Images as sequence of points of an elliptic curve

Cidjeu Djeuthie Diderot*, Tieudjo Daniel**

Department of Mathematics and Computer Science
The University of Ngaoundere
PO Box 455 Ngaoundere
Cameroon
*cidjeu@gmail.com, **tieudjo@yahoo.com

RÉSUMÉ. Plusieurs transformations sont effectuées sur les images (Transformée en Cosinus Discrète, Transformée en Ondelettes Discrete, etc.) pour faciliter leur traitement et garantir leur sécurité. Cependant, ces transformations ne donnent pas toujours meilleure satisfaction lorsqu'on applique les algorithmes cryptographiques (crypto-compression par exemple) sur ces images. La cryptographie basée sur les courbes elliptiques offre de nos jours des performances remarquables. En vue d'appliquer la cryptographie basée sur les courbes elliptiques aux images, il est nécessaire de transformer ces images en séquences de points sur des courbes elliptiques. Dans ce papier, nous décrivons une méthode de transformation d'une image en séquence de points d'une courbe elliptique.

ABSTRACT. Several transformations are performed on images (Discrete Cosine Transform, Discrete Wavelet Transform, etc.) to facilitate their processing and ensure their security. However, these transformations do not always offer better satisfaction when applying cryptographic algorithms such as crypto-compression. Nowadays, Elliptic Curve Cryptography (ECC) have demonstrated remarkable performances in cryptography. To apply ECC on images, it is necessary to transform these images into sequences of points of elliptic curves. In this paper, we describe a method to transform an image into sequence of points of an elliptic curve.

MOTS-CLÉS : Crypto-compression, courbe elliptique, image

KEYWORDS : Crypto-compression; elliptic curve; image.

1. Introduction

Images are digital data transferable through public channels, thus need to be secured. The main solutions to secure images are watermarking and encryption. While watermarking enables image authentication, encryption ensures its confidentiality [1]. Several works have been done on the security of images ([2, 3, 4, 5, 6]). Generally, before performing encryption algorithms to images, various image transformations (Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT),... are used to facilitate image processing. Then quantification (compression) can be applied to reduce the size of the image. So, to practically secure images, compression is joined to encryption to obtain a hybrid process called crypto-compression. Compression algorithms aim to reduce the size of images, and such facilitate the transfer, storage, encryption, etc. Some crypto-compression algorithms can be found in [4, 5, 7, 8, 9, 10, 11]. Compression algorithms consider image as bytes matrix (table of digits) and most encryption algorithms used for image security are based on Number Theory (RSA, AES, DES,). Consequently, crypto-compression systems on images require keys of very large size, which is a problem in practice. Moreover, cryptosystems based on Number Theory are exposed to quantum attacks. Elliptic Curves Cryptography (ECC) presents several advantages compared to Number Theory based Cryptography : it offers smaller key sizes (160-bit for 1024-bit with RSA for example), has faster and more efficient implementation issues [12], etc. ECC has not yet been applied to image security. For image to be secured by ECC, it has to be seen as points of an elliptic curve.

In this paper, we propose an algorithm to transform an image into sequence of points of an elliptic curve. Following Koblitzs algorithm presented in [13], which transforms a character (letter, digit, etc) to a point of an elliptic curve, we describe how a pixel value can be represented as point on an elliptic curve. Finally, we describe how a whole image can be represented and seen as a sequence of points of a elliptic curve.

2. Preliminaries

Let $\mathbb{K}$ be a field of characteristic different from 2 and 3. An elliptic curve E over $\mathbb{K}$ is the set of points

$$E = \{O\} \cup \{(x, y) \in \mathbb{K} \times \mathbb{K}, y^2 = x^3 + ax + b\}$$

where O is a specific point called the point at infinity.
For example, Figure 1 below shows the elliptic curve $y^2 = x^3 - 2x - 2$ on the real field $\mathbb{R}$.

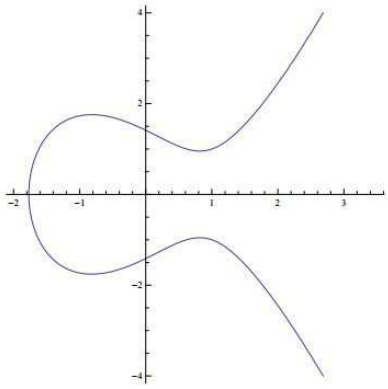

Figure 1. *Graph of the elliptic curve* $y^2 = x^3 - 2x - 2$ *on the real field* $\mathbb{R}$

More on elliptic curves and Elliptic Curve Cryptography (ECC) can be found in [12, 13, 14].

