.

Step 3: “Decryption by party B”

- He separates the pixel values of ciphered image into blocks of size four as:

$$C_1 = \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix}, C_2 = \begin{bmatrix} 242 \\ 115 \\ 252 \\ 128 \end{bmatrix}, C_3 = \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix}, \dots$$

- He computes the values of $P_1, P_2, P_3, \ldots$ as:

$$P_1 = K_m \cdot C_1 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix} mod256 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix},$$

$$P_2 = K_m \cdot C_2 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 242 \\ 115 \\ 252 \\ 128 \end{bmatrix} mod256 = \begin{bmatrix} 247 \\ 244 \\ 237 \\ 231 \end{bmatrix},$$

$$P_3 = K_m \cdot C_3 = \begin{bmatrix} 27 & 19 & 230 & 237 \\ 21 & 34 & 235 & 223 \\ 28 & 19 & 229 & 237 \\ 21 & 35 & 235 & 222 \end{bmatrix} \begin{bmatrix} 142 \\ 128 \\ 147 \\ 129 \end{bmatrix} mod256 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix}, \dots$$

- He reconstructs the plain image from the values of P_i as:

	1	2	3	4	5	6	7	8	9
1	249	246	244	245	247	244	237	231	…
2	249	246	244	245	246	244	237	230	…
3	249	246	243	244	246	244	237	230	…
4	248	245	243	244	246	243	236	230	…
5	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

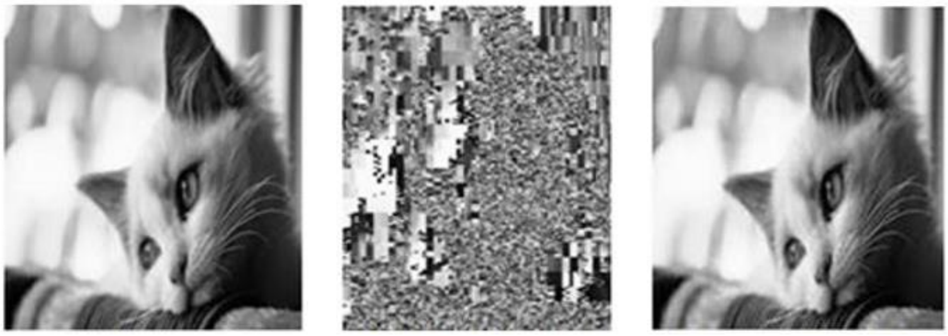
(a) (b) (c)

Figure 1. (a) Plain image, (b) ciphered image, and (c) deciphered imagefor cat image/the proposed algorithm first case

The second case for the proposed algorithm will apply by doing the following steps:
Step 1: “Generating of keys”

- Party A Chooses the private key $n_A = 11 \in [1,36]$
- He computes the public key $P_A = n_A \cdot G = 11(0,15) = (3,25)$
- He computes the$K = n_A.P_B = 11(26,20) \ = (26,17) = (x,y)$
- He computes $K_1 = x.G = 26(0,15) = (26,17) = (k_{11},k_{12})$

and $K_2 = y.G = 17(0,15) = (19,31) = (k_{21},k_{22})$

- He constructs $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} = \begin{bmatrix} 26 & 17 \\ 19 & 31 \end{bmatrix}$
- He calculates $K_{12} = (I - K_{11})mod256 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 26 & 17 \\ 19 & 31 \end{bmatrix}\right) mod256 = \begin{bmatrix} 231 & 239 \\ 237 & 226 \end{bmatrix}$
- He calculates $K_{21} = (I + K_{11})mod256 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 26 & 17 \\ 19 & 31 \end{bmatrix}\right) mod256 = \begin{bmatrix} 27 & 17 \\ 19 & 32 \end{bmatrix}$
- He solves $K_{11} + K_{22} = 0$, so $K_{22} = (-K_{11})mod\ 256 = \begin{bmatrix} 230 & 239 \\ 237 & 225 \end{bmatrix}$
- He constructs the self invertible key matrix $K_m = \begin{bmatrix} 26 & 17 & 231 & 239 \\ 19 & 31 & 237 & 226 \\ 27 & 17 & 230 & 239 \\ 19 & 32 & 237 & 225 \end{bmatrix}$
- Finally, he calculates

