Journal of King Saud University – Computer and Information Sciences 30 (2018) 349–355

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher

Ziad E. Dawahdeh *, Shahrul N. Yaakob, Rozmie Razif bin Othman

*School of Computer and Communication Engineering, UniMAP University, Perlis, Malaysia*

ARTICLE INFO

*Article history:*
Received 4 April 2017
Revised 4 June 2017
Accepted 23 June 2017
Available online 29 June 2017

*Keywords:*
Elliptic Curve Cryptography
Hill cipher
Self-invertible
Entropy
Peak Signal to Noise Ratio
Unified Average Changing Intensity

ABSTRACT

Image encryption is rapidly increased recently by the increasing use of the internet and communication media. Sharing important images over unsecured channels is liable for attacking and stealing. Encryption techniques are the suitable methods to protect images from attacks. Hill cipher algorithm is one of the symmetric techniques, it has a simple structure and fast computations, but weak security because sender and receiver need to use and share the same private key within a non-secure channels. A new image encryption technique that combines Elliptic Curve Cryptosystem with Hill Cipher (ECCHC) has been proposed in this paper to convert Hill cipher from symmetric technique to asymmetric one and increase its security and efficiency and resist the hackers. Self-invertible key matrix is used to generate encryption and decryption secret key. So, no need to find the inverse key matrix in the decryption process. A secret key matrix with dimensions 4 × 4 will be used as an example in this study. Entropy, Peak Signal to Noise Ratio (PSNR), and Unified Average Changing Intensity (UACI) will be used to assess the grayscale image encryption efficiency and compare the encrypted image with the original image to evaluate the performance of the proposed encryption technique.

## 1. Introduction

Cryptography is one of the mathematical techniques that are used to protect images from adversaries and increase the security of communications. Encryption is done by the sender to convert the original grayscale image to encrypted image before sending it via the internet to the other user (recipient). Decryption is done by the receiver to return the ciphered image back to the original image. Symmetric (private key) and asymmetric (public key) encryption techniques are two groups of cryptography. In symmetric encryption, the same key (private key) is used for both encryption and decryption processes, whereas in asymmetric encryption the sender uses a private key different than the receiver's private key and each party generates the public and secret key separately after agreeing on the elliptic curve domain parameters (Darrel et al., 2004; Bokhari and Shallal, 2016). Both sender and receiver are exchanging their public keys, which are not secret by using Elliptic Curve Diffie-Hellman technique (1976) (Diffie and Hellman, 1976). Elliptic Curve Cryptography (ECC) is one of the effective public key cryptography techniques, it proposed separately by Miller (1985) and Koblitz (1987). The hardness of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) from the adversaries is one of the ECC advantages. ECC works on a small key size with a little amount of memory and low power compared to other systems like RSA (Alese et al., 2012; Gutub and Khan, 2011; Gutub et al., 2007; Gutub, 2003). Hill cipher algorithm is one of the symmetric techniques; it has high throughput, high speed, and simple structure, but weak security because both sender and receiver should use and share the same key (private key) via unsecured channels (Hill, 1929; Acharya et al., 2009).

A lot of researchers tried to develop Hill cipher technique and improve its security. Ismail et al. (2006) proposed a new Hill cipher (HillMRIV) that adjusting the encryption key and using a different key for each plaintext block instead of using one key matrix for all blocks and increasing the security of Hill algorithm, but it has a drawback when the plaintext block contains only zeroes (Ismail et al., 2006). Acharya et al. (2009) solved the decryption problem if the inverse key matrix that does not exist by proposing a novel advanced Hill algorithm (AdvHill) that uses the same involutory

* Corresponding author.
*E-mail addresses:* m_ziad_d@yahoo.com (Z.E. Dawahdeh), ysnizam@gmail.com (S.N. Yaakob), rozmie@unimap.edu.my (R. Razif bin Othman).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2017.06.004
1319-1578/© 2017 Production and hosting by Elsevier B.V. on behalf of King Saud University.

