Digital Signature of an Image by Elliptic Curve Cryptosystem

T.N. Shankar[1,*], G. Sahoo[2], and S. Niranjan[3]

[1] Department of Comp.Sc. and Engg., GMR Institute of Technology,
Rajam, Andhra Pradesh, India
tnshankar2003@yahoo.co.in

[2] Dept. of Information Technology, Birla Institute of Technology Mesra,
Ranchi, Jharkhand India
gsahoo@bitmesra.ac.in

[3] Department of Information Technology, PDM College of Engineering,
Bahadurgarh, Haryana, India
niranjan.hig41@gmail.com

Abstract. Digital Images for seamless transmission over mobile network are required to be ensured with authentication and increasing concern for their integrity, originality and non repudiation qualities. In this paper, a novel approach of digital signature scheme of an image is introduced. The proposed scheme consists of three main steps. First, pixel selection, second, pixel values digest through hash function, and finally, creation of digital signature using elliptic curve cryptography based on public key cryptosystem. In this process, the sender uses the private key for hash value encryption, and the recipient uses the public key for signature verification. Instead of directly sending an original image to the recipients, the embedded copy is sent with signed digital signature for authentication.

1 Introduction

With the introduction of mobile devices, smart cards and many other gadgets, it has become an easy task to deal with images over any mobile network. Further improvement in processing power, miniaturization of portable storage media have considerably enhance multimedia transmission capability [6][7]. Despite these technological advances over last few years, users are likely to face situations, where the contents so received may have the possibility of being tampered with, producing copies of, and illegally redistributing digital contents[15][16]. Without solving these security related issues, digital multimedia products and services cannot succeed in the domain of ecommerce. An attempt to work out a tangible solution for achieving image authentication using digital signature [10][11] by Elliptic Curve

[*] Corresponding author.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part III, LNICST 86, pp. 337–346, 2012.

Cryptography(ECC) is introduced. A comparative study of the conventional algorithm and the one that employs ECC justifies our contention.

Images are popular in multimedia with various applications. Each image is composed of pixels. Each pixel can be represented by eight bits [12][18] with its value between 0-255. In the proposed algorithm, a hash function SHA-1 is used to transform some pixels of the image to obtain the digest value followed by encrypting with private key to which ECC is applied for generation of digital signature[8][14]. The image so embedded with digital signature [2][17] received at the destination can be separated by using the method discussed in section 4.3. After separation of digital signature, the hash function SHA-1 is operated upon the selected pixels to generate the message digest. In the end, to assess any undesirable interference, the signature verification on digested message using the public key can be prepared. The security feature attributed to digital signature depends on encryption and verification. Conventional algorithms are not suitable for mobile devices due to more space complexity. Among the non-conventional algorithms, RSA encryption algorithm is not suitable due to large key size. Multimedia files are transmitted as compressed files. If digital signature [8][16] processing will have complexity issues, then its application with the mobile devices is not desirable.

In the following section, there is a brief discussion on background of ECC and point multiplication over GF(p) and $F_2{}^m$ in section 2. In Section 3, we discuss on digital signature. Section 4 describes on proposed algorithm. Section 5 presents a comparative analysis. Finally section 6 concludes the paper.

2 Elliptic Curve

Informally, an elliptic curve is a type of cubic curve defined by an equation of the form

$$y^2 = x^3 + ax + b \tag{1}$$

Where a and b are real numbers.

The definition of elliptic curve also requires that the curve be non-singular. Geometrically, this means that the graph has no cusps, self-intersections, or isolated points. Algebraically, this involves calculating the discriminant

$$D = 4a^3 + 27b^2 \neq 0 \tag{2}$$

The curve is non-singular if and only if the discriminant is not equal to zero. It is not safe to use singular curves for cryptography as they are easy to crack. Due to this reason we generally take non-singular curves for data encryption.

2.1 Elliptic Curves over $F_2{}^m$

The binary finite field implementation is that the elements of $GF(2^m)$ can be represented by m-bit binary code words . This implies that a finite field of form $GF(2^m)$is of characteristic 2. The equation of the elliptic curve on a binary field $F_2{}^m$ is

$$y^2 + xy = x^3 + ax^2 + b \text{ , where } b \neq 0. \tag{3}$$

Here the elements of the finite field are integers of length at most m bits. These numbers can be considered as a binary polynomial of degree
$m–1$.In binary polynomial the coefficients can only be 0 or 1.

