An Image Encryption Scheme Based on Elliptic Curve and a Novel Mapping Method

Ali Soleymani, Md Jan Nordin, Azadeh Noori Hoshyar, Zulkarnain Md Ali, Elankovan Sundararajan

Abstract

Images are an attractive data type that occasionally include secret information, such as faces, logos, signatures, places, or personal private albums. Cryptography is a solution to protect confidential images by encrypting them before transmission over unsecure channels or public networks. This paper’s proposed cryptosystem is based on elliptic curves, which is a public key technique. Elliptic Curve Cryptography (ECC) is based on computational operations (Add, Double, Multiply) on the points that lie on a predefined elliptic curve. Therefore, converting a message (pixel) to a coordinate on the affined curve is a mandatory prerequisite for any ECC-based encryption. This cryptosystem utilizes the novel proposed mapping method to convert the pixels of a plain image into the coordinates of points on the curve. Creating the map table, the converting process, and the encryption itself, are given in detail; along with their implementation. Finally, security analysis is performed to evaluate the strength of the proposed technique to statistical attacks.

Keywords: Image, Encryption, Decryption, Elliptic Curve, Prime Group Field

1. Introduction

In 1985, elliptic curves independently applied by Miller [1] and Koblitz [2] to introduce a new public key cryptosystem. Following this, many researchers tried to employ it on different data types; and thus improve their efficiency by proposing various techniques. In fact, ECC has attractive advantages that motivated cryptographers to use it. The most important of these benefits is greater security and a more computationally efficient performance, with equivalent key sizes, in comparison to other public keys. This characteristic changed ECC into an acceptable choice for real time multimedia applications. Due to large sizes and high data rates of multimedia data types, such as images, videos, and audio, a cryptosystem using a short key size, with high security, was needed.

ECC is a public key (or asymmetric) cryptography. This means that the encryption and decryption keys are different. Unlike private key cryptography, ECC is appropriate for applications where a secure channel is not available to transmit the private key. Figure 1 shows that ECC is an asymmetric cryptosystem that provides security and fast execution, rather than RSA.

Digital images are an attractive data type that offers a widespread range of use. Many users are interested in implementing content protection methods to their images, in order to restrict preview, copyright, or manipulation. In many applications, like military image databases, confidential video conferencing, medical imaging systems, cable TV, and online personal photograph albums, security is essential. Furthermore, the wide application of images within industrial processes turns it into a resource and asset. It is therefore important to protect confidential image data from unauthorized access.

In some applications, a secure channel is not available to transmit the private key, or other entities prefer to keep the decryption key secret. Hence, we have to use public key cryptography. Several existing researches, titled as public key image encryption, use a symmetric cryptosystem to encrypt the image, and only a public key cryptosystem is applied as a key exchange protocol.
Zhu et al. [3] tried to encrypt an image by scrambling pixels and adding a watermark to the scrambled image. Finally, they encrypted scrambling and watermarking parameters using ECC. Gupta and Silakari [4] proposed a symmetric encryption based on chaos maps and XOR operation. They used a 3D standard map to create a diffusion template, and 3D cat map and a standard map for shuffling color images. Next the shuffled image XORed using a diffusion template to encrypt the image. Finally, chaos parameters were encrypted using ECC for a secure exchange.

In this study, an ECC-based cryptosystem is proposed to encrypt the image. The most important phase in encryption (using ECC) is mapping a message to a point on the curve and converting the encrypted point to the current message type. Gupta et al. in [5] converted pixels to point \((X_m, Y_m)\) using:

\[
X_m = mk + j \tag{1}
\]

\[
Y_m = \sqrt{x^3 + ax + b}, j = 0, 1, 2, 3, \ldots
\]

To solve the equation (1), this process continues to find the first \(j\) which satisfy the \(Y_m\). This process was completely explained and implemented by Padma et al. [6], as the Koblitz Method.

