IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

Some elliptic curve based crypto-compression schemes for medical images

Cidjeu D.D.[1], Tieudjo D.[2]

PhD Student, Department of Mathematics and Computer Science,
The University of Ngaoundere, Ngaoundere, Cameroon[1]

Associate Professor, Department of Mathematics and Computer Science,
The University of Ngaoundere, Ngaoundere, Cameroon[2]

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 *Fq*. 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 and *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, Benabdellah 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 *Fq* and an elliptic curve *E* over *Fq*. For compression, we group *I* in 8×8 pixels blocks. For each row of any block *B*, we compute an element of *Fq* which represents that row. The next step consists of transforming the obtained elements of *Fq* 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

DOI 10.17148/IJARCCE.2019.8520

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 5, May 2019

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.

Public parameters	E, an elliptic curve over a finite field Fq $N=\#E$
Alice Private key	$e_A \in [1, n]$, such that $\gcd(e_A, N)=1$ $d_A=e_A^{-1} \bmod N$
Bob Private key	$e_B \in [1, n]$, such that $\gcd(e_B, N)=1$ $d_B=e_B^{-1} \bmod N$
Alice want to send to Bob a message which is a point of E, P_m	
Encryption	Alice send to Bob $e_A P_m$ Bob send to Alice $e_B e_A P_m$ Alice send $P'_m = d_A e_B e_A P_m$
Decryption	Bob compute $d_B P'_m = P_m$

Figure 1. Encryption/decryption by Massey-Omura's scheme [9]

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.

Public parameters	E, an elliptic curve over a finite field Fq A base point B
Bob Public key	aB, where a is an integer
Bob Private key	The integer a
Alice want to send to Bob a message which is a point of E, P_m	
Encryption	Alice chooses an integer k and send to Bob the pair of points $(kB, P_m + kaB)$
Decryption	Bob compute $P_m + kaB - akB = P_m$

Figure 2. Encryption/decryption by ElGamal's scheme [9]

Once again, the Discrete Logarithm Problem is the base of the security of the scheme. By solving it, one can compute a and recover the message.

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 $0 \leqslant m \leqslant M \in N$. For example, letters (A to Z) are numbers between 0 and 25. For a given character m,

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

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 *Fq* such that *q* is on the form $q = p^r$, *p* prime, $r > 0$; and $q \geqslant Mk + 1$, where *k* is generally set to 30 or 50. Given the curve $y^2 = x^3 + ax + b$ over the finite field *Fq* 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 *Fq*.

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.

Algorithm 1 Transform a character into a point of an EC (CharEC)

Require: a character m, $\mathbb{F}_q$, k, a, b
Ensure: a pair (x, y) representing m, such that $y^2 = x^3 + ax + b$

1. j=1
2. while $j \leq k$
3. compute $\tilde{x} = mk + j$
4. write $\tilde{x}$ with r digits $m_{r-1}...m_1 m_0$
5. compute $x = \sum_{i=0}^{r-1} m_i g^i \in \mathbb{F}_q$, where g is a generator of $\mathbb{F}_q$
6. compute $y^2 = f(x) = x^3 + ax + b$, and find a square-root for $f(x)$
7. if there exists a y such that $y^2 = f(x)$, then return $P_m = (x, y)$ else, increment j by 1.

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

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.

Algorithm 2 Transform pixel values to points on EC (PointsEC)

Require: an elliptic curve E over $\mathbb{F}_q$
Ensure: a sequence of points $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ representing the 256 pixel values

1. points=[]
2. For each pixel value m between 0 and 255
   2.1 execute Algorithm 1 to find P_m
   2.2 add P_m to list
3. Return points

Using algorithm 2, it is shown below (Algorithm 3) how to obtain a sequence of points of an elliptic curve, representing a given image.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

Algorithm 3 Transform an image to points on EC (ImageEC)

Require: an image I

Ensure: a sequence of points $(x, y) \in \mathrm{F}_q \times \mathrm{F}_q$ representing the given image

1. Define an elliptic curve E on the form $y^2 = x^3 + ax + b$ over F_q
2. computes points=PointsEC(E)
3. imageEC=[]
4. For each pixel m in I
   4.1 add points[m] to imageEC
5. Return imageEC

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.

