Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 472 – 481

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Image Encryption using Elliptic Curve Cryptography

Laiphrakpam Dolendro Singh* and Khumanthem Manglem Singh

National Institute of Technology, Manipur 795 001, India

Abstract

Million of images are transferred everyday across the network. Some of these images are confidential and we want these images to be transferred securely. Cryptography plays a significant role in transferring images securely. The exponentially hard problem to solve an Elliptic Curve Discrete Logarithm Problem with respect to key size of Elliptic Curve Cryptography, helps in providing a high level of security with smaller key size compared to other cryptographic technique which depends on integer factorization or Discrete Logarithmic problem. In this paper, we implement the Elliptic Curve cryptography to encrypt, decrypt and digitally sign the cipher image to provide authenticity and integrity.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Elliptic Curve Cryptography; Digital signature; Elliptic curve discrete logarithm problem; Authenticity; Integrity.

1. Introduction

A lot of information is perceived when we observe an image. Images have become an inevitable source of information. Everyday we come across various image from various sources. When images are confidential and we want the image to be transferred safe and securely, cryptography comes into play. The cryptographic technique which we have implemented in this paper is the Elliptic Curve Cryptography (ECC). Various study on ECC has concluded that the difficultly to solve an Elliptic Curve Discrete Logarithmic Problem is exponentially hard with respect to the key size used. This property makes ECC a very good choice for encryption/decryption process compared to other cryptographic techniques which are linearly difficult or sub exponentially difficult. ECC is a public key cryptography which was developed by Neal Koblitz and Victor S. Miller independently in the year 1985. ECC gains wide acceptance around 2004.

1.1 Mathematical operation

Some of the mathematical operation that we will be using while performing the implementation of the image encryption/decryption using Elliptic Curve Cryptography are described here.

*Corresponding author. Tel.: +918974867524.
E-mail address: ldsingh.cse@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.054

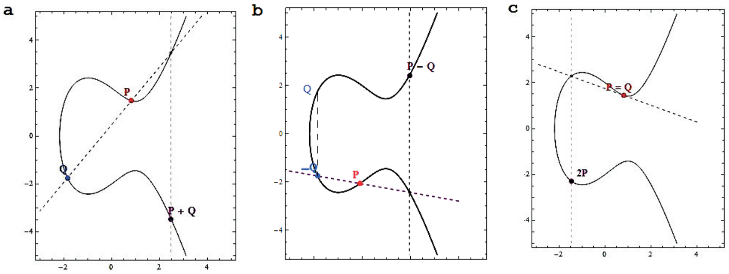

Fig. 1. (a) Point addition; (b) Point subtraction; (c) Point doubling.

1.1.1 Point addition

In Elliptic Curve Cryptography, operations are performed on the coordinate points of an elliptic curve. To perform addition of two distinct point coordinate the following calculation is used. Figure 1(a) shows graphical representation of point addition.

$$P(x_1, y_1) + Q(x_2, y_2) = R(x_3, y_3) \tag{1}$$

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p \tag{2}$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p \tag{3}$$

where

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} \bmod p \tag{4}$$

1.1.2 Point subtraction

To perform point subtraction, get a mirror coordinate of the subtracted point along x-axis and perform point addition on the resulting coordinate and the other coordinate. Figure 1(b) shows graphical representation of point subtraction.

$$P(x_1, y_1) - Q(x_2, y_2) = P(x_1, y_1) + Q(x_2, -y_2) \tag{5}$$

1.1.3 Point doubling

Point doubling is perform to add up two points which are same i.e. they have same coordinate value. Figure 1(c) shows graphical representation of point doubling.

$$P(x_1, y_1) + Q(x_1, y_1) = R(x_3, y_3) \tag{6}$$

$$x_3 = (\lambda^2 - 2x_1) \bmod p \tag{7}$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p \tag{8}$$

where

$$\lambda = \frac{(3x_1^2 + a)}{(2y_1)} \bmod p \tag{9}$$

1.1.4 Point multiplication

Multiplication is repeated addition of the base coordinate point. Many algorithm have been develop to perform point multiplication swiftly. $kP = P + P + P + \cdots + k$ times.