key matrix for encryption and decryption and eliminates the computations needed by the recipient to find the inverse key matrix, and also increased the cipher randomization which increased the efficiency of the algorithm compared with the original Hill cipher (Acharya et al., 2009). Hamissa et al. (2011) enhanced the original Hill cipher algorithm security by using logistic map chaotic functions and proposing a new encoder-decoder technique (ChaoEncoDeco) for images encryption (Hamissa et al., 2011). Panduranga et al. (2012) introduced an approach that consists of three stages including Hill cipher to improve the entropy of the encrypted image. Firstly, each pixel value in two input images is converted to eight binary bits and $k$ bits are rotated and reversed. Next, the lower nibbles of the pixels of the images are exchanged. Finally, Hill cipher algorithm is implemented on the pixel values (Panduranga et al., 2012). Rahman et al. (2013) proposed a new Hill algorithm (Hill++) that computed a random matrix key based on the previous blocks as an extra key for encryption and resisted all zeroes plaintext blocks, it combined Hill cipher with the affine cipher and produced an algorithm that increased attack resistance (Rahman et al., 2013). Agrawal and Gera (2014) produced a new method for encryption by using Hill cipher algorithm first to produce the ciphertext numerical values, and then convert it to points on the ECC by using scalar multiplication. This method increased the security but also increased the time of computations because scalar multiplication consumed a long time (Agrawal and Gera, 2014). Sharma and Chirgaiya (2014) proposed a method to solve the Hill cipher decryption problem if the key matrix is not invertible, they suggested using setting offset value one if the determinant of a matrix is zero and offset value $-1$ if the determinant is negative (Sharma and Chirgaiya, 2014). Mahmoud and Chefranov (2014) proposed an effective modification for the Hill cipher (HCM$-$PRE) that resists known plaintext-ciphertext attack by using pseudo-random eigenvalues and changing key matrix for each block dynamically to make the proposed technique faster than other modifications (Mahmoud and Chefranov, 2014). Rajput and Gulve (2014) proposed a system that consists of three stages; the first stage divides the image into n blocks, then performs XOR between blocks, the pixel value of the image is converted into 8-bit binary in the second stage, and in the last stage the image is encrypted by using the extended hill cipher (Rajput and Gulve, 2014).

A new encryption technique has been proposed in this paper to combine Elliptic Curve Cryptosystem (ECC) with Hill cipher (HC) technique to strengthen the security and produce a new approach (ECCHC) similar in principle to the work proposed in Gutub and Khan (2012). The new approach uses 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 main drawbacks in Hill cipher algorithm is that the inverse of the key matrix does not always exist. So, if the key matrix is not invertible, the decryption process cannot be done, and the receiver cannot get the original data. This paper avoids this problem by using the self-invertible key matrix (the key matrix is self-invertible if $k = k^{-1}$) which reduces the computational process needs during the decryption process to compute key matrix inverse (Acharya et al., 2007). Both sender and receiver construct the self-invertible key matrix and use it for encryption and decryption with no need to generate the inverse of the key matrix. The new technique will be implemented and tested on the grayscale images. The efficiency and performance of the new technique will be assessed by using some security measures like Entropy, Peak Signal to Noise Ratio (PSNR), and Unified Average Changing Intensity (UACI). MATLAB R2013a (8.1.0.604) 32-bit software on Core i5 computer with CPU 2.53 GHz and RAM 4 GB will be used for encryption and decryption processes.

The rest of this paper is organized as follows. An introduction to elliptic curve function is presented in Section 2. Section 3 describes the original Hill cipher algorithm. Section 4 explains the proposed hybrid encryption approach. An implementation example of the proposed approach is given in Section 5. Security Analysis for some measures is explained in Section 6. Finally, the conclusion and the advantages of the proposed approach are shown in Section 7.

## 2. Elliptic curve function

Elliptic Curve Cryptography (ECC) is a suitable encryption technique to be used in portable devices, embedded systems, and mobile devices because it can provide high security with smaller key size and fewer computations with less memory usage and lower power consumptions (Gutub and Khan, 2011; Gutub et al., 2007; Gutub, 2003; Gutub and Ibrahim, 2003).

### 2.1. Definition

An elliptic curve $E$ over a prime field $F_p$ is defined by

$$E : y^2 \equiv x^3 + ax + b(mod\, p)$$

Where $a, b \in F_p$, $p \neq 2, 3$, and satisfy the condition $4a^3 + 27b^2 \neg \equiv 0(\text{modp})$. The elliptic curve group $E(F_p)$consists of all points $(x, y)$ that satisfy the elliptic curve $E$ and the point at the infinity $O$ (Darrel et al, 2004; Hoffstein et al., 2014).

### 2.2. Elliptic curve operations

This section describes the primary operations related to elliptic curve function. Scalar multiplication is the main operation on EC that consumes more time in encryption and decryption operations, it depends on point addition and point doubling (Gutub et al., 2013; Gutub, 2010).

#### 2.2.1. 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 $E$. Adding the two points $P_1$ and $P_2$ giving a third point $R$ that should lie on the same curve $E$ (Agrawal and Gera, 2014; Dawahdeh et al., 2016)

$$R = P_1 + P_2 = (x_3, y_3)$$

where

$$s = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

$$x_3 \equiv (s^2 - x_1 - x_2)(mod\, p)$$

$$y_3 \equiv (sx_1 - sx_3 - y_1)(mod\, p)$$

#### 2.2.2. Point doubling

Adding the point $P = (x_1, y_1)$ that lies on the elliptic curve $E$ to itself is called point doubling. The point $R$ that results from doubling the point $P$ is also lies on the elliptic curve $E$ (Nayak, 2014; Panduranga et al., 2012)

$$R = 2P = P + P = (x_3, y_3)$$

where

$$s = \frac{3x_1^2 + a}{2y_1}$$

$$x_3 \equiv (s^2 - 2x_1)(mod\, p)$$

$$y_3 \equiv (sx_1 - sx_3 - y_1)(mod\, p)$$

### 2.2.3. Scalar multiplication

The scalar multiplication of an integer $k$ by the point $Q = (x_1, y_1)$ that lies on the curve $E$ can be defined by repeating the addition of the point $Q$ to itself $k$ times. The result point $R$ also lies on the elliptic curve $E$.