Below we will consider the field $\mathbb{K}$ to be the finite field $\mathbb{F}_q$, where $q = p^r$ for p prime and integer $r > 0$.

3. Transforming a character to a point of an elliptic curve

In [13], Koblitz described a process to transform a character to a point of an elliptic curve. A character c is represented as an integer m, such that $0 \leq m < M \in N$. For example, letters A to Z can be considered as integers between 0 and 25 ($M = 26$). For a given character c represented by an integer 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 $\mathbb{F}_q$ such that q is on the form $q = p^r$, p prime, $r > 0$; and $q \geq Mk + 1$, where k is generally set to 30 or 50. Given the curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_q$ and given a character represented by an integer m.

Compute for each $j = 1, ...k$,

$$mk + j$$

Let x be the corresponding element of $mk + j$ in $\mathbb{F}_q$.

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 $k = 50$ the algorithm always return a good result [13]. 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 $\left\lfloor \frac{(\tilde{x}-1)}{k} \right\rfloor$, 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 $\mathbb{F}_q$.

Algorithm 1 Transform a character to a point of an EC

Require: a character m, $\mathbb{F}_q$, k, a, b
Ensure: a pair $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ representing m

1. j=1
2. while $j \leq k$
   3. compute $\tilde{x} = mk + j$
   4. write $\tilde{x}$ with r digits $m_{r-1}...m_1m_0$
   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.

4. Transforming image to points on elliptic curve

Pixel values are digits between 0 and 255. So, Algorithm 1 can be used to compute the point of the elliptic curve, representing each pixel value. Algorithm 2 shows how to compute the 256 points of an elliptic curve (E) representing the 256 possible pixel values. An illustration is presented below (Figure 2).

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

With this algorithm, for a given image, the sequence of points representing that image on the elliptic curve can be produced as presented in Algorithm 3.

At the end of this algorithm, of a given image I can be encrypted or processed as points of an elliptic curve.

When an image is so transformed to points of an elliptic curve, the original image can be recovered. The 256 points representing the 256 pixels values are also known as computed by Algorithm 2. Given a point representing a pixel value in an image, the index (rank) of that point in the list of 256 points computed in Algorithm 2 is the corresponding pixel value of the given point.

Finally, the original image can be reconstituted by substitution of each point by the corresponding pixel value.