Amounas and Kinani [7] proposed a new mapping technique based on matrix properties. In this methodology, the message with length \(n\) was divided by 3, and a matrix of \(M_{3 \times r}\) is created. Next, a non-singular matrix of \(A_{3 \times 3}\) that \(|A| = \pm 1\) multiplied by message matrix and the result \(Q = M.A\), is a matrix of mapped points; and encryption will be performed on the elements of matrix \(Q\). After decryption, in order to decode the decrypted points to the message, matrix \(Q\) is multiplied by inverse of \(A\), \(M = A^{-1}Q\). This method is proposed for alphanumeric characters. Another mapping technique by Amounas and Kinani [8] is also based on matrix.

Rao et al. in [9] proposed two mapping methods. Static mapping method, which is one-to-one was found to be very weak; but in dynamic mapping, for one character, different options were available to choose as a point. In this case, by having the mapped point, it is very difficult to find the corresponding character of the plain message.

Gupta et al.’s study [10] provided a scheme for medical image transmission. This scheme is a combination of DCT transformation, quantization, compression, ECC encryption, and error detection and correction. In this research, after compression the image behaves as plaintext and will be encrypted.

In this study, a new mapping scheme is proposed to convert pixels of input image to a point on an elliptic curve. This mapping scheme is on a map table, which is created and used for both encryption and decryption processes.

**Figure 1.** Estimate the security level of ECC and RSA in MIPS years
2. A Review of Mathematical Basics

An Elliptic curve is a cubic equation [9] with the form (2):

\[ y^2 = x^3 + ax + b \]  \hspace{1cm} (2)

![Graph of an elliptic curve](image)

Figure 2. Graph of an elliptic curve

where, \( a \) and \( b \) are integers that satisfy (3) and \( p \) is a large prime number. Figure 2 shows an elliptic curve over the real field \( \mathbb{R} \) and point addition \((p_1 + p_2)\) on an elliptic curve:

\[ 4a^2 + 27b^2 \neq 0 \mod p \]  \hspace{1cm} (3)

To encrypt a message, Alice and Bob chose an elliptic curve and took an affine point \((G)\) that lies on the curve. Plaintext \( M \) is encoded into a point \( P_M \). Alice chose a random prime integer \( x \) and Bob chose a random prime integer \( y \). \( x \) and \( y \) are Alice and Bob’s private keys, respectively. To generate the public key, Alice computes (4) and Bob Computes (5).

\[ P_A = xG \]  \hspace{1cm} (4)

\[ P_B = yG \]  \hspace{1cm} (5)

To encrypt a message point \( P_M \) for Bob, Alice chooses another random integer \( k \) and computes the encrypted message \( P_C \) using Bob’s public key \((P_B)\). \( P_C \) is a pair of points (6):

\[ P_C = [(kG), (P_M + kP_B)] \]  \hspace{1cm} (6)

Alice Sends the encrypted message \( P_C \) to Bob. He receives the ciphered message and multiplying his private key, \( y \), with \( kG \) and subtract it from the second point in the encrypted message to compute \( P_M \). The result is corresponds to the plaintext message \( M \) (7):

\[ P_M = (P_M + kP_B) - [ykG] \]  \hspace{1cm} (7)

Addition operation for two points \( P \) and \( Q \) over an elliptic group; if \( P + Q = (X_3, Y_3) \) is given by (8):

\[ X_3 = \lambda^2 - X_P - X_Q \mod p \]  \hspace{1cm} (8)

\[ Y_3 = \lambda(X_P - X_3) - Y_P \mod p \]
Multiplication \( kP \) over an elliptic group is computed by repeating the addition operation \( k \) times using (8). The strength of an ECC-based cryptosystem depends on the difficulty of finding the number of times \( G \) is added to itself to get \( P \). This reverse operation is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP) and is exploited in cryptography.

3. Proposed Algorithm Implementation

3.1. Mapping Methodology

Every image consists of pixels. In grayscale images, each pixel has an 8-bit value of between 0 and 255. A pixel in color images is represented by 3 octet values separately; indicate the Red, Green and Blue intensity.

To encrypt an image using ECC, each pixel should be considered as a message and mapped to a point on a predefined elliptic curve. Proposed mapping method in this study is based on a map table. To create this table, the elliptic group \( E_p(a, b) \), which is all possible points on the finite field, is generated first, and then these points are placed into 256 groups. Each group has \( \lceil \frac{N}{256} \rceil \) members. The row indexes start from 0 and end with 255. Each row stands for a pixel intensity value; however, for same values, there are multiple points. If \( N \) is not a multiple of 256, then extra rows in the last column are filled with zero, and the last column will not be consider for mapping.