1.2 Encryption and decryption using ECC

Let Alice and Bob be the two communicating party. The communicating parties agrees upon the Elliptic curve equation and a Generator (G).

$$y^2 = \{x^3 + ax + b\} \bmod [p] \tag{10}$$

Suppose Alice want to encrypt a message 'Pm' and send to Bob. The cipher text is given by $Pc = [kG, Pm + kPb]$ where 'k' is a random integer and 'Pb' is the public key of Bob computed using the private key of Bob 'nB', $Pb = nBG$. Bob decrypts the cipher message as, message $= [Pm + kPb - nBkG]$. Since $Pb = nBG, kPb$ and $nBkG$ cancel each other and 'Pm' remains, which is the message sent by Alice.

2. Literature Survey

Darrel Hankerson, Alfred Menezes and Vanstone explained the various elliptic curve arithmetic, issue with implementation and cryptographic protocols in details[1]. Lawrence C. Washington provided proofs to various theory related to elliptic curve[2]. Jouko Teeriaho shows implementation of various aspect of ECC using Mathematica[3]. To securely transfer image across the network various techniques have been develop in recent years using ECC. Ahmed A, Abd El-Latif and Xiamu Niu presented an image encryption technique using cyclic elliptic curve and chaotic system. They proposed a technique to generate a pseudo random key stream using cyclic elliptic curve point and chaotic system which in turn is used for encryption of data stream from the image[4]. Hong Liu and Yanbing Liu gave a cryptanalysis of image encryption scheme based on hybrid chaotic system and cyclic elliptic curve. They found that known-plain text attack and choosing a plain image with all the pixel value 0 can generate the encrypted image[5]. S. Maria Celestin Vigila and K. Muneeswaran proposed an algorithm to perform image encryption using ECC[6]. They use a coupled Linear congruential generator to generate the private key and a random integer 'k'. To obtain the cipher image point multiplication is performed for each pixel value with the generator to fit into the elliptic curve coordinate. A mapping table is required while performing decryption. Ali Soleymani, Md Jan Nordin and Zulkarnain Md Ali proposed an encryption technique using Elliptic Curve over Prime Group field[7]. They created a mapping table which has values of 0 to 255 along the row and the corresponding row contains the elliptic curve coordinate. Pixel value of the image are maped onto elliptic curve coordinate using the table and encrypted using the public key of the receiver. To view the encrypted data as cipher image the table is used again to map the values back to the range of 0 to 255. S. Behnia, A. Akhavan, A. Akhshani, A. Samsudin proposed a novel image encryption algorithm using Jacobian elliptic map[8]. They transform the plain image data matrix into one dimension matrix after operating with the key which is the initial condition and control parameters. Elements of the matrix are encrypted using an equation and matrix is reshaped back to original dimension. Li Li, Ahmed A. Abd El-Latif, Xiamu Niu proposed an encryption scheme using Elliptic curve ElGamal based homomorphic image for sharing secret images[9]. They selected the parameter of the elliptic curve to resist Pollard's rho, isomorphic and Pohlig Hellman attack. The experimental result indicates that it is better than encryption using RSA and ElGamal. Don Johnson, Alfred Menezes and Scott Vanstone describe the implementation, related security and interoperability issue of Elliptic Curve Digital Signature Algorithm[10]. Lo'ai Tawalbeh, Moad Mowafi and Walid Aljoby presented two ECC based encryption on image. First one is on selective quantised DCT coefficients and second is on selective bit plane[11]. Dr. Mamfred Lochter propose a set elliptic curve domain parameters over a finite prime field[12]. Ann Hibner Koblitz, Neal Koblitz, Alfred Menezes describe various ups and down with ECC and people acceptance with time. They also provide the idea of social construction of technology[13]. W. Stalling gave a detailed idea of various cryptographic algorithm and network security idea[14]. ECC Brainpool[15] and NIST (National Institute of Standards and Technology)[16] gave various recommended ECC parameters of various bit sizes. Images that we have used here are obtained from USC-SIPI (University of Southern California-Signal and Image Processing Institute)[17].