Algorithm 3 Transform an image to points on EC

Require: an image I

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

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

5. Illustration

Figure 2 presents a list of the 256 points representing the 256 pixels values on the elliptic curve $y^2 = x^3 + x + 1$ over $\mathbb{F}_{7681}$. Implementation has been done using the computer algebra system SAGE [14]. With SAGE, the points are represented by triplets, which are their projective coordinates. For any point P, the projective coordinates $(X_P : Y_P : Z_P)$ correspond to the affine coordinates $(X_P/Z_P, Y_P/Z_P)$ if Z_P is non-zero, and O if Z_P is zero.

```
[(0 : 1 : 0), (0 : 1 : 1), (0 : 7680 : 1), (1 : 316 : 1), (1 : 7365 :
1), (2 : 196 : 1), (2 : 7485 : 1), (9 : 1621 : 1), (9 : 6060 : 1), (12 :
3734 : 1), (12 : 3947 : 1), (14 : 3674 : 1), (14 : 4007 : 1), (17 : 1331
: 1), (17 : 6350 : 1), (18 : 3831 : 1), (18 : 3850 : 1), (22 : 3483 :
1), (22 : 4198 : 1), (24 : 2822 : 1), (24 : 4859 : 1), (25 : 17 : 1),
(25 : 7664 : 1), (32 : 307 : 1), (32 : 7374 : 1), (33 : 538 : 1), (33 :
7143 : 1), (34 : 832 : 1), (34 : 6849 : 1), (35 : 3124 : 1), (35 : 4557
: 1), (37 : 388 : 1), (37 : 7293 : 1), (39 : 1254 : 1), (39 : 6427 : 1),
(40 : 1771 : 1), (40 : 5910 : 1), (50 : 1157 : 1), (50 : 6524 : 1), (51
: 1462 : 1), (51 : 6219 : 1), (55 : 648 : 1), (55 : 7033 : 1), (56 :
3810 : 1), (56 : 3871 : 1), (60 : 2050 : 1), (60 : 5631 : 1), (62 : 2594
: 1), (62 : 5087 : 1), (63 : 1532 : 1), (63 : 6149 : 1), (65 : 612 : 1),
(65 : 7069 : 1), (67 : 3784 : 1), (67 : 3897 : 1), (68 : 1928 : 1), (68
: 5753 : 1), (71 : 2015 : 1), (71 : 5666 : 1), (72 : 611 : 1), (72 :
7070 : 1), (73 : 71 : 1), (73 : 7610 : 1), (74 : 3411 : 1), (74 : 4270 :
1), (75 : 2416 : 1), (75 : 5265 : 1), (76 : 1693 : 1), (76 : 5988 : 1),
(78 : 3751 : 1), (78 : 3930 : 1), (79 : 96 : 1), (79 : 7585 : 1), (81 :
886 : 1), (81 : 6795 : 1), (82 : 1520 : 1), (82 : 6161 : 1), (83 : 3141
: 1), (83 : 4540 : 1), (84 : 2903 : 1), (84 : 4778 : 1), (85 : 3547 :
1), (85 : 4134 : 1), (87 : 2441 : 1), (87 : 5240 : 1), (88 : 259 : 1),
(88 : 7422 : 1), (92 : 766 : 1), (92 : 6915 : 1), (95 : 1404 : 1), (95 :
6277 : 1), (97 : 1865 : 1), (97 : 5816 : 1), (99 : 1771 : 1), (99 : 5910
: 1), (100 : 3290 : 1), (100 : 4391 : 1), (101 : 3334 : 1), (101 : 4347
: 1), (102 : 1526 : 1), (102 : 6155 : 1), (105 : 399 : 1), (105 : 7282 :
1), (107 : 186 : 1), (107 : 7495 : 1), (108 : 3335 : 1), (108 : 4346 :
1), (110 : 570 : 1), (110 : 7111 : 1), (111 : 2821 : 1), (111 : 4860 :
1), (112 : 902 : 1), (112 : 6779 : 1), (117 : 1100 : 1), (117 : 6581 :
1), (118 : 2446 : 1), (118 : 5235 : 1), (119 : 1110 : 1), (119 : 6571 :
1), (121 : 2338 : 1), (121 : 5343 : 1), (124 : 3398 : 1), (124 : 4283 :
1), (126 : 2602 : 1), (126 : 5079 : 1), (130 : 3080 : 1), (130 : 4601 :
1), (131 : 2201 : 1), (131 : 5480 : 1), (132 : 3767 : 1), (132 : 3914 :
1), (133 : 1274 : 1), (133 : 6407 : 1), (134 : 2357 : 1), (134 : 5324 :
1), (137 : 1479 : 1), (137 : 6202 : 1), (141 : 1003 : 1), (141 : 6678 :
1), (146 : 1876 : 1), (146 : 5805 : 1), (147 : 1547 : 1), (147 : 6134 :
1), (148 : 3350 : 1), (148 : 4331 : 1), (150 : 3769 : 1), (150 : 3912 :
1), (151 : 513 : 1), (151 : 7168 : 1), (153 : 1274 : 1), (153 : 6407 :
1), (155 : 649 : 1), (155 : 7032 : 1), (157 : 532 : 1), (157 : 7149 :
1), (161 : 2321 : 1), (161 : 5360 : 1), (162 : 3254 : 1), (162 : 4427 :
1), (163 : 532 : 1), (163 : 7149 : 1), (166 : 3404 : 1), (166 : 4277 :
1), (173 : 718 : 1), (173 : 6963 : 1), (174 : 3523 : 1), (174 : 4158 :
1), (175 : 2520 : 1), (175 : 5161 : 1), (178 : 307 : 1), (178 : 7374 :
1), (182 : 1658 : 1), (182 : 6023 : 1), (183 : 495 : 1), (183 : 7186 :
1), (185 : 2501 : 1), (185 : 5180 : 1), (189 : 1744 : 1), (189 : 5937 :
1), (190 : 2638 : 1), (190 : 5043 : 1), (191 : 3656 : 1), (191 : 4025 :
1), (192 : 3281 : 1), (192 : 4400 : 1), (193 : 1938 : 1), (193 : 5743 :
1), (198 : 2583 : 1), (198 : 5098 : 1), (201 : 3778 : 1), (201 : 3903 :
1), (203 : 128 : 1), (203 : 7553 : 1), (206 : 1847 : 1), (206 : 5834 :
1), (208 : 1517 : 1), (208 : 6164 : 1), (209 : 1648 : 1), (209 : 6033 :
1), (211 : 2594 : 1), (211 : 5087 : 1), (212 : 1897 : 1), (212 : 5784 :
1), (213 : 2089 : 1), (213 : 5592 : 1), (216 : 21 : 1), (216 : 7660 :
1), (217 : 2874 : 1), (217 : 4807 : 1), (218 : 3814 : 1), (218 : 3867 :
1), (220 : 1283 : 1), (220 : 6398 : 1), (221 : 2158 : 1), (221 : 5523 :
1), (222 : 3516 : 1), (222 : 4165 : 1), (224 : 2415 : 1), (224 : 5266 :
1), (225 : 3409 : 1), (225 : 4272 : 1), (226 : 172 : 1), (226 : 7509 :
1), (228 : 1956 : 1), (228 : 5725 : 1), (229 : 2492 : 1), (229 : 5189 :
1), (230 : 2784 : 1), (230 : 4897 : 1), (231 : 1800 : 1), (231 : 5881 :
1), (232 : 76 : 1), (232 : 7605 : 1), (237 : 2989 : 1), (237 : 4692 :
1), (239 : 2026 : 1), (239 : 5655 : 1), (240 : 3279 : 1), (240 : 4402 :
1), (242 : 1740 : 1), (242 : 5941 : 1), (243 : 3572 : 1), (243 : 4109 :
1), (245 : 3077 : 1), (245 : 4604 : 1), (248 : 3190 : 1), (248 : 4491 :
1), (250 : 3676 : 1), (250 : 4005 : 1), (251 : 2141 : 1), (251 : 5540 :
1), (253 : 1031 : 1), (253 : 6650 : 1), (254 : 1161 : 1), (254 : 6520 :
1), (255 : 3824 : 1)]
```

