European Journal of Scientific Research
ISSN 1450-216X Vol.77 No.2 (2012), pp.252-258

http://www.europeanjournalofscientificresearch.com

Proposed New Elliptic Curve Cryptography Protocol Based on Dictionary Techniques

Hala Bahjat Abdul Wahab
Computer Science Depart., University of Technology Baghdad, Iraq
E-mail: hala_bahjat@yahoo.com

Abdul Monem S. Rahma
Computer Science Depart., University of Technology, Baghdad, Iraq
E-mail: monemrahma@yahoo.com

Abstract

In recent years, Elliptic Curve Cryptography (ECC) has attracted the attention of researchers and product developers due to its robust mathematical structure and highest security compared to other existing algorithms like RSA (Rivest Adleman and Shameer Public key Algorithm). It is found to give an increased security compared to RSA for the same key-size or same security as RSA with less key size [9]. In this paper proposed a new application for elliptic curve cryptosystem work as symmetric cryptographic system by investigate from dictionary techniques. A new protocol that proposed implemented to transfer secret massages between the sender and receiver using dictionary techniques for English text. The new protocol test it by execute encryption and decryption process with more flexible and efficient.

Keywords: Elliptic curve, Key exchange, Cryptography protocols, Dictionary techniques for English text and Finite field.

1. Introduction

With the proliferation of the handheld wireless information appliances, the ability to perform security functions with limited computing resources has become increasingly important. In mobile devices such as personal digital assistants (PDAs) and multimedia cell phones, the processing resources, memory and power are all very limited, but he need for secure transmission of information may increase due to the vulnerability to attackers of the publicly accessible wireless transmission channel [1]. New smaller and faster security algorithms provide part of the solution, the elliptic curve cryptography ECC provide a faster alternative for public key cryptography. Much smaller key lengths are required with ECC to provide a desired level of security, which means faster key exchange, user authentication, signature generation and verification, in addition to smaller key storage needs. The terms elliptic curve cipher and elliptic curve cryptography refers to an existing generic cryptosystem which use numbers generated from an elliptic curve. Empirical evidence suggests that cryptosystems that utilize number derived from elliptic curve can be more secure [2]. As with all cryptosystems and especially with public-key cryptosystems, it takes years of public evaluation before a reasonable level of confidence in a new system is established. ECC seem to have reached that level now. In the last couple of years, the first commercial implementations are appearing, as toolkits but also in real-world applications, such as

email security, web security, smart cards, etc. The security of ECC has not been proven but it is based on the difficulty of computing elliptic curve discrete logarithm in the elliptic curve group [3]. In this paper produced new application for EEC work as symmetric key system that make the secret massages transfer more faster and more efficient.

2. Previous Research

Some cryptographic algorithms have gained popularity due to properties that make them suitable for use in constrained environment like mobile information appliances, where computing resources and power availability are limited. One of these cryptosystems is Elliptic curve which requires less computational power, memory and communication bandwidth compared to other cryptosystem. A collection of papers related some of them Kolhekar & Jadhav [10], study and give a brief background of key exchange and encryption/decryption using ECC. Then explain implementation of these algorithms on text documents, used C++ as the tool for implementation.

3. Elliptic Curves Over Z_p

Elliptic curve cryptography makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field. Two families of elliptic curves are used in cryptographic applications: prime curves over Z_p and binary curves over $GF(2^m)$. For a prime curve over Z_p, we use a cubic equation in which the variables and coefficients all take on values in the set of integers from 0 through p -1 and in which calculations are performed modulo p. For a binary curve defined over $GF(2^m)$, the variables and coefficients all take on values in $GF(2^n)$ and in calculations are performed over $GF(2^n)$. [4] Points out that prime curves are best for software applications, because the extended bit-fiddling operations needed by binary curves are not required; and that binary curves are best for hardware applications, where it takes remarkably few logic gates to create a powerful, fast cryptosystem.

4. Elliptic Curve Cryptography [4]

In general, public-key cryptography systems use hard-to-solve problems as the basis of the algorithm. The most predominant algorithm today for public-key cryptography is RSA, based on the prime factors of very large integers. While RSA can be successfully attacked, the mathematics of the algorithm has not been comprised, computational brute-force has broken the keys. The defense is "simple" — keeps the size of the integer to be factored ahead of the computational curve.

