Some elliptic curve based crypto-compression schemes for medical images

Cidjeu D.D.¹, Tieudjo D.²

¹PhD Student, Department of Mathematics and Computer Science, The University of Ngaoundere, Ngaoundere, Cameroon
²Associate Professor, Department of Mathematics and Computer Science, The University of Ngaoundere, Ngaoundere, Cameroon

Abstract: In this paper, we propose two lossless crypto-compression schemes based on Elliptic Curve (EC) for the security of medical images. Pixels values are computed as elements of a finite field $F_q$. For compression, these elements are grouped in blocks in an appropriate manner. After transforming these blocks into points of an elliptic curve, two EC-based encryption schemes are applied. We obtain two crypto-compression schemes which compared to some existing systems, offer better performances in terms of compression rate, image quality and execution time.

Keywords: Image, Crypto-compression schemes, Elliptic Curve Cryptography, Discrete Logarithm Problem (DLP).

I. INTRODUCTION

Elliptic Curve Cryptography (ECC) has performed remarkably well in recent years. Indeed, operations on elliptic curves are faster and more efficient for the implementation of cryptosystems [1]. In addition, for an equivalent level of security, cryptosystems based on elliptic curves require smaller key sizes than those based on Number Theory. For example, a security level of 80 bits is reached with 160-bit keys with ECRSA against 1024 bits for RSA [5]. This makes EC an appropriate platform for developing cryptographic systems. Many cryptosystems based on EC have been developed. Among them we have encryption schemes such as Elliptic Curve Integrated Encryption Scheme (ECIES) [5][11], ElGamal’s analog [9] and Massey-Omura’s analog [9]; key agreement protocols like Elliptic Curve Diffie-Hellman (ECDH) key exchange [5]; digital signatures protocols like Elliptic Curve Digital Signature Algorithm (ECDSA) and Edwards curve Digital Signature Algorithm (EdDSA) [5]; pseudo-random generators [5], etc.

A crypto-compression scheme is a hybrid process that reduces the size of an image while ensuring its confidentiality. There are many crypto-compression schemes in the literature. In 2006, Puech et al. proposed a crypto-compression scheme based on AES encryption and DCT compression [15]. But the DCT compression brings remarkable losses on the image from a certain level of compression rate [4]. So, Benabellah and al. proposed in 2006 and 2007 two crypto-compression schemes based on Discrete Wavelet Transform (DWT). The first scheme uses DES [2] while the second uses AES [3]. These two schemes use encryption based on Number Theory (AES and DES), which is said will be made vulnerable by the quantum computer. Furthermore, Number Theory based cryptosystems generally need big key size to be secured, which is a problem in practice. The execution time needed in decompression is also a limit in schemes based on DWT. Another platform, chaos-based cryptography, has also been used for image crypto-compression ([8], [11]). However, although it is fast, it lacks robustness and good security, as mentioned in [18]. Crypto-compression schemes are evaluated by the image quality after reconstitution, the compression rate, the execution time and the security level. Having a fast cryptosystem which guarantees more security and offers better quality and better compression rate is still a challenge and a real need, precisely for medical images, where good security, higher image quality and small execution time are all required. Moreover, as we know, there is no image crypto-compression scheme based on EC.

In this paper, we propose two lossless crypto-compression schemes based on EC. Given an image $I$, a finite field $F_q$ and an elliptic curve $E$ over $F_q$. For compression, we group $I$ in 8x8 pixels blocks. For each row of any block $B$, we compute an element of $F_q$ which represents that row. The next step consists of transforming the obtained elements of $F_q$ into points of the elliptic curve $E$. Then, two EC-based encryption schemes (Massey-Omura and ElGamal) are applied to encrypt the points of $E$ representing $I$. This provides two crypto-compression schemes. Applied to some medical images, the results obtained are more efficient in terms of image quality, compression rate (weight) and speed (execution time), compared to some existing in the literature.
II. PRELIMINARIES

A. EC based encryption

Encryption with elliptic curves has been presented in [9]. We describe below two encryption schemes based on EC. In these cryptosystems, a message is seen as a point of an elliptic curve.

A.1. Massey-Omura’s encryption scheme

Given an information represented by a point \( P_m \) of a given elliptic curve, the encryption of \( P_m \) can be performed with the scheme described below.

This scheme is built on the difficulty to solve the Discrete Logarithm Problem [9]. If one can solve this problem, then he can compute \( e_A \) and \( e_B \) and then recover the message.

A.2. ElGamal’s encryption scheme

A message seen as point of an elliptic curve can also be encrypted by the ElGamal’s analog presented here.

For these cryptosystems to be applied on images, a transformation of image into points of elliptic curve is needed. This transformation is described in the next section.