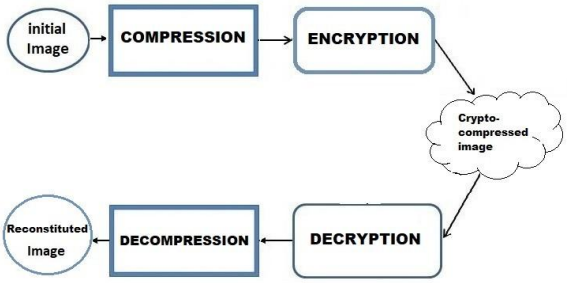

Figure 3. Classical approach to crypto-compression

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).

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

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 × (8 × 3) pixels block of the image to a 8× 3 block of element of Fp as described below.
Let divide M into 8 × 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 × 8 pixels block of M.
For every row i in B, compute

$$elt_i = (elt_{i,1}; elt_{i,2}; elt_{i,3}),$$

where

$$elt_{i,t} = \sum_{j=8(t-1)}^{8t-1} x_{i,j} \times (10^3)^{8t-j-1}$$

We obtain

$$elt_{i,1} = \sum_{j=0}^{7} x_{i,j} \times (10^3)^{7-j}$$

$$elt_{i,2} = \sum_{j=8}^{15} x_{i,j} \times (10^3)^{15-j}$$

and

$$elt_{i,3} = \sum_{j=16}^{23} x_{i,j} \times (10^3)^{23-j}$$

The values of B are computed as elements of a finite field Fp. Let Fp be a finite field such that p is prime and $p \geqslant (12^{24}) \times k + 1$, where k is set to 50 or 30; let E be an elliptic curve over Fp. Since the $x_{i,j}$ are from 0 et 255 and $p \geqslant (10^{24}) \times k + 1$, 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 Fp representing line i of B. By this operation, we move from a matrix B of size 8 × 3 × 8 to a matrix B' of size 8 × 3. The obtained block B' are merged to form a matrix M' of size $L \times \left\lceil \frac{l}{8} \right\rceil$, where $\lceil x \rceil$ 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 $L \times \left\lceil \frac{l \times 3}{8} \right\rceil$ matrix M' of elements of F_p, representing the compressed image

1. Cut M in 8x(8x3) 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 $elt_{i,3} = \sum_{j=16}^{23} x_{i,j} \times (10^3)^{23-j}$
6. $B'[i] = elt_i = (elt_{i,1}; elt_{i,2}; elt_{i,3})$
7. Add B' to M'
8. Return M'

By Algorithm 4, a compression ratio of $1 - \frac{1}{8} = 87.5\%$ is performed.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 5, May 2019

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 *Fp*, representing a compressed image, into a matrix of points of an elliptic curve. Assume that we have a finite field *Fp* such that *p* is prime and $p \geqslant (1025) \times k + 1$, where *k* is generally set to 30 or 50. Given the curve $y^2 = x^3 + ax + b$ over the finite field *Fp* and given an image *I* represented by a matrix *M'* of elements of *Fp*. 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

Require: F_p, a matrix M' of elements of F_p representing I 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 $P_m = CharEC(m, F_p, k, a, b)$ and put P_m at the place of m in ECM
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.