2.2 Arithmetic in GF(2^m)

In this section, we want to introduce briefly the arithmetic operations needed to implement ECC point multiplication over binary polynomial fields GF(2^m). The operations include modular addition, subtraction, multiplication, squaring, division, and inverse, where the operands are polynomials with coefficients of either 0 or 1. We use the polynomial basis representation where a polynomial $a(x) \in GF(2\ ^m)$ in canonical form is written as

$$a(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + ...+ a_2x^2 + a_1x + a_0 \tag{4}$$

$a_i \in GF(2)$. For computation purposes, an m bit binary vector can be used to represent the coefficients. For example, the polynomial $x^3 + x^2 + 1$ can be written as 1101.

2.3 Point Multiplication Algorithm

Scalar multiplication is the computation of the form $Q = kP$ where P and Q are the elliptic curve points and k is an integer. This is achieved by repeated elliptic curve point addition and doubling operations. Point negation also includes as a miscellaneous operation, which is about to be suggested for fast implementation algorithm of kP.

The integer k is represented as

$$k = x_{m-1}2\ ^{m-1} + x_{m-2}2\ ^{m-2} + \dots + x_1 + x_0 \tag{5}$$

where

$x_i \in \{1,0\}$ and $m = 0,1,2, \dots , n$-1

The addition of two points on a curve over $F_2{}^m$ is defined as.

Let $P(x_1,y_1)$, and $Q(x_2,y_2)$ be two different points on the curves, when $P \neq Q$ the operation $P + Q = (x_3, y_3)$ can then be derived as shown as shown in

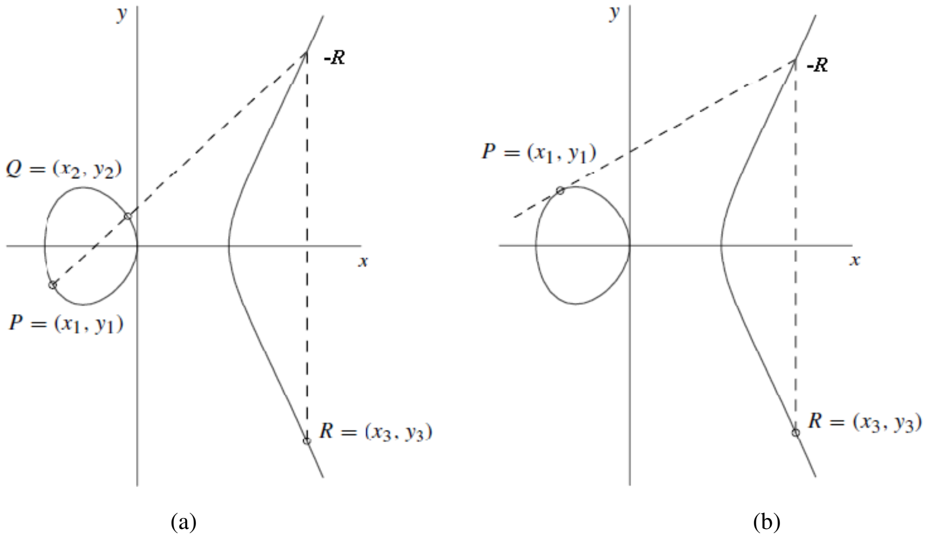


Fig. 1. a. Point Addition P+Q= R, b. The operation $P + P = 2P$

Algorithm.1.*a*

Algebraic Formulae Over *Fp*

Point addition

$P+ \infty= P$

Case I. If $P \neq Q$, then $((x_1 \neq x_2)$ and $(y_1 \neq y_2))$

Step1. Find $\alpha = (y_2 - y_1) / (x_2 - x_1)$

Step2. $P+Q = (x_3,y_3) = (\alpha^2 - x_1 - x_2 , \alpha(x_1 - x_3) - y_1)$

Point doubling

Case II. If $P = Q$, then $((x_1=x_2)$ and $(y_1=y_2))$

Step1. Find $\alpha = (3x_1{}^2 + a)/(2y_1)$

Step2. $2P = (x_3,y_3) = (\alpha^2 - x_1 - x_2 , \alpha (x_1 - x_3) - y_1)$

Algorithm.1.b

Algebraic Formulae Over $F_2{}^m$

Point addition

```
Case I.
Step1.If  $P \neq Q$, then $((x_1 \neq x_2)$ and $(y_1 \neq y_2))$
Step2. Find $\alpha = (y_1 + y_2) / (x_1 + x_2)$
Step3. $P+Q = (x_3,y_3) = (\alpha^2 + \alpha + x_1+x_2 + a , \alpha(x_1+x_3)+x_3+y_1)$
```