B. EC transformation of images

Let \( I \) be an image. In this section, a process to transform \( I \) into a sequence of points of an elliptic curve is presented. Each pixel will be seen as a point of a given elliptic curve. Any pixel \( P_m \) can then be encrypted by the encryption schemes presented previously.

B.1. Transforming a character to a point of an elliptic curve

For ECC to be applied directly on any data, these data have to be transformed into points on elliptic curve. In [9], Koblitz described a process to transform a character into a point of an elliptic curve. A character is seen as an integer \( m \), such that \( 0 \leq m \leq M \in N \). For example, letters (A to Z) are numbers between 0 and 25. For a given character \( m \),
Algorithm 1 below computes a pair \((x, y)\) which is a point of an elliptic curve, representing the given character. Assume that we have a finite field \(F_q\) such that \(q\) is on the form \(q = p^r, p\) prime, \(r > 0;\) and \(q = Mk + 1\), where \(k\) is generally set to 30 or 50. Given the curve \(y^2 = x^3 + ax + b\) over the finite field \(F_q\) and given a character represented by an integer \(m\).

Compute for each \(j = 1, \ldots, k\),

\[ mk + j \]

Let \(x\) be the corresponding element of \(mk + j\) in \(F_q\).

For such \(x\), we compute \(y^2 = f(x) = x^3 + ax + b\) and find a square-root for \(f(x)\). If there exists a \(y\) such that \(y^2 = f(x)\), the point of the elliptic curve representing \(m\) is \(P_m = (x, y)\). If there is no square-root for \(f(x)\) for the current \(j\), we jump to the next \(j\). With \(k = 30\) or \(k = 50\) the algorithm always returns a good result \([9]\). This process is detailed in Algorithm 1.

From Algorithm 1, given a point \((x, y)\) representing a character, this initial character \(m\) can be recovered by computing \(\left\lfloor \frac{\bar{x} - 1}{k} \right\rfloor\), where \([y]\) represents the integer part of \(y\) and \(\bar{x}\) is the integer which corresponds to \(x\) in the equivalence between the integers and the elements of \(F_q\).

B.2. Transforming image to points of an elliptic curve

Algorithm 1 above describes how to transform a character into a point of an elliptic curve. An image is a sequence of integer values between 0 and 255. We present below how to generate the sequence of points of an elliptic curve representing a given image. Algorithm 2 shows how the integer values between 0 and 255 are represented as points of a given elliptic curve.

Using algorithm 2, it is shown below (Algorithm 3) how to obtain a sequence of points of an elliptic curve, representing a given image.
By using Algorithm 3, an image can be viewed as points of an elliptic curve, fixed in advance.

Consequently, what is done on images can be done on points of an elliptic curve and what is done on elliptic curves can be applied on images.

C. Image crypto-compression

In medical applications, the size of scanned images is usually very big. Compression is therefore used to improve storage capacity and reduce the transmission time of medical images across networks. Some known compression approaches are the Discrete Cosine Transform (DCT) based compression and the Discrete Wavelet Transform (DWT) based compression. So, to secure images, encryption is generally joined to compression to obtain a hybrid process called crypto-compression, which produces a compressed and encrypted image, as described in Figure 3 below.

The scheme takes an image, compresses it and then encrypts certain compression parameters so that the crypto-compressed image is not decipherable. When the encrypted parameters are decrypted and decompression applied, we obtain the reconstituted image [4].

III. PROPOSED IMAGE CRYPTO-COMPRESSION SCHEMES

A. Image crypto-compression

A.1 General description

Let I be an \( L \times l \) image. Assume \( I \) is a colour image. Then, \( I \) is initially represented by a \( L \times (l \times 3) \) byte matrix. The proposed crypto-compression schemes have three main phases:

1. Block compression and decompression presented in section III.A.1.1,
2. EC transformation and inverse transformation (section III.A.1.2),
3. Encryption and Decryption (section III.A.1.3).
A.1.1. Block compression/decompression

Let \( I \) be a colour image of \( L \) lines and \( l \) columns. Each pixel is coded by 3 values from 0 to 255. Let \( M \) be the \( L \times (l \times 3) \) bytes matrix representing \( I \). For all \( i \) and \( j \) such that \( 0 \leq i < L \) and \( 0 \leq j < l \times 3 \), we have \( 0 \leq M(i, j) = x_{i,j} \leq 255 \). The block compression reduces every \( 8 \times (8 \times 3) \) pixels block of the image to a \( 8 \times 3 \) block of element of \( F_p \) as described below.

Let divide \( M \) into \( 8 \times 8 \) blocks of pixels. Let \( B = (x_{i,j})_{0 \leq i < 8, 0 \leq j < 8 \times 3} \) be the \( 8 \times (8 \times 3) \) bytes matrix representing a \( 8 \times 8 \) pixels block of \( M \).