Starting from the first pixel in the plain image, the corresponding point with the intensity value in the table is mapped to this pixel and continues to the last pixel. For repetitive intensity values, the next point in the corresponding row is selected. For all intensities, if all \( N-1 \) points are selected, then for next one we start from the first again.

After mapping all pixels to their related point one by one, the next step is to encrypt these points using the receiver’s public key and equation (6). Encrypting a point on an elliptic curve results in a set of two points. In this case, the first point \( (kG) \) in (6) is the same for all pixels, but the second point \( (P_M+kP_B) \) is different for each pixel. After encrypting all pixels, the result can be shown as an image. The final step is releasing the encrypted points as an image. We refer to the mapping table and find out the current index according to each point and replace it with the related value.

3.2. Implementation Method

In order to show and define the implementation steps clearly, both the sender and the receiver decide on a simple elliptic curve \( E_{123457} \) \((5376,2438)\) that is represented by:

\[
y^2 \equiv x^3 + 5376x + 2438 \pmod{123457}
\]

Table 1 shows a part of the generated points. To create the mapping table, the first point is placed into row 0, which corresponds to the pixel with an intensity value of 0, and then continues to the next point with the next value. After placing the first 256 points in the first column of the table, the next 256 points will be placed in the second column, and continue until the last point is reached. In this example, there are 123387 points on the curve. These points completely fill 481 columns and 250 rows of the 482nd column. The remaining free spaces of the last column are filled with zeros.

\[
E_{123457} \ (5376,2438) \\
= \{(42908,0),(95914,0),(108092,0),(3,31443),(5,11660),(6,2174),(7,58403),(8,29200),
(10,54073),(11,11372),(13,20768),(14,17567),(15,57945), ...,(12344,108644),
(12344,90529),(123445,83950),(123446,92960),(123448,77200),(123452,110966),
(123453,72536),(123454,82304),(123455,114527),(123456,95491)\}
\]
According to (4) and (5), to encrypt this image, some parameters should be defined as follows: \( G = (2225, 75856) \) as a generator point, \( y = 36548 \) as the receiver’s private key, and \( k = 23412 \) as a random integer defined by the sender. Using these values, according to (5), the receiver’s public key is calculated giving the result: \( P_B = (30402, 35513) \).

Before encrypting an image using this technique, all pixels are mapped into corresponding points using Table 1. The first row of Table 2 shows 9 pixel’s intensity value of the Lena image in position \((1,1)\) to \((1,10)\). The second row shows the result of mapping each pixel’s value to a point on elliptic curve using Table 1. After encrypting all the points using (6), the results are shown in the third row. Encrypting a point results in two points. The first point is the same for all pixels and will be sent once, but the second point is different for each pixel. In order to show the encrypted points as an image, first we create a matrix the same size as that of the image, to find each point on the Table 1 and then place the row index in the equivalent element of the created matrix. The last row in Table 2 contains the converted values from the encrypted points to encrypt the pixel’s value, which can then be viewed as an image.