$$K_{\tilde{m}} = K_m + H_{4\times4} = \left(\begin{bmatrix} 26 & 17 & 231 & 239 \\ 19 & 31 & 237 & 226 \\ 27 & 17 & 230 & 239 \\ 19 & 32 & 237 & 225 \end{bmatrix} + \begin{bmatrix} 1 & 4 & 5 & 2 \\ 4 & 5 & 2 & 3 \\ 5 & 2 & 3 & 6 \\ 2 & 3 & 6 & 7 \end{bmatrix}\right) mod256$$
$$= \begin{bmatrix} 27 & 21 & 236 & 241 \\ 23 & 36 & 239 & 229 \\ 32 & 19 & 233 & 245 \\ 21 & 35 & 243 & 232 \end{bmatrix}.$$

In the other side:

- Party BChooses the private key $n_B = 13 \in [1,36]$
- He computes the public key $P_B = n_B.G = 13(0,15) = (26,20)$
- Hecomputes$K = n_B.P_A = 13(3,25) = (26,17) = (x,y)$
- He computes $K_1 = x.G = 26(0,15) = (26,17) = (k_{11},k_{12})$

and $K_2 = y.G = 17(0,15) = (19,31) = (k_{21},k_{22})$

- He will do the other step same like party A to construct the matrix

$$K_{\breve{m}} = \begin{bmatrix} 27 & 21 & 236 & 241 \\ 23 & 36 & 239 & 229 \\ 32 & 19 & 233 & 245 \\ 21 & 35 & 243 & 232 \end{bmatrix}$$

- Finally, he computes

$$K_{\breve{m}}^{-1} = \begin{bmatrix} 97 & 235 & 48 & 69 \\ 57 & 118 & 141 & 119 \\ 136 & 53 & 43 & 177 \\ 73 & 125 & 107 & 106 \end{bmatrix}.$$

Step 2: “Encryption by party A”

- He separates the pixel values of “CAT IMAGE” into blocks of size four as he did in the proposed algorithm first case:
- He computes the values of $C_1, C_2, C_3, \ldots$ as:

$$C_1 = K_{\breve{m}} \cdot P_1 = \begin{bmatrix} 27 & 21 & 236 & 241 \\ 23 & 36 & 239 & 229 \\ 32 & 19 & 233 & 245 \\ 21 & 35 & 243 & 232 \end{bmatrix} \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix} mod\ 256 = \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix},$$

$$C_2 = K_{\breve{m}} \cdot P_2 = \begin{bmatrix} 27 & 21 & 236 & 241 \\ 23 & 36 & 239 & 229 \\ 32 & 19 & 233 & 245 \\ 21 & 35 & 243 & 232 \end{bmatrix} \begin{bmatrix} 247 \\ 244 \\ 237 \\ 231 \end{bmatrix} mod\ 256 = \begin{bmatrix} 4 \\ 103 \\ 196 \\ 238 \end{bmatrix},$$

$$C_3 = K_{\breve{m}} \cdot P_3 = \begin{bmatrix} 27 & 21 & 236 & 241 \\ 23 & 36 & 239 & 229 \\ 32 & 19 & 233 & 245 \\ 21 & 35 & 243 & 232 \end{bmatrix} \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix} mod\ 256 = \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix}, \ldots$$

- He reconstructs the ciphered image from the values of C_i as:

	1	2	3	4	5	6	7	8	9
1	6	236	239	179	4	103	196	238	…
2	6	236	239	179	248	107	175	241	…
3	41	24	17	216	248	107	175	241	…
4	249	221	222	160	247	88	179	219	…
5	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

- He sends it to the other party B.