Point double

```
Case II.
Step1. If $P = Q$, then $((x_1=x_2)$ and $(y_1=y_2))$,
Step2. Find $\alpha = x_1 + (y_1 / x_1)$
Step3. $2P = (x_3,y_3) = (\alpha^2 + \alpha + a , x_1^2 + (\alpha + 1)x_3)$
```

Algorithm:1.c

Point negation

```
Step1. If $Q = (x, y)$ is over $F_2{}^m$ then
        $-Q = (x, x + y)$ is over $F_2{}^m$
```

3 Digital Signature by ECC

INPUT: Domain parameters = (a, b, P ,h, n) private key g, pixel values m.
OUTPUT: Signature (s_1, s_2)

1. Select $k \in$ R $[1,n\text{-}1]$
2. Compute $kP = (x_1, y_1)$ and convert x_1 to an integer $\overline{x_1}$.
3. Compute $s_1 = x_1$ mod n. If $s_1 = 0$ then go to step 1.
4. Compute $d = SHA\text{-}1\ (m)$.
5. Compute $s_2 = k^{-1}(d + g\ s_1)$ mod n. If $s_2 = 0$ then go to step 1.
6. Return(s_1, s_2).

3.1 ECDSA Signature Verification

INPUT: Domain parameters=(q, a, b, P, n) public key Q, pixel values m,
signature(s_1, s_2).
OUTPUT: Acceptance or rejection of the signature.

1. Verify that s_1 and s_2 are integers in the interval $[1, n-1]$. If any verification failsthen return ("Reject the signature").

2. Compute $d=SHA\text{-}1\ (m)$.
3. Compute $w=s_2-1$ mod n.
4. Compute $v_1 = dw$ mod n and $v_2=s_1\ w$mod n.
5. Compute $X = v_1P+v_2Q$.
6. If $X = \infty$then Invalid the signature";
7. Convert the x-coordinate x_1 of X to an integer x_1; compute z=x_1 mod n.
8. If $z=s_1$ then return ("Accept the signature");
9. Else return (" Reject the signature ").

4 Experimental Results

Select the pixels according to desire. More pixels may be assumed for better signature strength. Preparation of digital signature through all the pixels is not so easy and it is too complicated with more time and space complexity. To avoid this situation, select 100 to 1000 number of pixels for preparation of it with best security level will well suit the purpose.

Select the pixels

1. RGB = imread('Litesh.jpg')
2. col = [m ,n, …]
3. row = [x ,y,…]
4. pixels = im pixel (RGB, col, row)

Assume 4 pixels from Fig.2.a

For testing we have assumed only four pixels of Fig.2, and their representation is as like as the following.

Table 1. Four pixels with the pixel value eight bit each

0	1	1	0	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	1
1	1	0	0	0	1	1	0

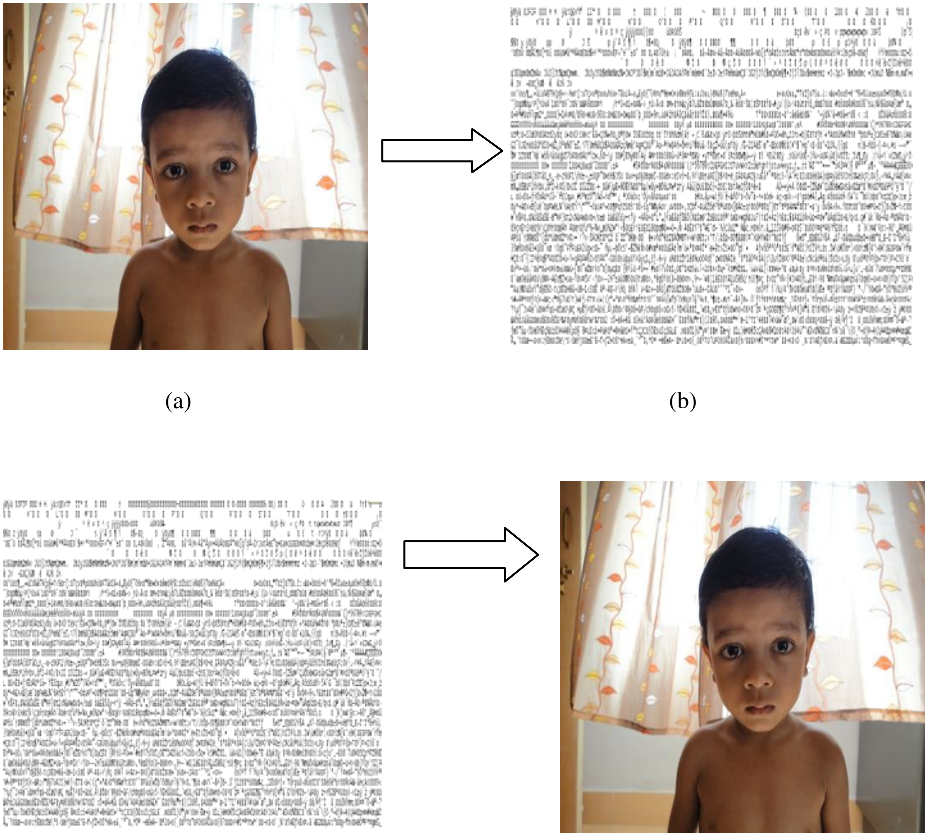
Message M = "01100001 01100010 01100011 11000110"