$$R = kQ = \underbrace{Q + Q + \ldots + Q}_{k-times}$$

For example, computing 15Q can be done using point addition and point doubling as follows (Nayak, 2014):

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

## 3. Hill cipher algorithm

Hill cipher is a symmetric block cipher technique innovated by the mathematician Lester Hill in 1929 (Hill, 1929). Both sender and receiver should share and use the same key matrix for ciphering and deciphering. The main concept of this technique based on assign each letter by a numerical value, for example, a = 0, b = 1, ..., z = 25. Then divide the plaintext (message) into blocks consist of the same size $m$ depending on the key matrix size $m \times m$. For example, if the block size is two ($P_{2\times1}$), then the key matrix ($K_{2\times2}$) should be of size $2 \times 2$, and the encryption process will produce ciphertext block with two numerical values ($C_{2\times1}$) as follows (Agrawal and Gera, 2014):

$$\text{If} \quad P = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \quad \text{and} \quad K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \quad \text{then}$$

$$C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} (k_{11}p_1 + k_{12}p_2)\text{mod } 26 \\ (k_{21}p_1 + k_{22}p_2)\text{mod}26 \end{bmatrix}$$

To decrypt the ciphertext message $C$, the recipient needs to compute the key matrix inverse ($K^{-1}$) where $K.\ K^{-1} = I$, $I$ is the identity matrix, then use the following equation to produce the plaintext $P$ (original message) (Hill, 1929; Acharya et al., 2009)

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

## 4. The proposed cryptosystem

The new proposed approach of Elliptic Curve Cryptosystem and Hill Cipher (ECCHC) has been introduced in this section. This modification increases the security and makes the system more efficient than the original Hill cipher technique, also it speeds up the decryption computations since it does not need the computation of the key matrix inverse. Suppose the sender (User A) wants to send an image $M$ to the other party (User B) using ECCHC over an insecure channel. Firstly, they should agree on the elliptic curve function $E$ and share the domain parameters $\{a, b, p, G\}$, where $a, b$ are the coefficients of the elliptic function, $p$ is a large prime number, and $G$ 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 follows

$$P_A = n_A.G$$

$$P_B = n_B.G$$

Each user multiplies his private key by the public key of the other user to get the initial key $K_I = (x, y)$

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

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 ciphertext. To solve this problem, the self-invertible key matrix will be generated, and the same key will be used for encryption and decryption (the matrix $K$ is self-invertible if $K = K^{-1}$), and no need to find the inverse key matrix. The new approach will be implemented on grayscale images of size 256 × 256 pixels. The image will be divided into blocks of size four pixel values. So, each party produces the 4 × 4 self-invertible key matrix $K_m$ by using the proposed method in Acharya et al. (2007):

$$\text{Let } K_m = \left[\begin{array}{cc:cc} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ \hdashline k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{array}\right] \text{ be self-invertible matrix}$$

partitioned as $K_m = \left[\begin{array}{c:c} K_{11} & K_{12} \\ \hdashline K_{21} & K_{22} \end{array}\right]$. The proposed approach assumes that $K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$, then the values of the other partitions of the secret matrix key $K_m$ is obtained by solving $K_{12} = I - K_{11}$, $K_{21} = I + K_{11}$, and $K_{11} + K_{22} = 0$, 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 $(P_1, P_2, P_3, \ldots)$, then multiply the self-invertible key matrix $K_m$ by each vector and take modulo 256 to get the ciphered vectors $(C_1, C_2, C_3, \ldots)$. After that, reconstruct the ciphered image $C$ from the values in the ciphered vectors and send it to the other party. The following calculations are repeated for each block:

$$\text{Let } P_1 = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \end{bmatrix} \text{ then}$$

$$\begin{aligned} C_1 = K_m.\ P_1 &= \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \end{bmatrix} \\ &= \begin{bmatrix} ((k_{11}p_{11} + k_{12}p_{21} + k_{13}p_{31} + k_{14}p_{41})\text{mod } 256 \\ ((k_{21}p_{11} + k_{22}p_{21} + k_{23}p_{31} + k_{24}p_{41})\text{mod } 256 \\ ((k_{31}p_{11} + k_{32}p_{21} + k_{33}p_{31} + k_{34}p_{41})\text{mod } 256 \\ ((k_{41}p_{11} + k_{42}p_{21} + k_{43}p_{31} + k_{44}p_{41})\text{mod } 256 \end{bmatrix} \\ &= \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \\ c_{41} \end{bmatrix} \end{aligned}$$

Decryption processes start when the recipient receives the ciphered image $C$ by separating the image pixel values into blocks of size four, then arranging each block into four rows column vector. After that, multiplying the self-invertible key matrix $K_m$ by each vector $(C_1, C_2, C_3, \ldots)$ and taking modulo 256 to get the plain image vectors $(P_1, P_2, P_3, \ldots)$ that construct the original image.