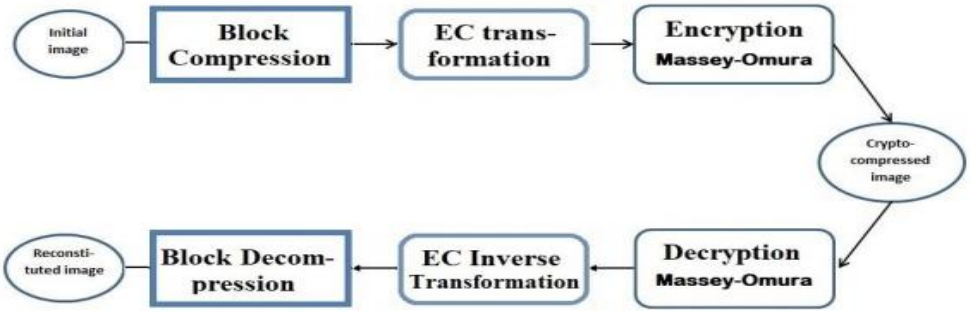

Figure 4. Crypto-compression scheme based on Massey-Omura encryption

C. *The ElGamal type scheme*

Here, the ElGamal encryption scheme is used to encrypt the points representing the compressed image.

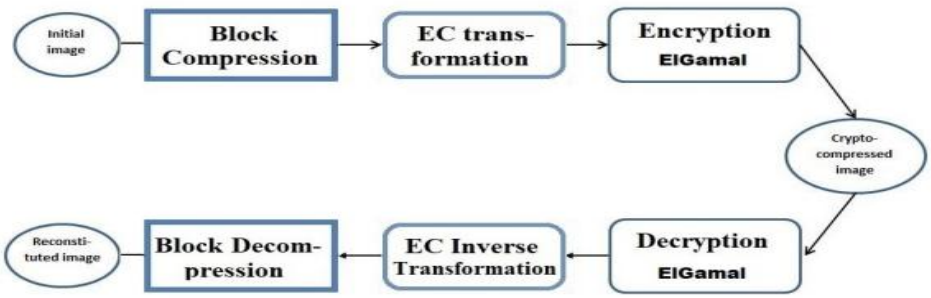

Figure 5. Crypto-compression scheme based on ElGamal encryption

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

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.

Figure 6. Sample image ”lena”

Let *B* be the 8 × (8 × 3) pixels block below, extracted from *I* (Figure 7).

```
[181  85  89 194  98 102 210 114 118 212 116 120 208 109 114 205 105 113 204 104 112  201 101 109]
[196  97 102 210 111 116 214 115 120 210 111 116 207 108 113 206 106 114 198 101 108 190  93 100]
[212 113 118 214 115 120 206 105 111 204 103 109 209 108 114 202 102 110 193  96 103  187  90  97]
[215 114 120 210 109 115 205 101 108 204 100 107 205 104 110 191  91  99 190  93 100  191  94 101]
[209 108 114 205 104 110 209 105 112 206 102 109 197  96 102 180  83  90 188  91  98 197 100 107]
[215 114 120 205 104 110 196  95 101 195  94 100 197  96 102 202 106  110 210 114  118 195  99 103]
[211 112 117 203 102 108 197  96 102 197  96 102 198  99 104 200 104 108 207  111 115  197  98 103]
[206 107 112 199  99 107 196  96 104 199  98 106 202 101 109 201  102 107 201  102 107 198  99 104]
```

Figure 7. An 8 × (8 × 3) pixels bloc extracted from I

The elliptic curve defined by

$$y^2 = x^3 + 860770308606921160374038 9 * x + 394475340916385286497616 5 \text{ over the finite field of size}$$
$$p = 919494093566275580569108 7$$

has been generated.

The block compression of *B* gives *B′*, the 8×3 matrix presented in figure 8. The elements of this matrix are all in *Fp*. The first element, compute from the first 8 pixel values of B is

$$\text{elt}_{0,1} = 181085089194098102210114.$$

```
[181085089194098102210114 118212116120208109114205 105113204104112201101109]
[196097102210111116214115 120210111116207108113206 106114198101108190093100]
[212113118214115120206105 111204103109209108114202 102110193096103187090097]
[215114120210109115205101 108204100107205104110191  910991900931001910941011]
[209108114205104110209105 112206102109197096102180  830901880910981971001071]
[215114120205104110196095 101195094100197096102202 106110210114118195099103]
[211121172031021081970 96 102197096102198099104200 104108207111115197098103]
[206107112199099107196096 104199098106202101109201 102107201102107198099104]
```