Figure 2. *List of points corresponding to the 256 pixels values on the elliptic curve* $y^2 = x^3 + x + 1$ *over* $\mathbb{F}_{7681}$

The next figures (Figure 3 and Figure 4) show an image and a sequence of points representing a part of that image.

Figure 3. *Original image "lena"*

[(68 : 91 : 1), (70 : 74 : 1), (79 : 197 : 1), (75 : 219 : 1), (70 : 74
: 1), (60 : 204 : 1), (54 : 202 : 1), (52 : 58 : 1), (48 : 204 : 1), (48
: 47 : 1), (69 : 32 : 1), (87 : 192 : 1), (101 : 198 : 1), (107 : 32 :
1), (98 : 36 : 1), (69 : 32 : 1), (57 : 55 : 1), (72 : 109 : 1), (68 :
160 : 1), (39 : 25 : 1), (48 : 204 : 1), (42 : 191 : 1), (20 : 95 : 1),
(92 : 44 : 1), (121 : 80 : 1), (92 : 207 : 1), (87 : 59 : 1), (140 : 65
: 1), (167 : 122 : 1), (150 : 74 : 1), (143 : 47 : 1), (161 : 98 : 1),
(173 : 152 : 1), (161 : 98 : 1), (180 : 14 : 1), (190 : 37 : 1), (167 :
129 : 1), (72 : 142 : 1), (31 : 74 : 1), (41 : 213 : 1), (36 : 209 : 1),
(48 : 204 : 1), (47 : 31 : 1), (45 : 73 : 1), (45 : 178 : 1), (48 : 204
: 1), (47 : 220 : 1), (42 : 191 : 1), (41 : 213 : 1), (42 : 191 : 1),
(48 : 204 : 1), (54 : 202 : 1), (54 : 202 : 1), (54 : 49 : 1), (60 : 204
: 1), (69 : 219 : 1), (69 : 219 : 1), (62 : 106 : 1), (59 : 240 : 1),
(62 : 145 : 1), (69 : 219 : 1), (53 : 29 : 1), (68 : 160 : 1), (59 : 11
: 1), (52 : 58 : 1), (69 : 219 : 1), (81 : 174 : 1), (75 : 219 : 1), (69
: 219 : 1), (57 : 55 : 1), (45 : 178 : 1), (37 : 95 : 1), (87 : 192 :
1), (138 : 50 : 1), (122 : 194 : 1), (81 : 174 : 1), (110 : 73 : 1),
(180 : 237 : 1), (160 : 208 : 1), (148 : 5 : 1), (155 : 114 : 1), (160 :
208 : 1), (164 : 170 : 1), (179 : 13 : 1), (190 : 214 : 1), (202 : 111 :
1), (119 : 124 : 1), (31 : 177 : 1), (36 : 42 : 1), (59 : 240 : 1), (58
: 12 : 1), (68 : 91 : 1), (63 : 93 : 1), (63 : 158 : 1), (63 : 93 : 1),
(69 : 219 : 1), (73 : 215 : 1), (70 : 74 : 1), (60 : 204 : 1), (69 : 32
: 1), (80 : 215 : 1), (81 : 174 : 1), (75 : 219 : 1), (80 : 36 : 1), (92
: 207 : 1), (83 : 120 : 1), (81 : 77 : 1), (87 : 192 : 1), (98 : 36 :
1), (98 : 36 : 1), (82 : 249 : 1), (72 : 142 : 1), (59 : 11 : 1), (47 :
31 : 1), (59 : 11 : 1), (82 : 2 : 1), (81 : 77 : 1), (68 : 160 : 1), (51
: 210 : 1), (36 : 42 : 1), (37 : 95 : 1), (75 : 32 : 1), (122 : 194 :
1), (148 : 5 : 1), (107 : 219 : 1), (105 : 133 : 1), (140 : 65 : 1),
(169 : 160 : 1), (159 : 227 : 1), (150 : 74 : 1), (147 : 39 : 1), (156 :
82 : 1), (169 : 160 : 1), (176 : 22 : 1), (173 : 152 : 1), (169 : 160 :
1), (91 : 101 : 1), (51 : 210 : 1), (41 : 38 : 1), (48 : 204 : 1), (54 :
202 : 1), (63 : 158 : 1), (73 : 215 : 1), (81 : 77 : 1), (82 : 249 : 1),
(96 : 73 : 1), (101 : 53 : 1), (101 : 53 : 1), (98 : 215 : 1), (101 : 53
: 1), (108 : 110 : 1), (113 : 99 : 1), (112 : 199 : 1), (48 : 204 : 1),
(47 : 31 : 1), (58 : 239 : 1), (62 : 106 : 1), (39 : 25 : 1), (81 : 77 :
1), (110 : 178 : 1), (96 : 178 : 1), (73 : 215 : 1), (69 : 219 : 1), (72
: 142 : 1), (97 : 117 : 1), (91 : 150 : 1), (70 : 74 : 1), (57 : 196 :