### 4.1. The proposed approach (ECCHC)

Step 1: Key Generation
1.1. User A (The sender)
1.1.1. Choose the private key $n_A \in [1, p-1]$
1.1.2. Compute the public key $P_A = n_A.G$
1.1.3. Compute the initial key $K_I = n_A.P_B = (x, y)$
1.1.4. Compute $K_1 = x.G = (k_{11}, k_{12})$ and $K_2 = y.G = (k_{21}, k_{22})$
1.1.5. Generate the self-invertible key matrix $K_m$

1.2. User B (The receiver)

1.2.1. Choose the private key $n_B \in [1, p-1]$
1.2.2. Compute the public key $P_B = n_B.G$
1.2.3. Compute the initial key $K_I = n_B.P_A = (x, y)$
1.2.4. Compute $K_1 = x.G = (k_{11}, k_{12})$ and $K_2 = y.G = (k_{21}, k_{22})$
1.2.5. Generate the self-invertible key matrix $K_m$

Step 2: Encryption (User A)
2.1. Separate the original image pixel values into blocks of size four.
2.2. Arrange each block into four rows column vector $(4 \times 1)$.
2.3. Multiply the self-invertible key matrix $K_m$ by each vector $(P_1, P_2, P_3, \ldots)$ and take modulo 256 for each value $C_1 = (K_m.\ \ P_1)$ mod 256.
2.4. Construct the ciphered image $C$ from the values in the ciphered vectors $(C_1, C_2, C_3, \cdots)$.
Step 3: Decryption (User B)
3.1. Separate the ciphered image pixel values into blocks of size four.
3.2. Arrange each block into four rows column vector $(4 \times 1)$.
3.3. Multiply the self-invertible key matrix $K_m$ by each vector $(C_1, C_2, C_3, \ldots)$ and take modulo 256 for each value $P_1 = (K_m.\ \ C_1)$ mod 256.
3.4. Construct the original image from the values in the deciphered vectors $(P_1, P_2, P_3, \cdots)$.

**Table 1**
The points of the function $E : y^2 \equiv x^3 + x + 3(mod\ 31)$.

| | | | | |
|---|---|---|---|---|
| (1, 6) | (6, 15) | (15, 13) | (21, 4) | (26, 11) |
| (1, 25) | (6, 16) | (15, 18) | (21, 27) | (26, 20) |
| (3, 8) | (9, 11) | (17, 2) | (22, 3) | (27, 11) |
| (3, 23) | (9, 20) | (17, 29) | (22, 28) | (27, 20) |
| (4, 3) | (12, 10) | (18, 5) | (23, 14) | (28, 2) |
| (4, 28) | (12, 21) | (18, 26) | (23, 17) | (28, 29) |
| (5, 3) | (14, 8) | (20, 5) | (24, 5) | (30, 1) |
| (5, 28) | (14, 23) | (20, 26) | (24, 26) | (30, 30) |

## 5. Implementation example

Assume that User A wants to send an image $M$ to User B and they agreed to use the elliptic curve function

$$E : y^2 \equiv x^3 + x + 3(mod\ 31)$$

where $A = 1, B = 3, p = 31$; which satisfies the condition $4A^3 + 27B^2 = 4(1)^3 + 27(3)^2 = 4 + 243 = 247\ \ mod\ \ 31 = 30 \neq 0$. The elliptic curve $E_{31}(1, 3)$ points are shown in Table 1 (Dawahdeh et al., 2016).

Since the order of the elliptic curve $E_{31}(1, 3)$ is 41, which is a prime number, any point from Table 1 can be chosen to represent the base point or generator point $G$. So, if we choose $G = (1, 6)$, the domain parameters for $E$ are $\{A, B, p, G\} = \{1, 3, 31, (1, 6)\}$.

If User A wants to send the grayscale image (Lena 256 × 256) to User B. Both sender and receiver should apply the proposed approach (ECCHC) on the image as shown in the next steps. Fig. 1 shows the original image, ciphered image, and deciphered image with their histograms. MATLAB R2013a (8.1.0.604) 32-bit software on Core i5 computer with CPU 2.53 GHz and RAM 4 GB is used for encryption and decryption processes.