Figure 8. The 8×3 matrix obtained after row compression

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

The sequence of points of *E* representing *B'* is shown on figure 9.

```
[[(5432552675822943066303421 : 2181074533967633087284250:1),
(3546363483606243273426150 : 2984377973641591015086 38 : 1),
(3153396123123366033033271 : 7764119264625625265469 7 : 1)],
[(5882913066303333486423450 : 3544334886326394563167469: 1),
(3606303333486213243396180 : 8085849773696732173661 37 : 1),
(3183425943033245702793000 : 1634707051192673592790909 :1)],
[(6363393546423453606183150 : 1403034160784585299936 41 : 1),
(3336123093276273243426064 : 2573772106825497898723330 : 1),
(3063305792883095612702910 : 8635333354791570047994 54 : 1)],
[(6453423606303273456153030 : 1145690416667479881785989 :1),
(3246123003216153123305730 : 3337073052674808546802738 : 1),
(2732975702793005732823035 : 1500474941823610023988 24 : 1)],
[(6273243426153123306273152 : 3115488627365979385578692 :1),
(3366183063275912883065400 : 2931432536768990021556017 : 1),
(2492705642732945913003210 : 1951224737429512321435284 :1)],
[(6453423606153123305882850 : 3671227255711398080608933 :1),
(3035852823005912883066060 : 2071970719391409736 8399 : 1),
(3183306303423545852973090 : 2363483130193579751119724 :1)],
[(6333363516093063245912880 : 1619542044935651323570932 :1),
(3065912883065942973126000 : 1782935150272687791129 48 : 1),
(3123246213333455912943090 : 4193464944833841470814307 :1)],
[(6183213365972973215882881 : 4940381319592056615195 72 : 1),
(3125972943186063033276032 : 4537586987392051618798923 : 1),
(3063216033063215942973121 : 3624192400884451079634437 :1)]]
```

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 = (5432552675822943066303421 : 2181074533967633087284250 : 1), if we compute

$$\frac{5432552675822943066303421 - 1}{k}$$

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 × 3 matrix of points instead of 8 × 8 × 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).

```
[[(5535246862812326655089602 : 8717083436257246816892187 : 1),
(4561259167576562245329541 : 2308733560152017645040690 : 1),
(3734489896832649918061684 : 4959299626945734066490 16 : 1)],
[(4187703346791914482371631 : 6143598798063330962999584 : 1),
(3191567661046227851500556 : 8687723643564988006084235 : 1),
(5429979856232743214149465 : 6574904858087717349494 83 : 1)],
[(2420781399291101473657124 : 8872861217371428284569783 : 1),
(5239895342389894224459681 : 6591154680635187776487 66 : 1),
(8660518482063398766953521 : 5049168417416406192736053 : 1)],
[(4067620835158295892539 63 : 2791610768992732318091233 : 1),
(7024556238974191243629 98 : 5098140938006150127092144 : 1),
(2941316125115508187424413 : 8345859178497910518175681 : 1)],
[(8032314453562085113271587 : 6506550916308722765992292 : 1),
(1411001627731723011287125 : 4335529103196365725336766 : 1),
(6442939077229760487160 44 : 1633319693519831554533860 : 1)],
[(5631915149046829586077047 : 7744917892737225662439495 : 1),
(3097323106866455199901524 : 8946844070150931489774 63 : 1),
(8091707203894113656741418 : 5190559008844868198368044 : 1)],
[(3211654084714450923752420 : 4534455026920318177654426 : 1),
(7466821325504894886545823 : 7470648326107203624423748 : 1),
(2062913160859684080704253 : 8354918965753570210130514 : 1)],
[(3691291012727697623931641 : 5506261578087758207533508 : 1),
(6209054904713607440716764 : 5596311114169497627122 62 : 1),
(2724638908517030437212322 : 3532314937466979378909092 : 1)]]
```

Figure 10. Sequence of points representing the block encrypted with Massey-Omura encryption