4.1 Digital Signature by ECC

SHA-1(M) = e364706816aba3e25717850c26c9cd0d89d60e4c

s_1 = 8bac1ab66410435cb7181f95b16ab97c92b341c0

s_2 = 41e2345f1f56df2458f426d155b4ba2db6dcd8c8



(a) (b)

(c) (d)

Fig. 2. a. Litesh.jpg, b. Machine Language Litesh.jpg, c. Machine Language after embedding digital signature(s_1, s_2), d. Litesh.jpg after embedding digital signature(s_1, s_2)

After creating the digital signature, it's necessary to embed within the image. Fig.2.a to d depict all the processing steps to embed the digital signature with the image.

Here we have illustrated a comparison between before Fig.2.b and after Fig.2.c embedding the digital signature within the image. If we will go after an observation then we can find out the differences amongst the symbols in machine language.

4.2 Embedded Digital Signature Extraction Process

In this process the data embeddable pixels are identified from the embedded image. These pixels are given as input to the next stage i.e., the color of each embeddable pixel and those of its four precedent neighbors are given as input to the color ordering and mapping function. If the output is '1' then the extracted secret bit is taken to be '1' otherwise, '0'. The extracted digital signature is compared with the extracted embedded data for verification.

Recipient will use the same positional pixels which were used by sender as the input to SHA-1 to create the digested message. The image embedded with digital signature can always be robustly verified for authenticity even though the attacker in the transit will attempt to interfere. However, the identical pixel selection at the source and at the destination ends to remain intact lest the authentication and the integrity of the image will be under siege.

4.3 ECDSA Signature Verification

z = 8bac1ab66410435cb7181f95b16ab97c92b341c0

$s_1 = z$, So the signature is acceptable

Suppose any tampering is there then there must be impact on "z" and z value must not be matched with s_1.

4.4 Experimental Result after Tampering

Litesh.jpg is tampered on the mid then we can get the output z as

z = 8b431ab665d1435cb7181f95b16df37c92b341c0

$s_1 \neq z$, As a consequence signature is not acceptable

5 Comparison between Digital Signature by RSA and ECC

Both the algorithms can be used for preparation of digital signature from an image. We can't use RSA algorithm for mobile devices due to its long key size. Desired level of security can be achieved with ECC using small key size. A comparison between RSA and ECC algorithms is shown in Figure-3. For this occasion Elliptic Curve Cryptography proved to be appropriate for image encryption on mobile devices.