For every row \( i \) in \( B \), compute

\[
elt_i = (elt_{i,1}; elt_{i,2}; elt_{i,3}),
\]

where

\[
elt_{i,1} = \sum_{j=0}^{24} x_{i,j} \times (10^3)^{8-j-1}
\]

We obtain

\[
etl_{i,1} = \sum_{j=0}^{7} x_{i,j} \times (10^3)^{7-j}
\]

\[
etl_{i,2} = \sum_{j=8}^{15} x_{i,j} \times (10^3)^{15-j}
\]

and

\[
etl_{i,3} = \sum_{j=16}^{33} x_{i,j} \times (10^3)^{31-j}
\]

The values of \( B \) are computed as elements of a finite field \( F_p \). Let \( F_p \) be a finite field such that \( p \) is prime and \( p \geq (12^{24}) \times k + 1 \), where \( k \) is set to 50 or 30; let \( E \) be an elliptic curve over \( F_p \). Since the \( x_{i,j} \) are from 0 et 255 and \( p \geq (10^{24}) \times k + 1 \), then \( elt_i = (elt_{i,1}; elt_{i,2}; elt_{i,3}) \in F_p^3 \). So, \( elt_i \) is a triple of elements of \( F_p \) representing line \( i \) of \( B \). By this operation, we move from a matrix \( B \) of size \( 8 \times 3 \times 8 \) to a matrix \( B' \) of size \( 8 \times 3 \). The obtained block \( B' \) are merged to form a matrix \( M' \) of size \( L \times \left[ \frac{l}{8} \right] \), where \( \left[ x \right] \) is the first integer greater or equal to \( x \). The block compression is performed by Algorithm 4.

---

**Algorithm 4 Block compression**

**Require:** a \( L \times (l \times 3) \) bytes matrix \( M \) representing a \( L \times l \) image \( I \), \( F_p \)

**Ensure:** a \( L \times \left[ \frac{l}{8} \times 3 \right] \) matrix \( M' \) of elements of \( F_p \), representing the compressed image

1. Cut \( M \) in \( 8 \times (8 \times 3) \) blocks
2. For every block \( B = (x_{i,j})_{0 \leq i < 8, 0 \leq j < 8 \times 3} \) and every row \( i \) of \( B \)
3. Compute \( elt_{i,1} = \sum_{j=0}^{7} x_{i,j} \times (10^3)^{7-j} \)
4. Compute \( elt_{i,2} = \sum_{j=8}^{15} x_{i,j} \times (10^3)^{15-j} \)
5. Compute \( elt_{i,3} = \sum_{j=16}^{33} x_{i,j} \times (10^3)^{31-j} \)
6. \( B'[i] = (elt_{i,1}; elt_{i,2}; elt_{i,3}) \)
7. Add \( B' \) to \( M' \)
8. Return \( M' \)

By Algorithm 4, a compression ratio of \( 1 - \frac{1}{8} = 87.5\% \) is performed.
A.1.2. Elliptic Curve Transformation/Elliptic Curve Inverse Transformation

At this step, we are transforming a matrix $M'$ of elements of a finite field $F_p$, representing a compressed image, into a matrix of points of an elliptic curve. Assume that we have a finite field $F_p$ such that $p$ is prime and $p > (1025) \times k + 1$, where $k$ is generally set to 30 or 50. Given the curve $y^2 = x^3 + ax + b$ over the finite field $F_p$ and given an image $I$ represented by a matrix $M'$ of elements of $F_p$. For every value $m$ in $M'$, Algorithm 1 is used to compute the point of $E$ representing $m$. Every point of $M'$, which is a block of 8 pixels, is then replaced by a point of $E$. This process is described in Algorithm 5 below.

Algorithm 5 Transform a compressed image into points of an EC

| Requires: | $F_p$, a matrix $M'$ of elements of $F_p$ representing a compressed, $k$, $a$, $b$ |
| Ensure: | a matrix ECM of couple $(x,y) \in F_p \times F_p$ representing the compressed image |
| 1. | For each value $m$ of $M'$ |
| 2. | Compute $P_m = \text{CharEC}(m, F_p, k, a, b)$ |
| 3. | Return ECM |

A.1.3. Encryption/Decryption

The encryption schemes used in this step are those presented in sections II.A.1 and II.A.2. The application of these two encryption schemes on compressed images seen as points of an elliptic curve leads us to the two crypto-compression scheme presented in the next sections.

B. The Massey-Omura type scheme

Here, the points representing the compressed image are encrypted by the Massey-Omura encryption scheme.

C. The ElGamal type scheme

Here, the ElGamal encryption scheme is used to encrypt the points representing the compressed image.
IV. IMPLEMENTATION, RESULTS AND DISCUSSION

In this section, the Massey-Omura and ElGamal based schemes are applied on the image "lena". We illustrate the implementation. From the recovered images after implementation, we compute the image quality and the compression rate. The results are compared to some existing crypto-compression schemes: JPEG-AES, FMT-AES and JPEG-LFSR.

