Journal of Engineering and Applied Sciences 12 (13): 3329-3333, 2017
ISSN: 1816-949X

A New Text Encryption Technique on Elliptic Curve Cryptography

[1]Wrya Karim Kadir, [1]Omed Hassan Ahmed and [2]Mohammed Rafiq Namiq
[1]Department of Information Technology, College of Science and Technology,
University of Human Development, Kurdistan, Iraq
[1]Department of Mathematics, College of Science, University of Sulaimani, Sulaymaniyah, Iraq

Abstract: Elliptic curve cryptography has been studied extensively for many years and the cryptosystems based on Elliptic Curve Cryptography (ECC) is becoming the recent trend of public key cryptography. This study presents the implementation of ECC using both Bob's public key and common shared key in the encryption process. This feature increases the security of scheme. In contrast with common proposed techniques we eliminate the shared look up table between the sender and receiver and we do not need to map the ASCII value to an affine point in the elliptic curve. The technique is designed in such a way that can be used to encrypt and decrypt any type of script with ASCII value defined.

Key words: Elliptic curve cryptography, public key cryptography, common shared key, ASCII value, techniques

INTRODUCTION

In 1985, Miller (1986) and Koblitz (1987) invented elliptic curve cryptography which is not new anymore. They proposed a public key cryptosystem analogue of El-Gamal scheme. They used a group of points on an elliptic curve defined over a finite field instead of using the group Z^*_p. The best-known algorithm for solving underlying hard mathematical problems such as discrete logarithm problem and integer factorization problem over elliptic curve takes fully exponential time. This is the main attraction of using elliptic curve cryptography over the known technologies such as RSA and DSA. Using significantly smaller parameters in ECC offers the equivalent level of security in compare to RSA. From the practical point of view, the performance of the ECC comes from the efficiency of the finite filed computations and quick algorithms for elliptic scalar multiplication.

Elliptic curves: Elliptic curves are mathematical constructs that have been studied by mathematicians since the seventeenth century. Elliptic curves are not ellipse. The elliptic curves arecurves having a specific base point, these are given by explicit polynomial equations called "Weierstrass equations". Using these explicit equations we show that the set of pointsof an elliptic curve forms an Abelian group (Yan, 2000). In cryptography we are interested to work on elliptic curves over prime finite field and binary finite field. In the present study we are working on prime finite field.

First assume that the finite field F_p has characteristic greater than three ($p>3$). The elliptic curve over F_p is defined by $E(F_p)$ is the set of all solution to the equation:

$$y^2 = x^3 + ax + b \bmod p \quad (1)$$

where $a, b \in F_p$ and $\Delta = 4a^3+27b^2 \neq 0 \bmod p$ together with an extra point O which is called point at infinity. The suitable curve for cryptography must be "non-supersingular" because it is vulnerable against attackers. The curve is said tobe "non-supersingular" if the number of elliptic curve points not equal to p+1 (i.e., $\#E(F_p) \neq p+1$) (Yan, 2000).

Addition of points on elliptic curve: In this subsection, we will describe the addition of points on elliptic curve. The addition of two points on an elliptic curve has its own rule and it is not adding two points in the coordinate system. Two points will be added geometrically on elliptic curve. Let:

$$E: y^2 = x^3+ax+b$$

and $P_1 = (x_1, Y_2)$, $P_2 = (x_2, y_2)$ be two points on the curve which satisfies the above equation E. To obtain the third point P_3 on that curve we can add P_1 and P_2. To find this point we draw a line L which passes through P_1 and P_2 and intersect the curve at P'_3. By taking the reflection across the x-axis we obtain P_3. So:

Corresponding Author: Wrya Karim Kadir, Department of Information Technology, College of Science and Technology, University of Human Development, Kurdistan, Iraq

$$P_3 = P_1 \oplus P_2$$

This means that adding P_1 and P_2 will gives P_3. Now we discuss the addition of P_1 and P_2 on elliptic curve in more detail. There are 4 cases:

- $P+O = O+P = P$ for all $P \in E(F_p)$
- If $P_1 = (x, y) \in E(F_p)$ and $P_2 = (x,-y) \in E(F_p)$ then $P_1+P_2 = 0$
- Let $P_1 = (x_1, y_1) \in E(F_p)$ and $P_2 = (x_2, y_2) \in E(F_p)$ where $P_1 \neq \pm P_2$. Then $P_1+P_2 = (x_3, y_3)$ where $x_3 = \lambda^2-x_1-x_3$ $y_3 = \lambda(x_1-x_3)-y_1$; $\lambda = y_2-y_1/x_2-x_1$
- Let $P_1 = P_2 \in E(F_p)$. Then $P_1+P_2 = 2P_1 = (x_3, y_3)$ where $x_3 = \lambda^2-2x_1$, $y_3 = \lambda(x_1-x_3)-y_1$

$$\lambda = \frac{\{3x_1^2 + a\}}{2y_1}$$

The operation in the fourth case is called the doubling of a point.

Literature review: Our research is somehow similar to the work by Sagheer (2012). In fact, our scheme is derived from the scheme by Sagheer (2012) in two steps with many modifications using proposed algorithms by Teeriaho *et al*. (2011). Many authors have explained the influence of ECC and come up with new ideas that can increase the security of the scheme. They have been implemented in various tasks like authentication, digital signature and encryption. Miller (1986) invented using elliptic curve in cryptography. His algorithm was similar to Diffie Hellman key exchange but 20% faster. The 2 years later, Koblitz (1987) explained using elliptic curve over finite field in public key cryptography. He mentioned that the DLP over finite field is harder than binary field. Koblitz *et al*. (2000) surveyed the development of elliptic curve cryptography since its invention. They provide the factors that can be used to choose a right elliptic curve for cryptography and also they discussed the advantages of employing elliptic curve in public key cryptosystems. In terms of mathematica implementation (Teeriaho, 2011) gave examples of elliptic curve key exchange, encryption and digital signature. While Kolhekar and Jadhav (2011) implement C++ in text encryption. There are several proposed algorithms (Kumar *et al*., 2012; Esfahani *et al*., 2013) that convert the message to ASCII values and then map the ASCII values to an affine point in the elliptic curve but these methods need sharing a look up table. It has been demonstrated by Gapanathy and Mani at (Tahmassebpour, 2017), using fuzzy modular arithmetic in AT89C51 micro-controller consumes less time compared to non-fuzzy modular arithmetic.

MATERIALS AND METHODS

Proposed algorithm: Similar to the other proposed algorithms, two communication parties agree upon an elliptic curve $E(F_p)$ that can be identified by the parameters a, b and p:

$$y^2 = x^3+ax+b \bmod p \quad (2)$$

To generate the second public parameter, Alice and Bob use the generator G which is a point in elliptic curve and their private key K_a and K_b to compute K_aG and K_bG. Alice and bob's public keys are:

$$W_a = K_aG \quad (3)$$

$$W_b = K_bG \quad (4)$$

Alice uses her private key and Bob's public key to get the common key which is:

$$K = K_aK_bG = (k_1, k_2) \bmod p \quad (5)$$

Also, Bob can similarly get the same key $K = (k_1, k_2)$ which is a point in the elliptic curve. If Alice wants to send a message to Bob use the Bob's public key and the common shared key. The message characters and symbols do not convert to affine points in the elliptic curve so the parties do not need to share the common look up table.

Firstly we map the message characters to ASCII value and then group them up using mathematica functions. The size of each group is given by:

$$\text{Group size} = \text{Length}\left[\text{Integer digits}\left[p, 256\right]\right]+1 \quad (6)$$

Integer digits [n, b] gives a list of the base b digits in the integer n for example, integer digits (25.2) = {11001}. Here we use b = 256 because the range of english char is 0-255 and ASCII value is defined till 255. Length (expr) gives the number of elements in the given expression. For example, length (integer digits (25.2) Length (11001) = 5. The above calculation will help us to find the maximum number of characters that can be grouped up.