3. Problem Statement

Usually image encryption using ECC are performed by mapping the pixel values to Elliptic curve coordinate. For mapping, a separate look up table is required or used the point multiplication operation of pixel value with Generator

'G' to produce affine coordinate on the elliptic curve. In these cases, mapping table is required while decryption process to generate the corresponding pixel value from the cipher image. In our algorithm, we work on group of pixels to reduce the number of computation. The group of pixels are transformed into big integer single digits keeping in mind that it should not exceed 'p' value which is one of the parameter in elliptic curve equation of finite field. These big integer values are paired and given as input denoted by 'Pm' in ECC operation. This operation help us to ignore the mapping operation and the need to share mapping table between sender and receiver.

4. Proposed Algorithm

To decrease the number of computational steps in ECC operation, we perform two operations defined below.

4.1 Pixel grouping into a single integer

Images are made up of pixels. If cryptographic operation is performed on every single pixel it will take more time as the number of pixels present is very large. So, it will be a good option to group the pixels together. The number of pixels to be group depends on the Elliptic Curve parameters used. The larger the parameter of the elliptic curve, the more pixel can be grouped. For example a 512 bit ECC parameter can group upto 63 pixels together. To get the number of pixels to be group, find the number of the list, of the base 256 digits in the integer 'p' minus 1. To convert the group of pixels into a big single integer we have used a function of Mathematica called FromDigits [list of pixels, b] which take a list of pixels and convert it to base b. We add random 1 or 2 to each pixel to avoid error caused while using FromDigits function of Mathematica, in case, the first pixel value of the group is 0 and also to provide low correlated pixel value for the cipher image generated with same pixel value plain image. Pixel value of image in byte form will range from 0 to 255. So the maximum possible pixel value of the image will be 257 including the 2 we added. So, we will use base value 'b' as 258.

4.2 Getting the group of pixels from the big integer

After the ECC operation the coordinate value will all be in the range of the bit size chosen for the ECC operation. To generate the cipher image from these coordinates we need to bring it down to 0 to 255 range. We performed using the IntegerDigits [big integer value, 256] function in Mathematica. It takes as input the big integer values in the range of the size chosen for ECC operation and with base 256, the output will be a list of values ranging from 0 to 255. The two function, FromDigits[ ] and IntegerDigits[ ] are inverse of each other so the pixels value are preserved during the operation.

Mathematical operation on an image is done on the pixels value of the image. So first, we get the pixels value of the image. The Elliptic curve parameters $\{a, b, G, p\}$ are agreed between the sender and the receiver. The sender use the public key 'Pb' of the receiver to generate the cipher image from the pixels of the plain image. The receiver use the private key 'nB' which was used to generate the public key, to decrypt the cipher image back to the plain image.

4.3 Image encryption

1. Get the pixel value of the image to be encrypted and randomly add 1 or 2 to each pixel. Record the number of channels present in the image.
2. Group the pixels and convert to single large integer value for each group. Number of pixel to be group using Mathematica is given by $grp =$ Length [IntegerDigits[p, 258]] $- 1$
3. Pair up the result obtained from step 2 and store as 'Pm' which is the plain message input for the ECC system.
4. Select a random 'k' and compute 'kG' and 'kPb' where 'Pb' is the public key of the receiver.
5. Perform point addition of 'kPb' with each value of 'Pm' and store as 'Pc' which is the cipher text.
6. Convert the cipher text list from step 5 to value ranging from 0 to 255.
7. Pad left with 0 to each list from step 6 which have less than $grp + 1$ number of elements, to make each list equal in length.