### Table 1. Mapping Table

<table>
<thead>
<tr>
<th>Index</th>
<th>1st Mapping</th>
<th>2nd Mapping</th>
<th>3rd Mapping</th>
<th>4th Mapping</th>
<th>5th Mapping</th>
<th>482nd Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(42908,0)</td>
<td>(512,47183)</td>
<td>(1033,54418)</td>
<td>(1533,9490)</td>
<td>(2093,30783)</td>
<td>(122949,83868)</td>
</tr>
<tr>
<td>1</td>
<td>(95914,0)</td>
<td>(513,33718)</td>
<td>(1035,9194)</td>
<td>(1534,56042)</td>
<td>(2095,41465)</td>
<td>(122951,74372)</td>
</tr>
<tr>
<td>2</td>
<td>(108092,0)</td>
<td>(515,24882)</td>
<td>(1039,3322)</td>
<td>(1535,33470)</td>
<td>(2096,16779)</td>
<td>(122953,121769)</td>
</tr>
<tr>
<td>3</td>
<td>(3,31443)</td>
<td>(516,49743)</td>
<td>(1041,8203)</td>
<td>(1543,36384)</td>
<td>(2097,9443)</td>
<td>(122954,66970)</td>
</tr>
<tr>
<td>4</td>
<td>(5,11660)</td>
<td>(519,6902)</td>
<td>(1043,46883)</td>
<td>(1544,10278)</td>
<td>(2098,19721)</td>
<td>(122958,73556)</td>
</tr>
<tr>
<td>5</td>
<td>(6,2174)</td>
<td>(520,20390)</td>
<td>(1044,52089)</td>
<td>(1546,38337)</td>
<td>(2099,57887)</td>
<td>(122959,84424)</td>
</tr>
<tr>
<td>6</td>
<td>(7,58403)</td>
<td>(521,20390)</td>
<td>(1046,3610)</td>
<td>(1548,55400)</td>
<td>(2100,39297)</td>
<td>(122961,71950)</td>
</tr>
<tr>
<td>7</td>
<td>(8,29200)</td>
<td>(524,59065)</td>
<td>(1049,55356)</td>
<td>(1550,3312)</td>
<td>(2102,15631)</td>
<td>(122962,91690)</td>
</tr>
<tr>
<td>154</td>
<td>(305,46853)</td>
<td>(824,9038)</td>
<td>(1339,50036)</td>
<td>(1896,22466)</td>
<td>(2412,59464)</td>
<td>(123276,99876)</td>
</tr>
<tr>
<td>155</td>
<td>(306,33458)</td>
<td>(825,50433)</td>
<td>(1340,25625)</td>
<td>(1897,20281)</td>
<td>(2413,4124)</td>
<td>(123277,107283)</td>
</tr>
<tr>
<td>156</td>
<td>(307,29631)</td>
<td>(831,5746)</td>
<td>(1341,37870)</td>
<td>(1898,37575)</td>
<td>(2414,49935)</td>
<td>(123279,104846)</td>
</tr>
<tr>
<td>157</td>
<td>(312,43431)</td>
<td>(832,39441)</td>
<td>(1342,48041)</td>
<td>(1899,60234)</td>
<td>(2417,26883)</td>
<td>(123280,82736)</td>
</tr>
<tr>
<td>158</td>
<td>(314,37257)</td>
<td>(834,26653)</td>
<td>(1344,60034)</td>
<td>(1902,7652)</td>
<td>(2418,11842)</td>
<td>(123285,119446)</td>
</tr>
<tr>
<td>159</td>
<td>(315,58283)</td>
<td>(835,23727)</td>
<td>(1346,20117)</td>
<td>(1905,19609)</td>
<td>(2420,6170)</td>
<td>(123287,87751)</td>
</tr>
<tr>
<td>160</td>
<td>(317,57467)</td>
<td>(836,30492)</td>
<td>(1351,35977)</td>
<td>(1906,43788)</td>
<td>(2422,33537)</td>
<td>(123288,84108)</td>
</tr>
<tr>
<td>161</td>
<td>(318,23904)</td>
<td>(840,29931)</td>
<td>(1355,30658)</td>
<td>(1908,59509)</td>
<td>(2426,2727)</td>
<td>(123289,81841)</td>
</tr>
<tr>
<td>250</td>
<td>(501,10872)</td>
<td>(1020,25923)</td>
<td>(1511,61516)</td>
<td>(2076,24517)</td>
<td>(2596,8145)</td>
<td>(123456,95491)</td>
</tr>
<tr>
<td>251</td>
<td>(504,34198)</td>
<td>(1021,52191)</td>
<td>(1514,18629)</td>
<td>(2079,52447)</td>
<td>(2597,20182)</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>(508,56806)</td>
<td>(1022,35175)</td>
<td>(1515,52722)</td>
<td>(2082,57101)</td>
<td>(2602,11307)</td>
<td>0</td>
</tr>
<tr>
<td>253</td>
<td>(509,17779)</td>
<td>(1023,21986)</td>
<td>(1525,20628)</td>
<td>(2084,10317)</td>
<td>(2605,60119)</td>
<td>0</td>
</tr>
<tr>
<td>254</td>
<td>(510,45297)</td>
<td>(1027,4468)</td>
<td>(1530,38375)</td>
<td>(2086,44235)</td>
<td>(2608,38695)</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>(511,58316)</td>
<td>(1029,53908)</td>
<td>(1531,18716)</td>
<td>(2087,43862)</td>
<td>(2610,60146)</td>
<td>0</td>
</tr>
</tbody>
</table>