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 5, May 2019

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.

```
[[(2140604642708543924265 16 : 8525627858472746584503 17 : 1),
(2862012023692007466625438 : 2320765898800632265907 51 : 1),
(9108131462754502592040240 : 8312306081162339731390307 : 1)],
[(1985984868415909361384719 : 5830766085361424523447976 : 1),
(3025371855175440449525 00 : 2578985092809310205224457 : 1),
(8114472708360581098857451 : 4098482905088903880067365 : 1)],
[(2135662420723279877380716 : 7861932038320269287726922 : 1),
(4229998466422814365496 28 : 6231533479965521579000623 : 1),
(3845433008519229927268286 : 1752954519719484136523616 : 1)],
[(7309649822429777025498 22 : 4346598127426425501746538 : 1),
(8395426931148273714837098 : 8233658225475306535030750 : 1),
(6938087571552847815024084 : 7685853814046644288993814 : 1)],
[(4390366661324957784710662 : 2378217254457109855097816 : 1),
(2087085422695338853387540 : 16473031781733695555816 : 1),
(3628495081788474939866025 : 8711559519853391190289 24 : 1)],
[(4194111991746713430187833 : 3068852350088253732866910 : 1),
(4833113760022328561755870 : 4838587234134247241854201 : 1),
(2951328810811337459421 14 : 5881804348035801022395518 : 1)],
[(4041044955985864962471093 : 3622340234424706755762192 : 1),
(2061891433420139067916877 : 9592628527930197058537 90 : 1),
(7946124439277627736842 77 : 5591055188785520929230715 : 1)],
[(6914513020402061599250061 : 3610499576760741696941475 : 1),
(4619976938623673782388880 : 1790207512146762294638344 : 1),
(4954023561976283752588974 : 5943106358029598722096095 : 1)]]
```

Figure 11. Sequence of points representing the block encrypted with ElGamal encryption

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

```
[[(5432552675822943066303421 : 2181074533967633087284250 : 1),
(3546363483606243273426150 : 2984377973641591015086 38 : 1),
(3153396123123366033033271 : 7764119264625625265469 7 : 1)],
[(5882913066303333486423450 : 3544334886326394563167469 : 1),
(3606303333486213243396180 : 8085849773696732173661 37 : 1),
(3183425943033245702793000 : 1634707051192673592790909 : 1)],
[(6363393546423453606183150 : 1403034160784585299936 41 : 1),
(3336123093276273243426064 : 2573772106825497898723330 : 1),
(3063305792883095612702910 : 8635333354791570047994 54 : 1)],
[(6453423606303273456153030 : 1145690416667479881785989 : 1),
(3246123003216153123305730 : 3337073052674808546802738 : 1),
(2732975702793005732823035 : 1500474941823610023988 24 : 1)],
[(6273243426153123306273152 : 3115488627365979385578692 : 1),
(3366183063275912883065400 : 2931432536768990021556017 : 1),
(2492705642732945913003210 : 1951224737429512321435284 : 1)],
[(6453423606153123305882850 : 3671227255711398080608933 : 1),
(3035852823005912883066060 : 2071970719391409736839 9 : 1),
(3183306303423545852973090 : 2363483130193579751119724 : 1)],
[(6333363516093063245912880 : 1619542044935651323570932 : 1),
(3065912883065942973126000 : 1782935150272687791129 48 : 1),
(3123246213333455912943090 : 4193464944833841470814307 : 1)],
[(6183213365972973215882881 : 4940381319592056615195 72 : 1),
(3125972943186063033276032 : 4537586987392051618798923 : 1),
(3063216033063215942973121 : 3624192400884451079634437 : 1)]]
```

Figure 12. The decrypted sequence of points

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]:

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

$$t = 1 - \frac{Size_{final}}{Siza_{initial}}$$

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

"lena"	Quality	Time(s)	Rate
Original image	/	/	/
JPEG-AES	with losses	11.15	84%
FMT-AES	with losses	3.37	86%
JPEG-EC-LFSR	with losses	13.47	84%
EC-MO	lossless	137.95	87.5%
EC-ElGamal	lossless	1.09	87.5%