8. Flatten the list from step 7, group them according to the number of image channels that we have recorded and partition them to width of the plain image.
9. Convert the values from step 8 into cipher image.

4.4 Digital signature on cipher image

For performing digital signature we can still use the ECC parameters used for encryption.

1. The sender selects a private key 'nA' and generate the public key $Pa = nAG$.
2. Get the Hash value of the pixel values of the cipher image and store as 'z'.
3. Sender select a random integer 'k' in the range of [1 to $n - 1$] where n is the cyclic order of the Elliptic Curve with G as Generator.
4. Compute the digital signature pair $\{r, s\}$ where

$$r = \{kG\} \bmod [n]_{x-\text{coordinate}} \tag{11}$$

$$s = \left\{\frac{z + rnA}{k}\right\} \bmod [n] \tag{12}$$

Send kG, cipher image, Digital Signature $\{r, s\}$.

4.5 Image Decryption

1. Get the pixel value of the cipher image and group by $grp + 1$ number of pixels and form single big integer value for each group with base 256. Record the number of image channels of the cipher image.
2. Pair up the value obtained from step 1.
3. Perform point multiplication of 'kG' with 'nB' where 'nB' is the private key of the receiver.
4. Perform point subtraction between values from step 2 with value from step 3.
5. Get the value in the range of 0 to 255 from step 4 with base 258 and subtract random 2 from each value.
6. Group the flatten value obtained in step 5 in term of recorded number of image channels of the cipher image and partition them to the width of the cipher image.
7. Convert the values from step 6 into plain image.

4.6 Verifying the signature

1. Calculate hash value of cipher image pixel value.
2. Obtain

$$w = \frac{1}{s} \bmod [n] \tag{13}$$

3. Calculate

$$u_1 = \{z * w\} \bmod [n] \tag{14}$$

$$u_2 = \{r * w\} \bmod [n] \tag{15}$$

4. Compute

$$\{x_1, y_2\} = u_1 G + u_2 Pa \tag{16}$$

5. If

$$r == \{x1\} \bmod [n] \tag{17}$$

signature is verified.

5. Simulation Result of Image Encryption/Decryption with Digital Signature

The implementation is performed on i7 CPU 2.20 GHz lenovo laptop with 8 GB RAM using Mathematica version 10. The Elliptic curve used here is the 512 bit Standard Elliptic curve given by ECC Brainpool

$$y^2 = \{x^3 + ax + b\} \bmod [p] \tag{18}$$

where p = 8948962207650232551656602815159153422162609644098354511344597187200057010413552439917934304191956942765446530386427345937963894309923928536070534607816947;

a = 62948605579730632276664213064763793240747157706227462271369104454503019142812760980279909684079839626911518536785638778342218340274397182380657258442641 38;

b = 32457890083289670592748495843420779165319090096375019183283236687361791765832634964635251284888282611559800773506973771797764811498834995234341530862286627;

G = {67920591404245751744356404312691950878431533901025218814680230127320474825798530775456474462728667949363715224107745326865824846179460139288742968443515 22, 65922445552401128733247483814296103413127129403262663313274450666870105454152564610977074832886502169926130901850429577163183011801592347885043076285093 30};

n = 89489622076502325516566028151591534221626096440983545113445971872000570104134185283789817306435249598574513983700292805830942156138820439733543921155441 69;
Here n is the cyclic order of the Elliptic curve with G as generator

nB = 96192759682482119853328425949563698712343813919172976158104477319333745612481875498805879175589072651261284189679678167647067832308974867524089740051 33;
Here nB is the private key of the receiver.