An Image Encryption Scheme Based on Elliptic Curve and a Novel Mapping Method
Ali Soleymani, Md Jan Nordin, Azadeh Noori Hoshyar, Zulkarnain Md Ali, Elankovan Sundararajan
Table 2. Results of mapping pixels to points, encryption and mapping encrypted points to pixels

<table>
<thead>
<tr>
<th>Pixel Value</th>
<th>161</th>
<th>159</th>
<th>157</th>
<th>158</th>
<th>161</th>
<th>159</th>
<th>156</th>
<th>157</th>
<th>159</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapped Point</td>
<td>(318, 23904)</td>
<td>(315, 58283)</td>
<td>(312, 43431)</td>
<td>(314, 29931)</td>
<td>(840, 23727)</td>
<td>(307, 29631)</td>
<td>(832, 39441)</td>
<td>(304, 9557)</td>
<td></td>
</tr>
<tr>
<td>Encrypted Points</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td>(117616, 24017)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encrypted Pixel</td>
<td>205</td>
<td>222</td>
<td>82</td>
<td>77</td>
<td>107</td>
<td>88</td>
<td>82</td>
<td>91</td>
<td>169</td>
</tr>
</tbody>
</table>

4. Security Analysis

To determine whether a cryptosystem is secure or not, performing some analysis is inevitable. Different analysis techniques are available for various types of attack [11]. Statistical attacks are more common due to the weakness of algorithms that allow an opponent to acquire knowledge related to a plain image. Some security analyses on the proposed algorithm were implemented and analysed using MATLAB on a PC with Intel 2.3 GHz CPU, 8 GB of RAM and a 64-Bit Mac OS. Experimental analysis was done on two different images. Lena - a common grey image in BMP format and another image containing a logo.

4.1. Histogram Analysis

An image histogram is a graphic illustration of pixel distribution at each grey level. Some useful pieces of information can be extracted if it is not completely changed after encryption. For example, the facial image of different people has a similar histogram or using an image histogram, it can be determined whether it is a cartoon or not. Therefore, the nature of an image is extracted from its histogram. To prevent leakage of information from a ciphered image, the histogram of an encrypted image should be significantly different from the original image, and usually contain a uniform distribution. Figure 3 shows the histograms of the Lena image before and after encryption.

4.2. Correlation Analysis

Two adjacent pixels in a plain image are strongly correlated vertically, horizontally, and diagonally. This is the property of any ordinary image. The maximum value of correlation coefficient is 1 and the minimum is 0. A robust encrypted image (to a statistical attack) should have a correlation coefficient value of ~0. Correlation coefficient is calculated using (9) for both plain and ciphered images and the results are given in Table 3. Correlation distributions for vertical, horizontal, and diagonal adjacent pixels are plotted for plain and ciphered images, as shown in Figures 4 and 5.

\[
\begin{align*}
E(x) &= \frac{1}{N} \sum_{i=1}^{N} x_i, \\
D(x) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \\
\text{cov}(x, y) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))
\end{align*}
\]
Figure 3. (a) Plain image (b) and its histogram (c) Encrypted image (d) and its histogram

Figure 4. Scatter plots to show the correlation relation between two adjacent pixels in location of (a) Vertical, (b) Horizontal, (c) Diagonal in plain image

Figure 5. Scatter plots to show the correlation relationship between two adjacent pixels in location of (a) Vertical, (b) Horizontal, and (c) Diagonal, in ciphered image
Table 3: Correlation values for plain image and cipher image

<table>
<thead>
<tr>
<th></th>
<th>Vertical Correlation</th>
<th>Horizontal Correlation</th>
<th>Diagonal Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain Image</td>
<td>0.96871</td>
<td>0.94062</td>
<td>0.91633</td>
</tr>
<tr>
<td>Cipher Image</td>
<td>0.0038034</td>
<td>0.004971</td>
<td>0.003519</td>
</tr>
</tbody>
</table>