### 5.1. The proposed approach (ECCHC)

Step 1: Key Generation
User A

1. Choose the private key $n_A = 13 \in [1, 30]$
2. Compute the public key$P_A = n_A.G = 13(1, 6) = (3, 23)$
3. Compute the initial key $K_I = n_A.P_B = 13(24, 5) = (20, 5) = (x, y)$

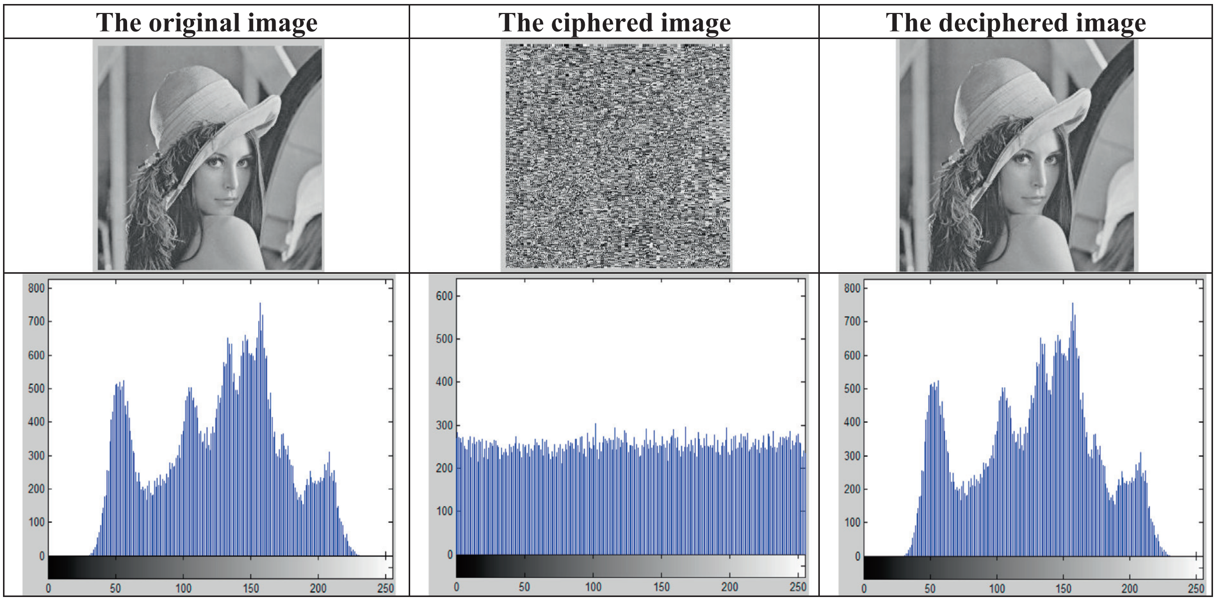

**Fig. 1.** The original image, ciphered image, and deciphered image with their histograms for Lena image.

4. Compute $K_1 = x.G = 20(1,6) = (4,28) = (k_{11}, k_{12})$ and $K_2 = y.G = 5(1,6) = (15,18) = (k_{21}, k_{22})$
5. Assume that $K_{11} = [\begin{smallmatrix} 4 & 28 \\ 15 & 18 \end{smallmatrix}]$, then the self-invertible key matrix

$$K_m = [\begin{matrix} 4 & 28 & 253 & 228 \\ 15 & 18 & 241 & 239 \\ 5 & 28 & 252 & 228 \\ 15 & 19 & 241 & 238 \end{matrix}]$$

User B

1. Choose the private key $n_B = 17 \in [1,30]$
2. Compute the public key $P_B = n_B.G = 17(1,6) = (24,5)$
3. Compute the initial key $K_I = n_B.P_A = 17(3,23) = (20,5) = (x,y)$
4. Compute $K_1 = x.G = 20(1,6) = (4,28) = (k_{11}, k_{12})$ and $K_2 = y.G = 5(1,6) = (15,18) = (k_{21}, k_{22})$
5. Assume that $K_{11} = [\begin{smallmatrix} 4 & 28 \\ 15 & 18 \end{smallmatrix}]$, then the self-invertible key matrix

$$K_m = [\begin{matrix} 4 & 28 & 253 & 228 \\ 15 & 18 & 241 & 239 \\ 5 & 28 & 252 & 228 \\ 15 & 19 & 241 & 238 \end{matrix}]$$

Step 2: Encryption (User A)

1. Separate Lena image pixel values into blocks of size four.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 2 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 3 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 4 | 163 | 165 | 163 | 162 | 162 | 161 | 158 | 159 | --- |
| 5 | : | : | : | : | : | : | : | : | --- |

2. $P_1 = \begin{bmatrix} 165 \\ 165 \\ 165 \\ 166 \end{bmatrix}, P_2 = \begin{bmatrix} 168 \\ 165 \\ 158 \\ 162 \end{bmatrix},$
3. The multiplication of $K_m$ by the first vector $P_1$ will be done, and the same process will be repeated for the other vectors.

$$C_1 = K_m.\ P_1 = \begin{bmatrix} 4 & 28 & 253 & 228 \\ 15 & 18 & 241 & 239 \\ 5 & 28 & 252 & 228 \\ 15 & 19 & 241 & 238 \end{bmatrix} \begin{bmatrix} 165 \\ 165 \\ 165 \\ 166 \end{bmatrix} \text{mod } 256 = \begin{bmatrix} 137 \\ 148 \\ 137 \\ 147 \end{bmatrix}$$