Pb = {15590930657409568825430318608173946658236459324800564696743236222451134371211804313902595174231019209568426636822542309107445298000868493241598468431010 49, 26876285442561939153229265600256697689944205079516735232855198767579543612512349739545593625623982738189077712025830446937430498896365776069720065519756 71};
Pb is the public key of the receiver, which is formed using the private key of the receiver, given by point multiplication of nB and G.

nA = 94268904488832477456261857430572424738096937640789516634942387772947070700232237988829761592077291198236058505886084604294126475673608974091172098560224 01;
nA is the private key of the sender

Pa = {77511187111048294650456399420708073707837290480871566989671986079254879520158149804269980291496112687534710426194837742349300715457321680491523551899648 49, 58735024067276542220759198140648261016906441523144403900269298371649521237910226239943370030592982018358818460509976413039545990712468144658861929061944 93};
Pa is the public key of the sender.

The sender sends the cipher image along with the Digital Signature calculated with the pixels value of the cipher image. The public key of the receiver was used to encrypt the image and the private key of the sender was used to provide digital signature to the cipher image being sent. While decrypting the cipher image, the receiver uses his private key to decrypt the cipher image. Authenticity provides the proof that the message came for the intended sender and integrity validates that the message was not altered or changed during the transit. An altered cipher image will generate

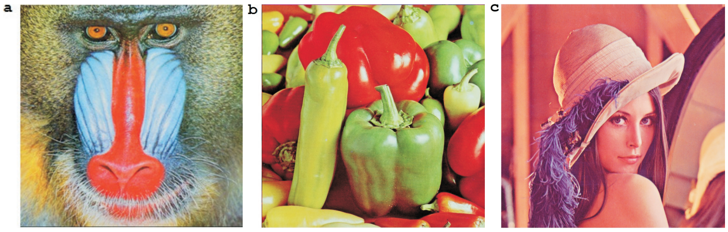

Fig. 2. (a) Plain image of mandrill; (b) Plain image of peppers; (c) Plain image of lena.

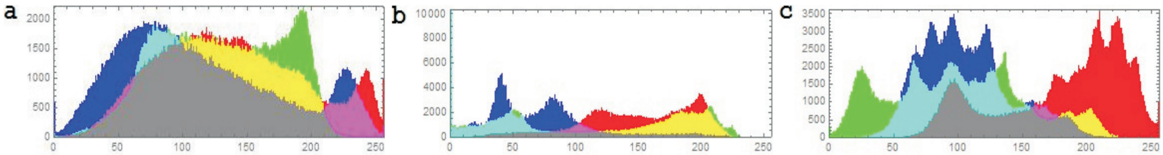

Fig. 3. (a) Histogram of mandrill; (b) Histogram of peppers; (c) Histogram of lena.

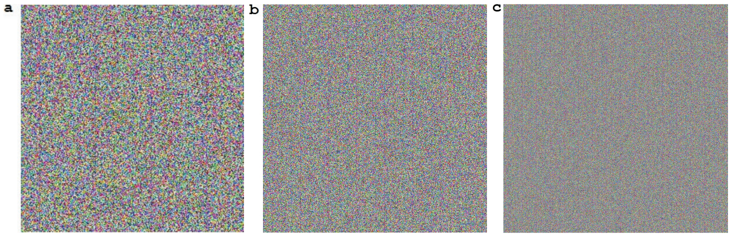

Fig. 4. (a)Cipher image of mandrill; (b) Cipher image of peppers; (c) Cipher image of lena.

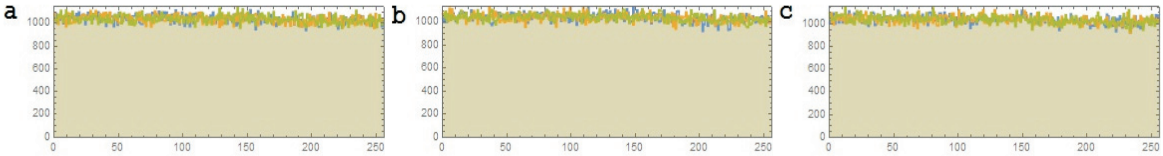

Fig. 5. (a) Histogram of cipher mandrill; (b) Histogram of cipher peppers; (c) Histogram of cipher lena.

a different hash value. To test the authenticity and integrity of the cipher image received, the public key of the sender and the hash value of the cipher image along with the digital signature $\{r, s\}$ are used to verify the Digital Signature.