Step 3: “Decryption by party B”

- He separates the pixel values of ciphered image into blocks of size four as:

$$C_1 = \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix}, C_2 = \begin{bmatrix} 4 \\ 103 \\ 196 \\ 238 \end{bmatrix}, C_3 = \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix}, \ldots$$

- He computes the values of $P_1, P_2, P_3, \ldots$ as:

$$K_{\breve{m}}^{-1} = \begin{bmatrix} 97 & 235 & 48 & 69 \\ 57 & 118 & 141 & 119 \\ 136 & 53 & 43 & 177 \\ 73 & 125 & 107 & 106 \end{bmatrix}$$

$$P_1 = K_{\breve{m}}^{-1} \cdot C_1 = \begin{bmatrix} 97 & 235 & 48 & 69 \\ 57 & 118 & 141 & 119 \\ 136 & 53 & 43 & 177 \\ 73 & 125 & 107 & 106 \end{bmatrix} \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix} mod256 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix},$$

$$P_2 = K_{\breve{m}}^{-1} \cdot C_2 = \begin{bmatrix} 97 & 235 & 48 & 69 \\ 57 & 118 & 141 & 119 \\ 136 & 53 & 43 & 177 \\ 73 & 125 & 107 & 106 \end{bmatrix} \begin{bmatrix} 4 \\ 103 \\ 196 \\ 238 \end{bmatrix} mod256 = \begin{bmatrix} 247 \\ 244 \\ 237 \\ 231 \end{bmatrix},$$

$$P_3 = K_{\breve{m}}^{-1} \cdot C_3 = \begin{bmatrix} 97 & 235 & 48 & 69 \\ 57 & 118 & 141 & 119 \\ 136 & 53 & 43 & 177 \\ 73 & 125 & 107 & 106 \end{bmatrix} \begin{bmatrix} 6 \\ 236 \\ 239 \\ 179 \end{bmatrix} mod256 = \begin{bmatrix} 249 \\ 246 \\ 244 \\ 245 \end{bmatrix}, \ldots$$

– He reconstructs the plain image from the values of P_i as:

	1	2	3	4	5	6	7	8	9
1	249	246	244	245	247	244	237	231	…
2	249	246	244	245	246	244	237	230	…
3	249	246	243	244	246	244	237	230	…
4	248	245	243	244	246	243	236	230	…
5	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

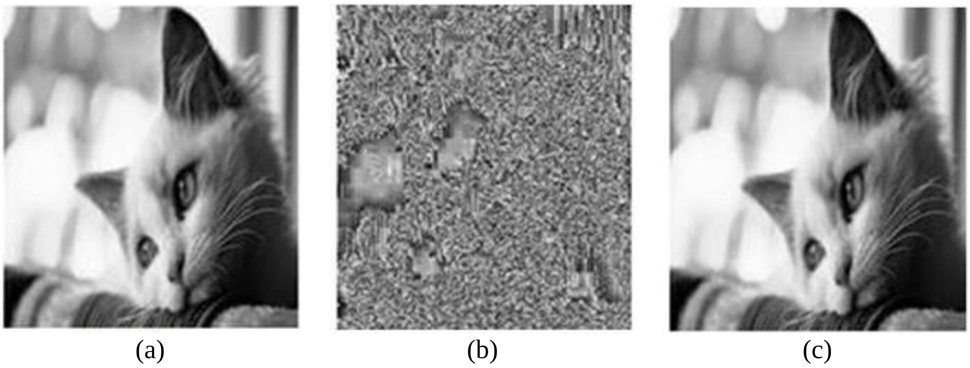

Figure 2. (a) Plain image, (b) ciphered image, and (c) deciphered image for cat image/the proposed algorithm second case

6. SECURITY ANALYSIS

A security analysis of the cryptographic algorithms is a basic procedure to guarantee the quality of cryptographic calculation [26]. So, to demonstrate the strength of the proposed algorithms we will discuss two analyzes PSNR and UACI.

6.1. Peak signal to noise ratio (PSNR)

It is an accessibility estimation of whether a critical unique image information is installed in the ciphered image. Actually, it can be defined by the computing the mean squared error (MSE) [27]. In this work, PSNRhas been calculated for the ciphered image and plain image that are shown in Figures 1 and 2 were 7.8654 and 7.6568 respectively, which means that the ciphered image is not like the plain image, so it is so hard for an aggressor to recover the plain image. The equation of the PSNR is as $\mathrm{PSNR} = 10\log_{10}\frac{255\cdot 255}{\mathrm{MSE}}$ where $\mathrm{MSE} = \frac{1}{\mathrm{M*N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}\sum_{\mathrm{j=1}}^{\mathrm{M}}(\mathrm{X(i,j)} - \mathrm{Y(i,j)})^2$, $\mathrm{X(i,j)}$ and$\mathrm{Y(i,j)}$ are the pixel value of plain image and ciphered image respectively.