4. The pixel values for the encrypted image are

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 2 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 3 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 4 | 247 | 216 | 247 | 219 | 230 | 255 | 234 | 1 | --- |
| 5 | : | : | : | : | : | : | : | : | --- |

Step 3: Decryption (User B)

1. Separate the ciphered image pixel values into blocks of size four.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 2 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 3 | 137 | 148 | 137 | 147 | 26 | 110 | 36 | 113 | --- |
| 4 | 247 | 216 | 247 | 219 | 230 | 255 | 234 | 1 | --- |
| 5 | : | : | : | : | : | : | : | : | --- |

2. $C_1 = \begin{bmatrix} 137 \\ 148 \\ 137 \\ 147 \end{bmatrix}, C_2 = \begin{bmatrix} 26 \\ 110 \\ 36 \\ 113 \end{bmatrix},$
3. The multiplication of $K_m$ by the first vector $C_1$ will be done, and the same process will be repeated for the other vectors.

$$P_1 = K_m.\ C_1 = \begin{bmatrix} 4 & 28 & 253 & 228 \\ 15 & 18 & 241 & 239 \\ 5 & 28 & 252 & 228 \\ 15 & 19 & 241 & 238 \end{bmatrix} \begin{bmatrix} 137 \\ 148 \\ 137 \\ 147 \end{bmatrix} \text{mod } 256 = \begin{bmatrix} 165 \\ 165 \\ 165 \\ 166 \end{bmatrix}$$

4. The pixel values for the decrypted image are

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 2 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 3 | 165 | 165 | 165 | 166 | 168 | 165 | 158 | 162 | --- |
| 4 | 163 | 165 | 163 | 162 | 162 | 161 | 158 | 159 | --- |
| 5 | : | : | : | : | : | : | : | : | --- |

## 6. Security analysis

There are some measures (parameters) used to assess the grayscale image encryption efficiency and compare the encrypted image with the original image to evaluate the performance of the encryption approach. A comparison between the proposed approach and the existing approaches (previous approaches) has been done. Various methods are applied in this research for the evaluation of the encryption efficiency, for instance, Entropy, Peak Signal to Noise Ratio (PSNR), and Unified Average Changing Intensity (UACI). These measurements are described in the next subsections.

### *6.1. The entropy*

Entropy is one of the statistical scalar parameters used for the image encryption evaluation. It shows the most frequency occurring patterns. It depends on the probability of the pixels values and measures the degree of randomness. The theoretical and ideal entropy value for the grayscale image of size 256 × 256 is eight, and the encrypted image efficiency is better if the entropy value is closed to eight. In general, the greater the entropy, the harder it becomes to break the cryptosystem (Panduranga et al., 2012). The following formula is used to calculate entropy

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

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

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

### 6.2. Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio is used to assess the performance of an image encryption algorithm. PSNR reflects the encryption quality and measures the distortion in the decrypted image compared with the original image. Higher PSNR value means the loss data in the decrypted image is zero or negligible, and this indicates that the decrypted image is identical to the original image, which leads to the higher efficiency of the encryption technique (Rajput and Gulve, 2014). The following formula is used to compute PSNR

$$PSNR = 20 \times \log_{10}\left[\frac{255}{MSE}\right]$$

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

$$MSE = \frac{1}{256 \times 256}\sum_{i=1}^{256}\sum_{j=1}^{256}(A_{ij} - B_{ij})^2$$

where $A_{ij}$ is the pixel value of the original image and $B_{ij}$ is the pixel value of the decrypted image (Panduranga et al., 2012). When comparing the original image with the encrypted image; if MSE increases, then PSNR decreases, and this indicates that the encrypted image is more randomness (Naveen Kumar et al., 2012). The high value of MSE and low value of PSNR indicates that the two images are different and not identical, and this leads to an efficient encryption technique (Naskar and Chaudhuri, 2014).

### 6.3. Unified Average Changing Intensity (UACI)

The difference between the original image and the ciphered image is measured by UACI. It is used to assess the strength of the encryption technique. Its value depends on the size and format of the image (Wu et al., 2011). UACI measures the average changing in intensity between the original and ciphered images. The highest UACI means that the proposed technique is resistant against differential attacks. UACI is calculated for the grayscale image of size 256 × 256 by the following equation:

$$UACI = \frac{1}{256 \times 256}\sum_{i=1}^{256}\sum_{j=1}^{256}\frac{|A(i,j) - B(i,j)|}{255} \times 100\%$$

where *A(i, j)* is the pixel value of the original image and *B(i, j)* is the pixel value of the encrypted image (Panduranga and Naveen Kumar, 2012; Naveen Kumar et al., 2012).

A comparison between the proposed technique and some other techniques for Lena grayscale image is shown in Table 2. It is clear that the Entropy in the proposed technique is higher than the other techniques and it is nearest to the theoretical value eight. PSNR and UACI values in the proposed technique are also better than Panduranga (Panduranga and Naveen Kumar, 2012) and Naveen Kumar (Naveen Kumar et al., 2012) techniques. From Table 2 we can conclude that the proposed technique is more efficient than the other techniques.