6. Security Analysis

Security analysis of a cryptographic process is an essential process to ensure the strength of the cryptographic technique. In this section we discuss some analysis of the implemented technique.

6.1 Histogram analysis

Histogram of an image depicts the frequency of each pixels. A good cipher image has a uniform frequency distribution of the pixel values. Figure 3 and Fig. 5 show the histogram of the plain image and cipher image respectively. We can see that plain image frequency distribution of pixels varies a lot and the cipher image histogram is evenly distributed which indicates the goodness of the cipher image generated.

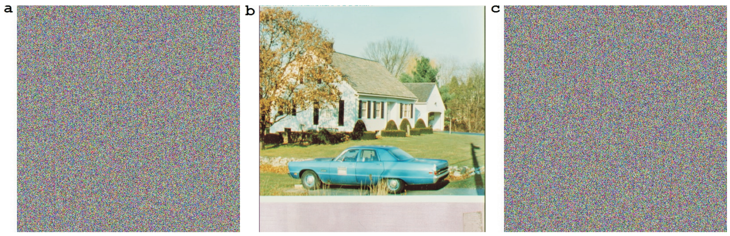

Fig. 6. (a) Cipher image of house; (b) Decrypted image with correct key nB; (c) Decrypted with key as $nB - 1$.

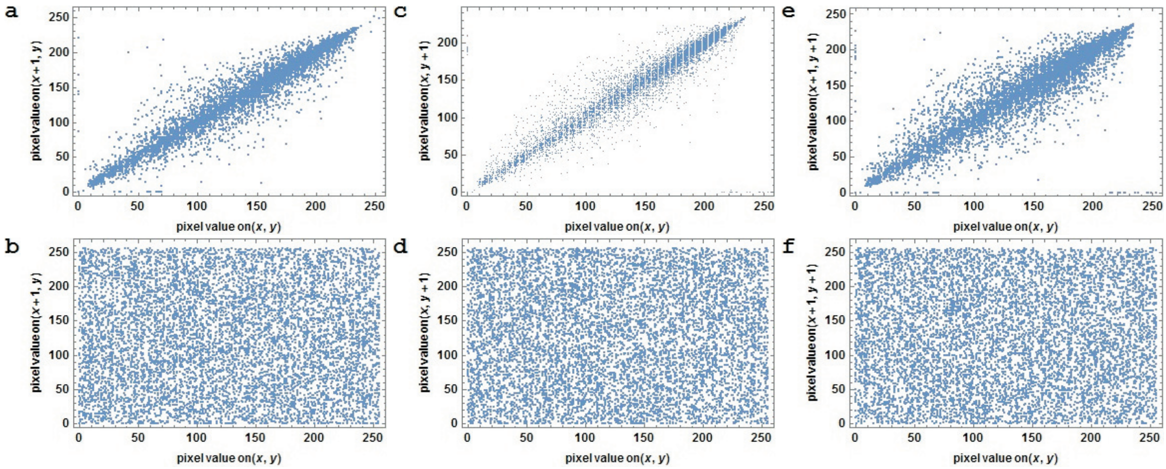

Fig. 7. Correlation of adjacent pixel (a) Along horizontal direction for plane image; (b) Along horizontal direction for cipher image; (c) Along vertical direction for plane image; (d) Along vertical direction for cipher image; (e) Along diagonal direction for plane image; (f) Along diagonal direction for cipher image.

6.2 Key space

Security of an encryption and decryption depend a lot on the size of the key used. The bigger the key size, the more it is difficult to perform an attack using Brute Force attack. ECC provides an exponentially difficult Elliptic Curve Discrete Logarithmic Problem with respect to the key size. In our implementation we have used a 512 bit standard Elliptic curve which has got an ample key size to provide the necessary security.