4.3. Entropy Analysis

According to Shannon’s information theory, entropy is a quantity that shows the optimal length of the code assigned to a pixel in an image. In image encryption analysis, entropy is a statistical scalar parameter that measures the randomness of an input image. The highest value of ~8 for entropy of a ciphered image means that it has a random texture. The entropy of an encrypted image using the proposed scheme is calculated using (10), giving the result 7.9981.

\[
Entropy = \sum_{i=0}^{n} P_i \log_2 P_i
\]  

(10)

4.4. Key Sensitivity Analysis

A secure algorithm should be completely sensitive to the secret key. This means that the encrypted image cannot be decrypted using slight changes in the secret key. Changing even one bit in decryption key and decrypting the ciphered image using this key will cause extreme changes in the result. Figure 6 illustrates the results of decryption using a correct private key and an incorrect key; changing only one bit, so that the encrypted image is resistant to brute-force attack.

![Figure 6](image1.png)

Figure 6: (a) Plain image, (b) Encrypted image, (c) Decrypted image with correct private key \(y=36548\), (d) Decrypted image with change one bit in private key \(y=36549\)

Figure 7 (a) shows another image with applied encryption and analysis for this study. This image contains a logo, and unlike the Lena image, almost 75% of this image consists of only 4 intensity values; as shown by histogram (b). As such, the image is not highly distributed with intensity values. Experiments using the proposed mapping method and encryption scheme resulted in a significantly different encrypted image and uniform histogram, as shown in Figure 8.
Table 4. Correlation values for plain image and cipher image

<table>
<thead>
<tr>
<th></th>
<th>Vertical Correlation</th>
<th>Horizontal Correlation</th>
<th>Diagonal Correlation</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain Image</td>
<td>0.7546</td>
<td>0.8055</td>
<td>0.6401</td>
<td>3.5628</td>
</tr>
<tr>
<td>Cipher Image</td>
<td>0.0023</td>
<td>0.0109</td>
<td>0.0042</td>
<td>7.8240</td>
</tr>
</tbody>
</table>

4.5. Running Time

To estimate the running time, encryption and decryption was performed based on the parameters of Curve P-131 (in Table 5) using Certicom and consumed times for mapping, encryption, and decryption. As illustrated in Table 5. Increasing the key size is time consuming. However, this could be combined with other techniques. Multiplying operations for encryption and decryption, implemented by LSB-First technique, as proposed by Yeh, a fast and parallelized algorithm for computing $kP$, where $k$ is a large integer and $P$ is a point on an elliptic curve.

Table 5. Certicom Parameters for ECC p-131

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>041CB121CE2B31F608A76FC8F23D73CB66</td>
</tr>
<tr>
<td>b</td>
<td>02F74F717E8DEC90991E5EA9B2FF03DA58</td>
</tr>
<tr>
<td>p</td>
<td>048E1D43F293469E33194C43186B3ABC0B</td>
</tr>
<tr>
<td>Gx</td>
<td>03DF84A96B5688EF574FA91A32E197198A</td>
</tr>
<tr>
<td>Gy</td>
<td>014721161917A44FB7B4626F36F0942E71</td>
</tr>
</tbody>
</table>

Table 6. Estimated running times
### 5. Conclusion

Elliptic Curve Cryptography is an almost new public key cryptosystem, and provides equivalent security with a smaller key size, low mathematical complexity, and is more computationally efficient than RSA. High-speed encryption and saving bandwidth, makes ECC an acceptable option for high data rate and real time applications, such as image and multimedia encryption. In this study, a new mapping method was introduced to convert a pixel’s value to a point on an affine elliptic curve over a finite field $GF(p)$ using a map table. This mapping technique is fast, has low complexity and computation, is easy to implement, and has similar performance on images with low diversity in grey levels. Security analyses on encrypted images proved the strength of the proposed scheme and its robustness to statistical attacks. For future work, this method could be combined with a chaos map to achieve hybrid cryptography to more diffusion and confusion, with respect to running time efficiency and expanding key space.

### 6. Acknowledgement

This work was supported by The Ministry of Higher Education Malaysia under research grant of FRGS/1/2012/SG05/UKM/02/1.

### 7. References