In 1985, Elliptic Curve Cryptography (ECC) was proposed independently by cryptographers Victor Miller (IBM) and Neal Koblitz (University of Washington). ECC is based on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). Like the prime factorization problem, ECDLP is another "hard" problem that is deceptively simple to state: Given two points, P and Q, on an elliptic curve, find the integer *n*, if it exists, such that P = nQ.

Elliptic curves combine number theory and algebraic geometry. These curves can be defined over any field of numbers (i.e., real, integer, complex) although we generally see them used over finite fields for applications in cryptography. An elliptic curve consists of the set of real numbers (x,y) that satisfies the equation: $y^2 = x^3 + ax + b$.

The set of all of the solutions to the equation forms the elliptic curve. Changing *a* and *b* changes the shape of the curve, and small changes in these parameters can result in major changes in the set of (x,y) solutions. Refered as Ep(a,b).

Figure 1: Elliptic curve Addition.

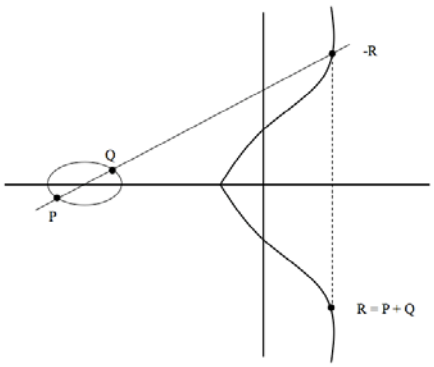

Figure-1 shows the addition of two points on an elliptic curve. Elliptic curves have the interesting property that adding two points on the elliptic curve yields a third point on the curve. Therefore, adding two points, P and Q, gets us to point R, also on the curve. Small changes in P or Q can cause a large change in the position of R. So let's go back to the original problem statement from above. The point Q is calculated as a multiple of the starting point, P, *or*, Q = nP. An attacker might know P and Q but finding the integer, *n*, is a difficult problem to solve. Q (i.e., nP) is the public key and *n* is the private key.

4.1. Blum Blum Shub Generator (BBS) [7,6]

A popular approach to generating secure pseudorandom number is known as the Blum, Blum, Shub (BBS) generator, named for its developers. It has perhaps the strongest public proof of its cryptographic strength. The procedure is as follows. First, choose two large prime numbers, p and q, that both have a remainder of 3 when divided by 4. That is, $p \equiv q \equiv 3 \pmod 4$, The BBS is referred to as a **cryptographically secure pseudorandom bit generator (CSPRBG)**. A CSPRBG is defined as one that passes the next-bit test, which, in turn, is defined as follows: A pseudorandom bit generator is said to pass the next-bit test if there is not a polynomial-time algorith that, on input of the first k bits of an output sequence, can predict the $(k + 1)^{st}$ bit with probability significantly greater than 1/2. In other words, given the first k bits of the sequence, there is not a practical algorithm that can even allow you to state that the next bit will be 1 (or 0) with probability greater than 1/2. For all practical purposes, the sequence is unpredictable. The security of BBS is based on the difficulty of factoring n. That is, given n, we need to determine its two prime factors p and q. The BBS generator produces a sequence of bits Bi according to the following algorithm:

$X_0 = s^2 \bmod n$
for i = 1 to ∞
$X_i = (X_{i1})^2 \bmod n$
$B_i = X_i \bmod 2$

5. Dictionary Technique [8]

Compression schemes like the popular and patented Lempel-Ziv algorithm are called dictionary schemes because they build a big list of common words in the file. This list can either be created in one swoop at the beginning of the compression or it could be built and adaptively as the algorithm processes the file. The algorithms succeed because a pointer describing the position in the dictionary takes up much less space than the common word itself. The dictionary is just a list of words. It is almost always 2n words because that makes the pointer to a particular word take up n bits. Each word can either be a fixed length or a flexible length. Fixed lengths are easier to handle, but flexible length do a better job of approximating the English and x86 machine code.

5.1. Generate Dictionary Words Algorithm [8]

Input: text file.

Output: dictionary file.

Processing:

Step-1: The file is analyzed to create a list of the 2n most common words.

Step-2: The file is processed by scanning from beginning to end.

Step-3: If the current word is in the dictionary then it is replaced by a tag, <indict>, followed by the position in the dictionary.

Step-4: If it isn't in the dictionary, then it is replaced by a tag, <Verbatim>, followed by the word that remains unchanged.

Step-5: end.

The success of the algorithm depends on the size of the tags (<indict> and<verbatim>), the size of the dictionary, and the number of times something is found in the dictionary. One basic and usually of the byte B is zero, then the next n bits represents a word in the dictionary. If the value of the byte B is greater than zero, then B bytes are copied verbatim out of the original file. This scheme allows the program to use flexible word size that work well with English.