6.3 Key sensitivity

A slight change in original key should give a drastic change in the recovered image obtained from the cipher image. Figure 6 shows the recovered image from cipher image using the correct private key of the receiver and a wrong key which is just one digit different from the original key.

6.4 Correlation analysis

Normal images that we see everyday have pixel values which have high correlation with their neighbours. A good cipher image will have a very low correlation to its neighbour pixel value. We randomly choose 3000 pixel values from the house image and plot the adjacent pixel values, along the horizontal, vertical and diagonal direction shown in Fig. 7. We can see from the figure that for house image the pixel plot are not spread everywhere while pixel plot of cipher image are spread everywhere which signifies confusion and diffusion property required for a cipher image.

Correlation coefficient of the RGB color channel along horizontal, vertical and diagonal direction of plane image and cipher image is given in Table 1 and 2.

Table 1. Correlation coefficient of pixel for plane image shown in Fig. 6(a).

Component	Horizontal	Vertical	Diagonal
Red	0.946726	0.931007	0.898708
Green	0.920307	0.912032	0.850211
Blue	0.968626	0.952602	0.927192

Table 2. Correlation coefficient of pixel for cipher image shown in Fig. 6(a).

Component	Horizontal	Vertical	Diagonal
Red	−0.00672293	0.000402062	0.0147189
Green	0.0177625	0.0175243	−0.00258394
Blue	−0.0153138	−0.000152592	−0.0207352

Table 3. Entropy analysis.

Cipher Image	Size	Entropy
Mandrill	256*260	7.99884
Pepper	512*520	7.99963
Lena	1024*1040	7.99986

Table 4. Speed analysis.

Image	Size	Encryption Time	Decryption Time	Digital Signature	Digital Signature Verification
Lena	1024*1024	2.47 sec	1.58 sec	4.37 sec	4.38 sec
Mandrill	512*512	0.79 sec	0.60 sec	1.39 sec	1.37 sec
Peppers	256*256	0.29 sec	0.30 sec	0.48 sec	0.44 sec

6.5 Entropy analysis

Entropy is the measure of degree of randomness. For image encryption, we want the cipher image pixel values to be highly random. A good cipher image will have an entropy value close to 8. Table 3 shows the various entropy values of the cipher image.

6.6 Speed analysis

The execution time of encryption and decryption depends on various factor like programming skills, algorithm, hardware where the program is implemented, size of the image etc. We have implemented the execution using the parameters we mention on section 4. Table 4 shows the encryption, decryption, digital signature and digital signature verification execution time.

6.7 Known plain text attack

Assume that an attacker knows the ciphertext, encryption algorithm and one or more plaintext-ciphertext pairs formed using a secret key. The beauty of ECC in encryption is that, it generates a totally different ciphertext using the same key with every execution. A random factor 'k' comes into computation during the encryption process which creates the difference in each execution. Figure 8(b) and (c) show the cipher images generated using the same key of

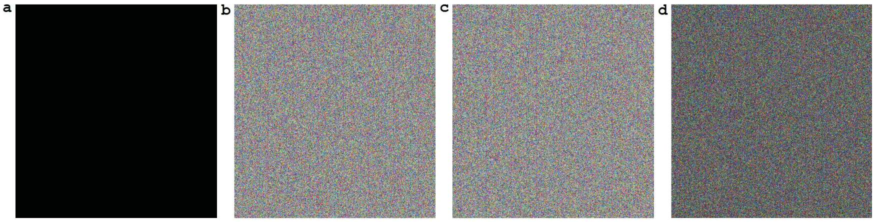

Fig. 8. (a) Totally black plane image; (b) Cipher image generated on first execution with key nB; (c) Cipher image generated on second run with key nB; (d) Image difference between the two cipher images.

a totally black image which have all the pixel value as 0, shown in Fig. 8(a). In our algorithm we added random 1 or 2 to each pixels of the image. The random integers helps in generating a low correlated cipher image. The two cipher images generated cannot be distinguish with our naked eye, so a function is used which shows the difference between the two cipher images by getting the corresponding absolute difference of the pixels value in Fig. 8(d). An attacker cannot find the secret key using Known Plain text attack.