A. Illustrations

Let $I$ be the image presented in figure 6.

![Sample image "lena"](image)

Let $B$ be the $8 \times (8 \times 3)$ pixels block below, extracted from $I$ (Figure 7).

![An 8 × (8 × 3) pixels bloc extracted from I](image)

The elliptic curve defined by

$$y^2 = x^3 + 860770308609211603740389 \times x + 3944753409163852864976165 \text{ over the finite field of size } p = 9194940935662755805691087$$

has been generated.

The block compression of $B$ gives $B'$, the $8 \times 3$ matrix presented in figure 8. The elements of this matrix are all in $F_p$. The first element, compute from the first 8 pixel values of $B$ is

$$\text{elt}_{0,1} = 181085089194098102210114.$$
The sequence of points of $E$ representing $B'$ is shown on figure 9.

Figure 9. Sequence of points representing the bloc

Figure 9 presents points as triples $(X_P : Y_P : Z_P)$, which are their projective coordinates, according to the affine coordinates $(X_P/Z_P, Y_P/Z_P)$ if $Z_P$ is non-zero, and $O$ (the point at infinity) if $Z_P$ is zero.

For the first point for example, $P_m = (543255267582943066303421 : 2181074533967633087284250 : 1)$, if we compute

$$\frac{543255267582943066303421}{k}$$

for $k = 30$, we obtain $181085089194098102210114$ which is the first element of the compressed block. So, the inverse transformation is assured.

This matrix representing the compressed block is a $8 \times 3$ matrix of points instead of $8 \times 8 \times 3$. This constitutes a compression rate of $1/8$. When the above sequence of points is encrypted with the Massey-Omura (MO) encryption scheme described in section II.A.1, we obtain the following sequence of points (Figure 10).

Figure 10. Sequence of points representing the block encrypted with Massey-Omura encryption
When the block is encrypted with the ElGamal encryption scheme described in II.A.2, the sequence of points shown on Figure 11 is obtained.

![Sequence of points representing the block encrypted with ElGamal encryption](image1.png)

The decryption of the two encrypted blocks returns the initial values, as we can see on figure 12.

![The decrypted sequence of points](image2.png)

So, when an image is so encrypted, the original image can be reconstituted. The encrypted points can be decrypted as presented in II.A.1 and II.A.2. Then, by applying the inverse operations described in III.B and III.C, the matrix containing the initial pixel values is recovered. Finally, the reconstituted image is obtained by repeating the same process for all the blocks of the image.

### B. Discussions

Table 1 below presents a comparison between the crypto-compression schemes implemented here and some known schemes (JPEG-AES [15], FMT-AES [4] and JPEG-EC-LFSR [6]). To compute the execution time, we used the SageMath function cputime() [17]. For the compression rate, the below formula has been used [4]:
For image quality, we considered the percentage of pixels not modified during the process [4]. In Table 1, "with losses" refers to an image quality less than 100% after reconstitution, while "lossless" indicates a 100% image quality.

Table 1. Comparison with some encryption- description systems

<table>
<thead>
<tr>
<th></th>
<th>Quality</th>
<th>Time(s)</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPEG-AES</td>
<td>with losses</td>
<td>11.15</td>
<td>84%</td>
</tr>
<tr>
<td>FMT-AES</td>
<td>with losses</td>
<td>3.37</td>
<td>86%</td>
</tr>
<tr>
<td>JPEG-EC-LFSR</td>
<td>with losses</td>
<td>13.47</td>
<td>84%</td>
</tr>
<tr>
<td>EC-MO</td>
<td>lossless</td>
<td>137.95</td>
<td>87.5%</td>
</tr>
<tr>
<td>EC-ElGamal</td>
<td>lossless</td>
<td>1.09</td>
<td>87.5%</td>
</tr>
</tbody>
</table>

From table 1, it appears clearly that ElGamal based crypto-compression is better than the others. Indeed, it offers the smallest execution time for a better compression rate and an optimal quality. The great execution time taken by the Massey-Omura encryption is explained by the fact that in each step of the encryption, a multiplication of a point of \( E \) by an integer is performed instead of a simple addition for ElGamal.

V. CONCLUSION AND PERSPECTIVES

Two crypto-compression schemes based on elliptic curves have been presented. Compared to others crypto-compression schemes, the results obtained here offer a better quality since the process is lossless, and a greater compression gain. The execution time is also clearly better with the ElGamal based encryption-compression. So, processing images as points of an elliptic curve improves their crypto-compression. Others operations on images (segmentation, watermarking, etc.) may also be improved by seeing images as points on elliptic curves.

REFERENCES