6. The Proposed Elliptic Curve Cryptography Protocol

In this section introduced new cryptography protocol for massages transfer between the sender and receiver by using algebraic description for addition operation over elliptic curve $\boldsymbol{Z_p}$ by generate table contain all possible points coordinates between (0 to p-1) at first stage. Then linked the generated table with the dictionary (using the algorithm that illustrated in section- 4) that contains all popular secure massages that used between the sender and receiver. The proposed protocol work as conventional cryptography system (symmetric key) and keep the key which represent as *(p, a, b) and dictionary* secret between the sender and receiver and work using secure channel for communication. In the following illustrated the complete algorithm for proposed cryptography protocol.

6.1. The Proposed Algorithm for New Elliptic Curve Cryptography Protocol

The proposed algorithm work to interested for the ability that satisfy in elliptic curve addition over $\boldsymbol{Z_p}$ that each addition two points represent third point in curve i.e. P(x, y)+Q(x, y)=R (x, y) this feature work well in the proposed protocol . in other word the point R(x, y) over Ep(a,b) have many choices for addition different points for P(x, y) and Q(x, y) gives unique point R(x, y) , the redundancy for P,Q points help to send the same massages using different points for P,Q. this feature make the protocol more flexible , the proposed cryptography protocol work according the following steps:

1. **Generate dictionary for secure sentences**: Bullied dictionary that contained short secure sentences instead of word according the algorithm that illustrated in section (4-1).
2. **Generate table for Elliptic curve points P(x, y) over Ep(a,b)**: The sender and receiver agreed between them about for the secret key information that represent the following:.
   - The Elliptic curve group that used Ep(a, b): the sender and receiver agreed about the prime number that used (very large prime number)and the parameters .(in this paper we used small prime number just to explain how the protocol work in details)
   - Permutation key PK and PK-1: the sender and receiver agreed about the permutation key (PK) and inverse permutation (PK-1) in order increased the randomizes for points sequence.
3. **Encode the plaintext secure messages *m's* as an x-y point Pm** :

Encode the massage as the x-y coordinate for point are not easy process, because not all such coordinates are in Ep(a, b), for this reason we proposed in this protocol new method for encoding the massages by combine between the dictionary sentences that generated in step-1 with the points table

that generate in step-2 as a way for encoding the massage where each sentences in the dictionary linked with unique points for R(x, y) according the rules for Elliptic curve addition .

4. **Encryption process:**
   - The sender select the secret massages that want to send from the dictionary, that generated according steps (1 and 2).
   - Convert the points R(x, y) in another form using table of points over Zp, i. e

   R(x, y) =P(x, y) +Q(x, y), by this way each point represent by two different points.
   - Used the secret PK (permutation key) for increased the randomized for points sequence of data Mi.
   - Convert the data points in to binary form.
   - Used BBS generator(that illustrated in section(2-1)) in order generate the secret key that used in encryption process :

   Mi Xor Bi = Ci

   Note: The decryption process work as the same steps using PK-1, with reveres order, i.e.
   Mi = Ci Xor Bi
5. **Sending process:**

For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key. Therefore, the strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key.

7. Implementation

In this section we implemented simple example that show the proposed algorithm using small prime number and let a=b=1 and p=11. In the following complete example that shows the proposed algorithm stages:

1. Generate Dictionary for Secure Sentences

The sender and reciver used the same dictionary according the secret information that used , where the size for dictionary rows depend on the EP(a,b), in this example used $E_{11}(1,1)$, therefore the dictionary construct from 13 row according the number for elliptic curve sets, using the schema that illustrated in section(4).in table (1) , show simple design for $E_{11}(1,1)$ dictionary.

Table 1: Simple design for $E_{11}(1, 1)$ dictionary

Index	Secure sentences
1	"I am very happy"
2	"work hard this day"
3	"this day is sunny day"
4	**"the meeting at 9 o'clock"**
5	"in the building"
6	"take another home"
7	"come back know"
8	**"good luck"**
9	"Alice she's very lazy"
10	"am reading at my home"
11	"bob buy new car"
12	"my mother she's very seed"
13	"at 7 o'clock"

2. Generate Addition Table for Elliptic Curve Points E_{11} (1,1)

In this sage generate table addition (13×13) by using the rules for addition over $E_{11}(1,1)$ correspond to the algebraic technique. In table (2) show the elliptic curve $E_{11}(1,1)$.