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

[1]. Bauer B., Donat-Bouillud P. and Durand V., *Courbes elliptiques et cryptographie*, Cours ENS Rennes, 2011.
[2]. Benabdellah M, Majid H.M., Zahid N., Regragui F. and Bouyakhf E.H., *Encryption- Compression of still images using the FMT transformation and the DES algorithm*, International Journal of Computer Sciences and Telecommunications, No. 4, 2006.
[3]. Benabdellah M, Majid H.M., Zahid N., Re- gragui F. and Bouyakhf E.H., *Encryption- compression of images based on FMT and AES algorithm*, International Journal Applied Mathematical Sciences, Vol. 1, 2007.
[4]. Bebabdellah M., *Outils de compression et de crypto-compression: applications aux images fixes et vidéo*, Thèse de doctorat, Université Mohammed V-Agdal, 2007.
[5]. Bos J. W., Halderman J. A., Heninger N.,Moore J., Naehrig M.and Wustrow E., *Elliptic Curve Cryptography in Practice*, IACR, 2013.
[6]. Cidjeu D.D. and Tieudjo D., *Image crypto-compression based on Elliptic Curves and Linear Feedback Shift Register (LFSR)*, International Journal of Computer Science and Telecommunications, Vol. 9, Issue 7, pp 7 – 13, 2018.
[7]. Ftérich S. and Ben Amar C., *Crypto- compression d¨images fixes par la méthode de Quadtree optimisée et AES*, CORESA, 2004.
[8]. Jalel H., Mohamed A., Ben F., Mounir S. and Abdennaceur K., *Crypto-compression of images based on chaos, 6th International Con- ference on Sciences of Electronics*, Technologies of Information and Telecommunications (SETIT), Sousse, pp. 344-350, 2012.
[9]. Koblitz N., *A Course in Number Theory and Cryptography - 2nd ed.*, Springer-Verlag, 1994.
[10]. Martinez V.G., Encinas L.H. and Avila C.S., *A Survey of the Elliptic Curve Int grated Encryption Scheme*, Journal of Com- puter Science and Engeneering, Vol 2, 2010.
[11]. Masmoudi A., Puech W., *Lossless chaos- based crypto-compression scheme for image protection*, ITE Image Processing, vol 8, 2014.
[12]. Meraoubi H., Brahimi Z., Ait Saadi K. et Zemouri A. , *Un système de crypto- compression des images médicales basé sur la DCT 2×2-IDS et l'AES*, 1st ISESC, 2005.
[13]. Puech W. and Rodrigues J.M., *A new crypto-watermarking method for medical images safe transfer*, 12th European Signal Pro- cessing Conference, Vienna, pp. 1481-1484, 2004.
[14]. Puech W. et Rodrigues J.M., *Crypto- Compression of medical images by selective encryption of DCT*, In EUSIPCO'05, Antalya, 2005.
[15]. Puech W., Rodrigues J.M. et Develay- Morice J.E., *Transfert sécurisé d'images médicales par codage conjoint : cryptage sélectif par AES en mode par flot et compression JPEG*, Traitement du signal (TS), numéro spécial Traitement du signal appliqué à la cancérologie, vol. 23, 5, 2006.
[16]. Puech W. et Coatrieux G., *Codage hy- bride - cryptage-marquage-compression pour la sécurisation de l'information médicale, In Compression des images et des signaux médicaux (chap. 10)*, Lavoisier, IC2, pp.269-298, 2007.
[17]. Stein W., Sage for Power Users, Computer Aided Mathematics, 2012.
[18]. Xingyuan W., Xiaojuan W., Jianfeng Z. and Zhenfeng Z., *Chaotic encryption algorithm based on alternant of stream cipher and block cipher, Nonlinear Dynamics*, vol 63, Issue 4, pp 587-597, 2011.