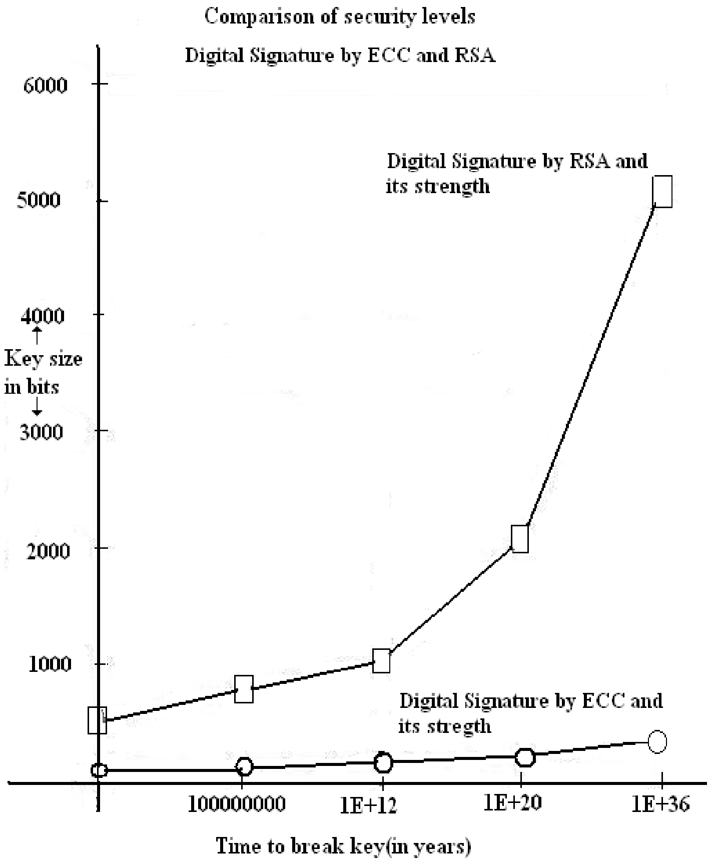


Fig. 3. Comparison between digital signature of an image by RSA and ECC

6 Conclusion

For image authentication, it is desired that the verification method be able to resist content-preserving modifications while being sensitive to content-changing modifications. In this paper, an attempt is made to embed digital signature of an image with ECC for achieving better result. We have assumed only 4 pixels for manual calculation. In the real applications, 100-1000 pixels will cover a secure and strong digital signature. Preparation of digital signature from all pixels requires more time and space that may not be suitable for small devices due to limited space and less processing power. To overcome such type of problem, we propose this technique with limited number of pixels for preparation of digital signature keeping well in view the micro devices with better strength.

References

[1] Koblitz, N., Menezes, A.J., Vanstone, S.A.: The state of elliptic curve cryptography. Design, Codes, and Cryptography 19(2-3), 173–193 (2000)

[2] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithm. IEEE Trans. Informn, Theory IT-31(4), 469–472 (1985)

[3] Shankar, T.N., Sahoo, G.: Cryptography with ellipitic curves. International Journal of Computer Science And Applications 2(1), 38–42

[4] Shankar, T.N., Sahoo, G.: Cryptography with ASCII codes. International Journal of Secure Digital Age Information 1(2), 141–121

[5] Younes, M.A.B., Jantan, A.: Image Encryption Using Block-Based Transformation Algorithm. IAENG International Journal of Computer Science, IJCS 35(1) (2003)

[6] Shankar, T.N., Sahoo, G., Niranjan, S.: Elliptic Curve Point Multiplication by Using Complementary Recording for Image Encryption. In: INCOCCI 2010, Ieee Xplore, pp. 546–551 (2010)

[7] Shankar, T.N., Sahoo, G., Niranjan, S.: Image Encryption for Mobile Devices. In: ICCCCT 2010, Ieee Xplore, pp. 612–516 (2010)

[8] Johnson, D., Menezes, A., Vanstone, S.: The Elliptic Curve Digital Signature Algorithm (ECDSA). Certicom Research, Canada

[9] Stallings, W.: Cryptography and Network Security: Principles and Practice, 2nd edn. Prentice-Hall (1999)

[10] Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P., Reginelli, M.: A New Chaotic Algorithm for Video Encryption. IEEE Trans. Consumer Electron. 48(4), 838–844 (2002)

[11] Sudharsanan, S.: Shared Key Encryption of JPEG Color Images. IEEE Trans. Consumer Electron. 51(4), 1204–1211 (2005)

[12] Dang, P.P., Chau, P.M.: Image Encryption for Secure Internet Multimedia Applications. IEEE Trans. Consumer Electron. 46(3), 395–403 (2000)

[13] Schneider, M., Chang, S.-F.: Robust Content Based Digital Signature for Image Authentication. In: Proceedings of IEEE International Conference on Image Processing (ICIP 1996), vol. 3, pp. 227–230 (1996)

[14] Lou, D.-C., Liu, J.-L.: Fault Resilient and Compression Tolerant Digital signature for Image Authentication. IEEE Transactions on Consumer Electronics 46(1), 31–39 (2000)

[15] Fridrich, J.: Robust Bit Extraction from Images. In: Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS 1999), vol. 2, pp. 536–540 (1999)

[16] Jansirani, A., Rajesh, R., Balasubramanian, R., Eswaran, P.: Hi-Tech Authentication for Palette Images Using Digital Signature and Data Hiding. The International Arab Journal of Information Technology 8(2), 117–123 (2011)

[17] Lu, C.-S., Liao, H.-Y.M.: Structural Digital Signature for Image Authentication, An Incidental Distortion Resistant Scheme. IEEE Transactions on Multimedia 5(2), 161–173 (2003)

[18] Alwan, R.H., Kadhim, F.J., Al-Taani, A.T.: Data Embedding Based on Better Use of Bits in Image Pixels. International Journal of Information and Communication Engineering 2(2), 104–107 (2006)