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

From Table 3, It is cleared that the entropy values in the table are closed to the ideal and theory value for entropy which is 8. Also, it is shown that the values of PSNR are low values and this reflects the encryption quality and measures the distortion in the decrypted image compared with the original image and indicates that the proposed technique is efficient. UACI measures the difference between the original image and the ciphered image. It is used to assess the strength of the encryption technique. The expected value for UACI is 33.46 for the grayscale images of size 256 × 256. Table 3 shows that the values of UACI are closed to the expected value 33.46 and this indicates the strength of the proposed approach against the adversaries.

**Table 2**
The Entropy, PSNR, and UACI for Lena Image.

| The Method | Lena Image (256 × 256) | | |
|---|---|---|---|
| | Entropy | PSNR | UACI |
| The proposed technique | 7.9970 | 8.5777 | 30.4814 |
| Panduranga and Naveen Kumar (2012) | 7.9961 | 27.6689 | 38.3301 |
| Ramyashree and Manjunatha | 7.9736 | 4.8909 | NA |
| Naveen Kumar et al. (2012) | NA | 8.6092 | 49.8 |
| Expected values | 8 | Minimum value is better | 33.46 |

**Table 3**
Image security measures for ECCHC.

| The image | Entropy | PSNR | UACI |
|---|---|---|---|
| Lena | 7.9970 | 8.5952 | 30.3842 |
| Cameraman | 7.9848 | 6.9999 | 35.5263 |
| Einstein | 7.9899 | 9.7483 | 26.9087 |
| Smandril | 7.9968 | 9.7208 | 27.3588 |

**Table 4**
Encryption and Decryption time for ECCHC.

| The image | Encryption and Decryption time (seconds) |
|---|---|
| Lena | 1.2615 |
| Cameraman | 1.2588 |
| Einstein | 1.2736 |
| Smandril | 1.2635 |

Table 4 shows the time consumed in encryption and decryption processes on different grayscale images of size 256 × 256. It is cleared that the proposed technique needs a little time for encryption and decryption and this indicates the efficiency of this method and the low time it needs in calculations.

## 7. Conclusions

Information security is one of the most important issues in the recent times. Elliptic Curve Cryptography (ECC) is one of the most efficient public key cryptosystems that is secured against adversaries because it is hard for them to find the secret key and solve the elliptic curve discrete logarithm problem. Its strengthened security also comes from the small key size that is used in it with the same level of safety compared to the other cryptosystems like RSA.

A novel approach cryptosystem (ECCHC) has been proposed in this paper combining ECC with standard Hill cipher algorithm to enhance and increase the security of the original Hill cipher for image encryption. It generates a new encryption/decryption key by using ECC approach which produces a strong secret key that resistant against intruders and provides better security because no need to share the key through the internet. Self-invertible key matrix is used for encryption and decryption. So, no need to find the inverse key matrix in the decryption process. Table 2 shows that the proposed approach on Lena image gives good results for Entropy, PSNR, and UACI better than other techniques. The Entropy value in the proposed approach 7.9970 is the nearest to the expected value 8. PSNR value 8.5777 is better in the proposed technique than Panduranga and Naveen Kumar techniques. Also, UACI value 30.4814 is the nearest to the expected value. Table 3 shows good results for other three images, and this supports the efficiency of the proposed technique. The proposed approach key matrix depends on the ECC and it's hard to solve the elliptic curve discrete logarithm problem to get it. So, the new approach is efficient and resistant against different attacks. Table 4 shows the little time consumed in encryption and decryption processes, and this also indicates that the proposed method does not consumed more time in its calculations.

The proposed approach can be used efficiently in wireless applications and suitable for small devices and embedded systems because it has a simple structure and faster computations. In this paper, we applied the new approach on the grayscale images, and in the future work, the proposed technique will be modified to be used for RGB images and real-time multimedia applications.

## References

Acharya, B., Rath, G.S., Patra, S.K., Panigrahy, S.K., 2007. Novel methods of generating self-invertible matrix for hill cipher algorithm. Int. J. Secur. 1 (1).

Acharya, B., Panigrahy, S.K., Patra, S.K., Panda, G., 2009. Image encryption using advanced hill cipher algorithm. Int. J. Recent Trends Eng. 1 (1).

Agrawal, K., Gera, A., 2014. Elliptic curve cryptography with Hill cipher generation for secure text cryptosystem. Int. J. Comput. App. 106 (1).

Alese, B.K., Philemon, E.D., Falaki, S.O., 2012. Comparative analysis of public-key encryption schemes. Int. J. Eng. Technol. 2 (9), 1552–1568.

Bokhari, M.U., Shallal, Q.M., 2016. A review on symmetric key encryption techniques in cryptography. Int. J. Comput. Appl. 147 (10).

Darrel, H., Alfred, M., Scott, V., 2004. Guide to elliptic curve cryptography. Hankerson Darrel, Menezes Alfred J., Vanstone Scott. Springer-Verlag Professional Computing Series. 2004. p. 11.