6.2. Unified average changing intensity (UACI)

It is one of differential analyses used to evaluate the strength of image encryption, where it is estimated the contrast between the ciphered image and plain image. The highest value of the UACI (approximately 33.46%) implies that the proposed procedure is safe against differential assaults [28]. In this work we get 31.6912 and 34.0998 for the proposed algorithm first and second case respectively, it is so hard for an attacker to recover the plain image. It can be calculated by the $\mathrm{UACI} = \frac{1}{256*256}\sum_{\mathrm{i=1}}^{256}\sum_{\mathrm{j=1}}^{256}\frac{\mathrm{X(i,j)-Y(i,j)}}{255}\cdot 100\%$, $\mathrm{X(i,j)}$ and $\mathrm{Y(i,j)}$ are the pixel value of plain image and ciphered image respectively. The first case and second case of the proposed algorithms are tested on for cat image, lena image, baboon image and caramen image and the results are summarized in Table 1.

Table 1. PSNR, and UACI

The Proposed Algorithm	Cat Image		Lena Image		Baboon Image		Caramen Image	
	PSNR	UACI	PSNR	UACI	PSNR	UACI	PSNR	UACI
First Case	7.8654	31.6912	9.2996	28.1854	9.8015	27.0906	9.0097	26.9897
Second Case	7.6568	34.0998	9.2549	28.529	9.7765	27.188	8.0926	32.092

7. CONCLUSION

ECC provides equivalent security with less key size, low mathematical rather than the Global RSA cryptosystem. Algorithms for image encryption based on elliptic curve cryptosystem in two cases are proposed in this work. Hilbert matrix is involved in the first and the second case for the proposed algorithms with dimension 2×2 and 4×4 respectively. Table 1 shows that the proposed algorithms for both cases on cat image 256×256 gives great outcomes PSNR, and UACI. PSNR has been computed for the ciphered image and plain image using the first and second proposed algorithm were 7.8654 and 7.6568 respectively, which means it is hard for an aggressor to recover the plain image. In the other hand,UACI has been computed for the ciphered image and plain image using the first and second proposed algorithm were 31.6912 and 34.0998 respectively, which means it is hard for an aggressor to recover the plain image.

REFERENCES

[1] Fahrnberger, G., “Editing Encrypted Messages without Decrypting or Understanding them,” Ph.D. thesis, University of Hagen, 2019.

[2] A.J. Menezes, et al., “Handbook of Applied Cryptography,” *CRC Press*, 1997.

[3] Jurisic, A. and Menezes, A., “Elliptic Curves and Cryptography,” *Dr. Dobb’s Journal*, pp. 26-36, 1997.

[4] Koblitz, N., “Elliptic Curve Cryptosystems,” *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[5] Miller, V. S., “Use of Elliptic Curves in Cryptography,” *Conference on the Theory and Application of Cryptographic Techniques*, vol. 218, 1986, pp. 417-426.

[6] Rivest, R. L., Shamir, A. and Adleman, L., “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[7] Khan, M. A., et al., “An efficient and provably secure certificateless blind signature scheme for flying ad-hoc network based on multi-access edge computing,” *Electronics*, vol. 9, no. 1, pp. 1-22, 2020.

[8] Koblitz, N., Menezes, A. and Vanstone, S., “The State of Elliptic Curve Cryptography*,” Designs, Codes and Cryptography*, vol. 19, pp. 103-123, 2000.

[9] Ali Soleymani, et al., “A survey on principal aspects of secure image transmission,”*International Journal of Computer, Electrical, Automation, Control and Information Engineerin*g, vol. 6, no. 6, pp. 780-787, 2012.

[10] Gupta, Anvita, et al., “An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1309-1324, 2020.

[11] Jasra, Bhat, and Ayaz Hassan Moon, “Image Encryption techniques: A Review,” *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2020, pp. 221-226.

[12] Naji, Maitham Ali, et al., “Cryptanalysis cipher text using new modeling: Text encryption using elliptic curve cryptography," *AIP Conference Proceedings*, vol. 2203, no. 1, 2020, pp. 1-9.