Figure 4. *Sequence of points representing a part of image "lena"*

6. Conclusion and perspectives

We presented how to transform an image into a sequence of points of an elliptic curve. With such representation, encryption schemes and various algorithms of Elliptic Curve Cryptography can be applied on images. Some other operations in image processing as watermarking, compression, can also be redefined on images seen as points on elliptic curve. These open doors for new interests in research on image processing and security.

7. Bibliographie

[1] Lafourcade P., *Security and Cryptography just by images*, Université Joseph Fourier, Verimag DCS, 2009

[2] Abdmouleh M.K., Bouhlel M.S., *Effective Crypto-compression Scheme for Medical Images*, International Journal of Signal Processing, 2017

[3] Miao Z., Xiaojun T., *Joint image encryption and compression scheme based on IWT and SPIHT*, Optics and Lasers in Engineering, vol 90, pp 254-274, 2017

[4] Xiaoyong J., Sen B., Guibin Z. and Bing Y., *Image encryption and compression based on the generalized knights tour, discrete cosine transform and chaotic maps*, Multimedia Tools and Applications, vol. 76, Issue 10, pp 1296512979, 2017

[5] Puech W. and Rodrigues J.M., *Crypto-Compression of medical images by selective encryption of DCT*, In EUSIPCO'05, Antalya, Turquie, Septembre 2005

[6] 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, n°5, 2006

[7] Waghmare A., Bhagat A., Surve A., Kalgutkar S., *Chaos Based Image Encryption and Decryption*, IJARCCE, vol 5, 2016

[8] 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

[9] Benabdellah M, Majid H.M., Zahid N., Regragui F. and Bouyakhf E.H., *Encryption-compression of images based on FMT and AES algorithm*, International Journal Applied Mathematical Sciences, Vol. 1, 2007

[10] Jalel H., Mohamed A., Ben F., Mounir S. and Abdennaceur K., *Crypto-compression of images based on chaos*, IEEE, 2013

[11] Masmoudi A., Puech W., *Lossless chaos-based crypto-compression scheme for image protection*, ITE Image Processing, vol 8, 2014

[12] Bos J. W., Halderman J. A., Heninger N.,Moore J., Naehrig M.and Wustrow E., *Elliptic Curve Cryptography in Practice*, IACR, 2013

[13] Koblitz N., *A Course in Number Theory and Cryptography - 2nd ed.*, Springer-Verlag, 1994

[14] Stein W., *Sage Tutorial - Release 8.1*, The Sage Development Team, 2017