Dawahdeh, Z.E., Yaakob, S.N., Othman, R.R.B., 2016. A new modification for menezes-vanstone elliptic curve cryptosystem. J. Theor. Appl. Inf. Technol. 85 (3).

Diffie, W., Hellman, M., 1976. New directions in cryptography. IEEE Trans. Inf. Theory 22 (6), 644–654.

Gutub, A.A.A., 2003. HIGH speed low power GF (2 k) elliptic curve cryptograpy processor architecture. In: IEEE 10th Annual Technical Exchange Meeting, KFUPM. Dhahran, Saudi Arabi.

Gutub, A.A.A., 2010. Preference of efficient architectures for GF (p) elliptic curve crypto operations using multiple parallel multipliers. Int. J. Secur. (IJS) 4 (4), 46.

Gutub, A.A.A., Ibrahim, M.K., 2003. Power-time flexible architecture for GF (2 k) elliptic curve cryptosystem computation. In: Proceedings of the 13th ACM Great Lakes Symposium on VLSI. ACM, pp. 237–240.

Gutub, A.A.A., Khan, E.A., 2011. Using subthreshold SRAM to design Low-Power crypto hardware. Int. J. New Comput. Archit. Appl. (IJNCAA) 1 (2), 474–483.

Gutub, A.A.A., Khan, F.A.A., 2012. Hybrid Crypto Hardware Utilizing Symmetric-Key and Public-Key Cryptosystems. In: Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on. IEEE, pp. 116–121.

Gutub, A.A.A., Ibrahim, M.K., Al-Somani, T.F., 2007. Parallelizing GF (P) elliptic curve cryptography computations for security and speed. In: Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on. IEEE, pp. 1–4.

Gutub, A.A.A., Tabakh, A.A., Al-Qahtani, A., Amin, A., 2013. Serial vs. parallel elliptic curve crypto processor designs. In: IADIS International Conference Applied Computing.

Hamissa, G., Sarhan, A., Abdelkader, H., Fahmy, M., 2011. Securing JPEG architecture based on enhanced chaotic hill cipher algorithm. In: Computer Engineering & Systems (ICCES), 2011 International Conference on. IEEE, pp. 260–266.

Hill, L.S., 1929. Cryptography in an algebraic alphabet. Am. Math. Mon. 36 (6), 306–312.

Hoffstein, J., Pipher, J., Silverman, J.H., 2014. Elliptic curves and cryptography. In: An Introduction to Mathematical Cryptography. Springer, New York, pp. 299–371.

Ismail, I.A., Amin, M., Diab, H., 2006. How to repair the Hill cipher. J. Zhejiang Univ. Sci. A 7 (12), 2022–2030.

Koblitz, N., 1987. Elliptic curve cryptosystems. Math. Comput. 48, 203–209.

Mahmoud, A., Chefranov, A., 2014. Hill cipher modification based on pseudo-random eigenvalues. Appl. Math. 8 (2), 505–516.

Miller, V.S., 1985. Use of elliptic curves in cryptography. In: Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin Heidelberg, pp. 417–426.

Naskar, P.K., Chaudhuri, A., 2014. A secure symmetric image encryption based on bit-wise operation. Int. J. Image, Graphics Signal Process. 6 (2), 30.

Naveen Kumar, S.K., Sharath Kumar, S.K., Panduranga, H.T., 2012. Encryption approach for images using bits rotation reversal and extended hill cipher techniques. Int. J. Comput. App. 59 (16).

Nayak, Biswojit, 2014. Signcryption schemes Based on Elliptic Curve Cryptography (Master Thesis). National Institute of Technology Rourkela, India.

Panduranga, H.T., Naveen Kumar, S.K., 2012. Advanced partial image encryption using two-stage hill cipher technique. Int. J. Comput. App. 60 (16).

Panduranga, H.T., Kumar, H.S., Kumar, S.N., 2012. Hybrid approach for dual image encryption using nibble exchange and Hill-cipher. In: Machine Vision and Image Processing (MVIP), 2012 International Conference on. IEEE, pp. 101–104.

Rahman, M.N.A., Abidin, A.F.A., Yusof, M.K., Usop, N.S.M., 2013. Cryptography: a new approach of classical Hill cipher. Int. J. Secur. App. 7 (2), 179–190.

Rajput, Y., Gulve, A.K., 2014. A comparative performance analysis of an image encryption technique using extended Hill cipher. Int. J. Comput. App. 95 (4).

Ramyashree, A.N., Manjunatha, C.N. An Adaptive Dual Image Encryption. http://ijsetr.com/uploads/165423skit%20paper.docx.

Sharma, N., Chirgaiya, S., 2014. A novel approach to Hill cipher. Int. J. Comput. Appl. 108 (11).

Wu, Y., Noonan, J.P., Agaian, S., 2011. NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J. Sel. Areas Telecommun. (JSAT), 31–38.