RESULTS AND DISCUSSION

Encryption

Step 1: Obtain the plaintext to send.

Step 2: Convert it to its corresponding ASCII value.

Step 3: Use the group size to partition the ASCII values into equally size groups with no overlapping and the later sub lists with lesser group size are left as it is without padding.

Step 4: Use from digits Mathematica's function to convert the obtained groups from the previous step to a big integer number with base 256:

$$\text{From digits } [\text{Group of ASCII values; 256}] \quad (7)$$

For example; from digits (list, b) = From digits (f 11001g; 2) = 25.

Step 5: Pad with 32 to the end of the list from the above step if the count of the above list is odd to make it even for performing complete pairing. The obtained big integers from the above step will be paired up and used as an elliptic curve point in the ECC systems.

Step 6: Use the obtained pairs as $M = (m_1, m_2)$ to compute $C = (c_1, c_2)$:

$$c_1 = m_2k_1 + m_1 \bmod p \quad (8)$$

$$c_2 = m_2 + m_2k_1k_2 + m_1k_2 \bmod p \quad (9)$$

Step 7: Select random λ value, λ = random value with range 1 to p-1. Compute λ G and λ W_b using point multiplication operation.

Step 8: Compute $C+\lambda W_b$ using point addition or point doubling as required.

Step 9: Alice sends $P_C = \{\lambda G, C+\lambda W_b\}$ as cipher text to Bob. Bob will receive the P_C as a package which contains λG and $C+\lambda W_b$ separately.

Decryption

Step 1: Obtain the cipher text P_c.

Step 2: Obtain the left part λG and the right part $C+\lambda W_b$ separately.

Step 3: Bob multiplies his private key K_b to the left part λG and subtract it from the right part to get C:

$$\{C + \lambda W_b\} - K_b\lambda G = C \quad (10)$$

Since, $W_b = K_bG$. Subtraction operation can be converted to addition by multiplying with -1 to the y coordinate. This operation can be justified with point addition operation. In point addition we used to get the mirror image point over the x-axis. For example, -(21, 37) = (21, -37).

Step 4: Bob uses the key $K = (k_1, k_2)$ to get $M = (m_1, m_2)$:

$$m_2 = c_2 - k_2c_1 \bmod p \quad (11)$$

$$m_1 = c_1 - k_1m_2 \bmod p \quad (12)$$

We can prove that the step 4 in the decryption process is the inverse of the step 6 in the encryption process (Sagheer, 2012):

$$m_2 = c_2 - k_2c_1 = (m_2 + m_2k_1k_2 + m_1k_2) - k_2(m_2K_1 + m_1) \bmod p = (m_2 + m_2k_1k_2 + m_1k_2 - m_2k_1k_2 - m_1k_2) = m_2 \bmod p$$

$$m_1 = c_1 - m_2k_1 = m_2k_1 + m_1 - m_2k_1 = m_1 \bmod p$$

We behave with the numbers m_1 and m_2 as two big integer values and we do not employ them as a pair we use them separately.

Step 5: The big integer values obtained from the step 4 are formed by combining group of ASCII values using from digits function. So, convert them back to get the group of ASCII values by using integer digits function with base 256:

$$\text{Integer Digits (big integer; 256)} \quad (13)$$

Step 6: Convert the list of ASCII values to its corresponding characters.

Example: Alice wants to send the following message Cryptography is the art of writing or solving codes. To Bob using the elliptic curve $y^2 = x^3+5829x+2079 \bmod 7829$ with G = (2436, 4951) and #E (F_{7829}) = 7897. Alice and Bob's Private keys are K_a = 7487 and K_b = 6737. So, the shared key will be $K_a K_b G = K = (k_1, k_2) = (3098, 7235)$.