7. Conclusion

In the paper we have presented the implementation technique of image encryption/decryption and inclusion of digital signature to the cipher image to provide authenticity and integrity to the received image. We have performed our operation by grouping the pixel and explained how many pixel can be grouped according to the ECC parameters. Pairing of the grouped pixel value was performed instead of mapping those values to Elliptic curve coordinate. It helps to ignore the used of reference mapping table for encryption and decryption. Our algorithm generates a low correlated cipher image even with a image which is made up of same pixel value. We have also analysed our technique to support the strength of the algorithm.

References

[1] Darrel Hankerson, Alfred Menezes and Scott Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, (2004).

[2] Lawrence C. Washington, *Elliptic Curves Number Theory and Cryptography*, Taylor & Francis Group, Second Edition, (2008).

[3] Jorko Teeriaho, *Cyclic Group Cryptography with Elliptic Curves*, Brasov, May (2011).

[4] A. Ahmed, Abd El-Latif and Xiamu Niu, A Hybrid Chaotic System and Cyclic Elliptic Curve for Image Encryption, In *AEU-International Journal of Electronics and Communications*, Elsevier, issue 2, vol. 67, pp. 136–143, (2013).

[5] Hong Liu and Yanbing Liu, Cryptanalyzing an Image Encryption Scheme based on Hybrid Chaotic System and Cyclic Elliptic Curve, In *Optics and Laser Technology*, Elsevier, vol. 56, pp. 15–19, (2014).

[6] S. Maria Celestin Vigila and K. Muneeswaran, Nonce Based Elliptic Curve Cryptosystem for Text and Image Applications, In *International Journal of Network Security*, vol. 14, no. 4, pp. 236–242, July (2012).

[7] Ali Soleymani, Md Jan Nordin and Zulkarnain Md Ali, A Novel Public Key Encryption based on Elliptic Curves Over Prime Group Field, In *Journal of Image and Graphics*, vol. 1, pp. 43–49, (2013).

[8] S. Behnia, A. Akhavan, A. Akhshani and A. Samsudin, Image Encryption based on the Jacobian Elliptic Maps, In *The Journal of System and Software*, Elsevier, vol. 86, pp. 2429–2438, (2013).

[9] Li Li, Ahmed A. Abd El-Latif and Xiamu Niu, Elliptic Curve ElGamal Based Homomorphic Image Encryption Scheme for Sharing Secret Images, In: *Signal Processing*, Elsevier, vol. 92, pp. 1069–1078, (2012).

[10] Don Johnson, Alfred Menezes and Scott Vanstone, The Elliptic Curve Digital Signature Algorithm (ECDSA), *Certicom Corporation*, (2001).

[11] Lo'ai Tawalbeh, Moad Mowafi and Walid Aljoby, Use of Elliptic Curve Cryptography for Multimedia Encryption, *IET Information Security*, vol. 7, issue 2, pp. 67–74, (2012).

[12] Manfred Lochter, *ECC Brainpool Standard Curves and Curve Generation v.1.0*, Bonn, (2005).

[13] Ann Hibner Koblitz, Neal Koblitz and Alfred Menezes, Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift, In *Journal of Number Theory*, Elsevier, vol. 131, pp. 781–814, (2011).

[14] Williams Stallings, *Cryptography and Network Security*, 4th Edition, Prentice Hall. Pearson, (2000).

[15] www.ecc-brainpool.org/download/Domain-parameters.pdf; (2005).

[16] www.csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf; (1999).

[17] www.sipi.usc.edu/database/database.php?volume=misc