[13] Boisvert, R., “The Quality of Numerical Software, Assessment and Enhancement,” *Quality of Numerical Software*, pp. 375-380, 1997.

[14] Singh, Laiphrakpam Dolendro and Khumanthem Manglem Singh, “Image encryption using elliptic curve cryptography," *Procedia Computer Science*, vol. 54, pp. 472-481, 2015.

[15] A. Ahmed, Abd El-Latif and Xiamu Niu A., “Hybrid Chaotic System and Cyclic Elliptic Curve for Image Encryption,” In *AEU-International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 136-143, 2013.

[16] S. Nagaraj, et al., “Image encryption using elliptic curve cryptography and matrix,” *Procedia Computer Science*, vol. 48, pp. 276-281, 2015.

[17] Silverman, J. H., “The Arithmetic of Elliptic Curves: Graduate Texts in Mathematics 106,” *2nd edn. New York: Springer*, vol. 106, 2009.

[18] Najlae Hameed Al-Saffar and M. Rushdan, “High Performance Methods of Elliptic Curve Scalar Multiplication,”*International journal of computer applications*, vol. 108, no. 20, pp. 39-45, 2014.

[19] Shomen Deb. and Md. Mokammel Haque, “Elliptic curve and pseudo-inverse matrix based cryptosystem for wireless sensor networks,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4479-4492, 2019.

[20] Faz-Hernández, et al., “High-performance implementation of elliptic curve cryptography using vector instructions,” *ACM Transactions on Mathematical Software*, vol. 45, no. 3, pp. 1-35, 2019.

[21] Hilbert, David, “EinBeitragzurTheorie des Legendre'schenPolynoms,” *ActaMathematica*, vol. 18, pp. 155-159, 1894.

[22] Roopaei, Hadi, “Norm of Hilbert operator on sequence spaces,” *Journal of Inequalities and Applications*, vol. 1, no. 2020, pp. 113, 2020.

[23] Jose, Selby, and Vijay Tiwari., “Study of 2×n right invertible matrix group via Suslin matrices,” *Journal of Xi'an University of Architecture & Technology*, vol. 12, no. 2, pp. 671-678, 2020.

[24] Costara, Constantin, “Nonlinear invertibility preserving maps on matrix algebras,” *Linear Algebra and its Applications*, vol. 602, no. 1, pp. 216-222, 2020.

[25] Acharya, B., et al., “Novel methods of generating self invertible matrix for hill cipher algorithm,” *International Journal of Security*, vol. 1, no. 1, pp. 14-21, 2007.

[26] Ibraheem, Ibraheem Nadher, et al., “Comparative Analysis & Implementation of Image Encryption & Decryption for Mobile Cloud Security,” *International Journal of Advanced Science and Technology*, vol. 29, no. 3s, pp. 109-121, 2020.

[27] Helmrich, Christian R., et al., “Xpsnr: A Low-Complexity Extension of The Perceptually Weighted Peak Signal-To-Noise Ratio For High-Resolution Video Quality Assessment,” *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 2727-2731.

[28] Ali, Tahir Sajjad, and Rashid Ali, “A Novel Medical Image Signcryption Scheme Using TLTS and Henon Chaotic Map,” *IEEE Access*, vol. 8, pp. 71974-71992, 2020.

BIOGRAPHIES OF AUTHORS

Zahraa Kadhim Obaidand received her B.Sc. degree from Kufa University/Iraq, in 2017. She Msc student/research level in the Department of Mathematics/Faculty of Computer Science & Mathematics/University of Kufa/Iraq. Her research interests are Number Theory, Cryptography. Email: zahraak.aljubouri@student.uokufa.edu.iq.

Najlae Falah Hameed Al Saffar received her B.Sc. degree from Kufa University/Iraq, in 1999 and Msc. Degree from Babylon University/Iraq in 2005. She obtained Ph.D. in Mathmatical Crptography UPM University/Malaysia in 2015. She is serving as Assistant Professor, Department of Mathematics/Faculty of Computer Science & Mathematics/University of Kufa/Iraq. She has published more than 15 papers in International and National journals and conference proceedings. Her research interests are Number Theory, Cryptography and their applications also Security and Algebraic Number Field. Email: najlaa.hameed@uokufa.edu.iq.