Encryption process

Step 1: The plaintext is: cryptography is the art of writing or solving codes.

Step 2: The corresponding ASCII values are: 67, 114, 121, 112, 116, 111, 103, 114, 97, 112, 104, 121, 32, 105, 115, 32, 116, 104, 101, 32, 97, 114, 116, 32, 111, 102, 32, 119, 114, 105, 116, 105, 110, 103, 32, 111, 114, 32, 115, 111, 108, 118, 105, 110, 103, 32, 99, 111, 100, 101, 115, 46.

Step 3: Use the equation group size = Length [integer digits [7829, 256]]+1 = 3 to get the groups size. The ASCII values are partitioned into equally sized groups with size 3 {67, 114, 121}, {112, 116, 111}, {103, 114, 97}, {112, 104, 121}, {32, 105, 115}, {32, 116, 104}, {101, 32, 97}, {114, 116, 32}, {111, 102, 32}, {119, 114, 105}, {116, 105, 110}, {103, 32, 111}, {114, 32, 115}, {111, 108, 118}, {105, 110, 103}, {32, 99, 111}, {100, 101, 115}, {46}.

Step 4: Convert each group into big integers using from digits (Group of ASCII value, 256) gives: 4420217, 7369839, 6779489, 7366777, 2124147, 2126952, 6627425, 7500832, 7300640, 7828073, 7629166, 6758511, 7479411, 7302262, 6909543, 2122607, 6579571, 46.

Step 5: The above list is even so we do not need to pad with 32.

Step 6: The obtained pairs from the previous steps are used as (m_1, m_2) in the following formula to get (c_1, c_2). The shared key $K = (k_1, k_2) = (3098, 7235)$:

$$c_1 = m_2k_1+m_1 \bmod p \quad c_2 = m_2+m_2k_1k_2+m_1k_2 \bmod p$$

(6209, 2063), (3795, 210), (6276, 3953), (5738, 5722), (5409, 3846), (1027, 2708), (6007, 7500), (5394, 6802), (4797, 384).

Step 7: Select random λ value. We can use different λ for different round of the program to get different ciphertext even for the same plaintext but here we use a single λ for all rounds. Let λ = 5467. Using point multiplication and doubling we get $\lambda G = (7257, 3229)$ and $\lambda W_b = (4785, 7286)$.

Step 8: Use the points obtained in step 6 as C to find $C+\lambda W_b$. (472, 1966), (3145, 5162), (5502, 3239), (2350, 7753), (6071, 2212), (1039, 528), (1883, 6698), (5920, 448), (3536, 1120).

Step 9: Alice sends $Pc = \{\lambda G, C+\lambda W_b\}$ to Bob.

Decryption process

Step 1: Bob obtains the cipher text $P_c = \{\lambda G, C+\lambda W_b\}$.

Step 2: The left part $\lambda G = (7257,3229)$ and the right part $C+\lambda W_b$ is (472, 1966), (3145, 5162), (5502, 3239), (2350, 7753), (6071, 2212), (1039, 528), (1883, 6698), (5920, 448), (3536,1120).

Step 3: Bob multiplies his private key K_b to the left part λ G and subtract it from the right part to get C:

$$\{C+\lambda W_b\} - K_b\lambda G = C \quad K_b\lambda G = (7257,3229) = (1063,829)$$

So:

$$(472, 1966)-(1063, 629) = (6209, 2063)$$
$$(3145, 5162)-(1063, 829) = (3795, 210)$$

and so on. Hence, Bob gets C as:

(6209, 2063), (3795, 210), (6276, 3953), (5738, 5722), (5409, 3846), (1027, 2708), (6007, 7500), (5394, 6802), (4797, 384)

Step 4: Bob uses the common shared key $K = (k_1, k_2) = (3098,7235)$ to get $M = (m_1, m_2)$ using the equations:

$$m_2 = c_2-k_2c_1 \bmod p$$
$$m_1 = c_1-k_1m_2 \bmod p$$

$$m_2 = 2063-(7235)\,(6209) = 2750 \equiv 7369839 \bmod p$$
$$m_1 = 6209-(3098)\,(2750) = 4661 \equiv 4420217 \bmod p$$

and so on. The recovered message $M = (m_1, m_2)$ is the following: (4420217, 7369839), (6779489, 7366777), (2124147,2126952),(6627425,7500832),(7300640,7828073), (7629166,6758511),(7479411,7302262),(6909543,2122607), (6579571,46).

Step 5: Use from digits functions to get the ASCII values back. {67, 114, 121}, { 112, 116, 111}, {103, 114, 97}, {112, 104, 121}, {32, 105, 115}, {32, 116, 104}, {101, 32, 97}, {114, 116, 32}, {111, 102, 32}, {119, 114, 105}, {116, 105, 110}, {103, 32, 111}, {114, 32, 115}, {111, 108, 118}, {105, 110, 103}, {32, 99, 111}, {100, 101, 115}, {46}.

Step 6: Convert the ASCII values to its corresponding characters. This is the original message. Cryptography is the art of writing or solving codes.

CONCLUSION

In this study, we present a new algorithm to perform text encryption using ECC. Here we map the text to ASCII value and divide these values onto groups. The presented base to determine the group size can be changed. The base p should be greater than all the ASCII values in the text. The algorithm can be generalized for any script from any language if we define p = 65536. Since, the ASCII value is defined till 65535. The ASCII values are mapped to a big integer value which increases the diffusion. We paired up the big integers but used them separately in the Eq. 8 and 9 the output of the step 6 will be paired up and fed up as plaintext in the ECC operation. This process helps removing the expensive operation of mapping the

values to affine points. By this we achieve the strength of the current studywhich is elimination the look up table. Let N be the number of bits we want our primes p to be, the overall complexity is O (N^3) . In practical point of view, our algorithm has lots of positive aspect. The encryption and decryption process can be performed quickly and even with large number of values as input gives smaller size of cipher text in compare to other available techniques and tis consumes less bandwidth while sending.

REFERENCES

Esfahani, M., M. Emami and H. Tajnesaei, 2013. The investigation of the relation between job involvement and organizational commitment. Manage. Sci. Lett., 3: 511-518.

Koblitz, N., 1987. Elliptic curve cryptosystems. Math. Comput., 48: 203-209.

Koblitz, N., A. Menezes and S. Vanstone, 2000. The state of elliptic curve cryptography. Des. Code Cryptogr., 19: 173-193.

Kolhekar, M. and A. Jadhav, 2011. Implementation of elliptic curve cryptography on text and image. Intl. J. Enterp. Comput. Bus. Syst., 1: 1-13.

Kumar, D., C.H. Suneetha and A. Chandrasekhar, 2012. Encryption 0f data using elliptic curve over finite field. Intl. J. Distrib. Parallel Syst., 3: 301-308.

Miller, V.S., 1986. Use of Elliptic Curves in Cryptography. In: Advances in Cryptology-CRYPTO'85, Williams, H.C. (Ed.)., Springer-Verlag, Berlin, Heidelberg, pp: 417-426.

Sagheer, A.M., 2012. Elliptic curves cryptographic techniques. Proceedings of the 6th International Conference on Signal Processing and Communication Systems (ICSPCS'12), December 12-14, 2012, IEEE, Gold Coast, Queensland, Australia, ISBN: 978-1-4673- 2392-5, pp: 1-7.

Tahmassebpour, M., 2017. A new method for time-series big data effective storage. IEEE Access, 2007: 10694-10699.

Teeriaho, J., 2011. Cyclic Group Cryptography with Elliptic Curves. Lapland University of Applied Sciences, Rovaniemi, Finland,.

Yan, S.Y., 2000. Number Theory for Computing. Springer, Berlin, Germany, ISBN:9783540654728, Pages: 381.