Table 2: Elliptic curve $E_{11}(1,1)$.

(4,6)	(8,9)	(6,5)	(3,8)	**(1,6)**	(8,2)	(0,10	(6,6)	(1,5)	(2,0)	(4,5)	0	(0,10)	**(0,1)**
(8,2)	(4,5)	(3,3)	(6,6)	(8,9)	(1,5)	(6,5)	(0,1)	**(1,6)**	(4,6)	(2,0)	(0,1)	O	**(0,10)**
(3,8)	(6,5)	(8,9)	(4,6)	(0,10)	(6,6)	(1,6)	(8,2)	(0,1)	O	(8,2)	(2,0)	(4,5)	**(1,5)**
(6,6)	(3,3)	(4,5)	(8,2)	(6,5)	(0,1)	(8,9)	(1,5)	(0,10)	(8,9)	O	(4,6)	(2,0)	**(1,6)**
(6,5)	(6,6)	(8,2)	(8,9)	(3,8)	(3,3)	(4,6)	(4,5)	(8,9)	(0,10)	(0,1)	(**1,6**)	(1,5)	**(2,0)**
(**1,6**)	(4,6)	(3,8)	(0,10)	(2,0)	(8,9)	O	(6,4)	(4,5)	(1,5)	(8,2)	(0,1)	(6,6)	**(3,3)**
(4,5)	(1,5)	(0,1)	(3,3)	(8,2)	(2,0)	(6,6)	O	(4,6)	(8,9)	**(1,6)**	(6,5)	(0,10)	**(3,8)**
(0,10)	(3,8)	(4,6)	**(1,6)**	**O**	(8,2)	(2,0)	(8,9)	(3,3)	(0,1)	(6,6)	(1,5)	(8,2)	**(4,5)**
(3,3)	(0,1)	(1,5)	(4,5)	(8,2)	**O**	(8,2)	(2,0)	(3,8)	(6,5)	(0,10)	(8,9)	**(1,6)**	**(4,6)**
(1,5)	(2,0)	O	(3,3)	(4,5)	**(1,6)**	(3,3)	(0,10)	(8,9)	(8,2)	(4,6)	(6,6)	(3,8)	**(6,5)**
(2,0)	(**1,6**)	(3,8)	O	(1,5)	(4,6)	(0,1)	(3,8)	(8,2)	(4,5)	(8,9)	(3,3)	(6,5)	**(6,6)**
O	(0,1)	(1,6)	(2,0)	(0,1)	(3,8)	(1,5)	(4,6)	(6,6)	(3,3)	(5,6)	(4,5)	(8,9)	**(8,2)**
(0,10)	O	(2,0)	(1,5)	(3,3)	(0,10)	(4,5)	**(1,6)**	(6,5)	(6,6)	(3,8)	(8,2)	(4,6)	**(8,9)**

3. Encode the Plaintext Secure Messages *m's* as an x-y Point P_m.

Table 3: Encoding process

Index	E11(1,1)	Secure sentences
1	**(0,1)**	" I am very happy"
2	**(0,10)**	" work hard this day"
3	**(1,5)**	"this day is sunny day"
4	**(1,6)**	**" the meeting at 9 o'clock"**
5	**(2,0)**	" in the building"
6	**(3,3)**	"take another home"
7	**(3,8)**	" come back know"
8	**(4,5)**	**"good luck"**
9	**(4,6)**	"Alice she's very lazy"
10	**(6,5)**	" am reading at my home"
11	**(6,6)**	"bob buy new car"
12	**(8,2)**	" my mother she's very seed"
13	**(8,9)**	"at 7 o'clock"

Example: " the meeting at 9 oclock" encode to R(x,y)= (1,6)
"good luck" encode to R(x,y)= (4,5)

4. Encryption Process

To encrypt the massage ***"the meeting at 9 o'clock, good luck***" the encoding process convert this massage to points (1,6),(4,5) . in this stage used each point used another form for represent , i.e (1,6)=(4,6)+(0,1) or (1,6)=(8,2)+(6,6),……. There many forms that represent unique (1,6) . this feature help to send the same sentences but with another form. In this example we select (1,6)=(4,6)+(0,1) and (4,5)=(2,0)+(3,3) . Then the massage converts to

((4,6),(0,1),(2,0), (3,3)) using secret Permutation key PK let= 3 2 4 1 ➔ ((2,0),(0,1),(3,3),(4,6))➔ (010, 000),(000, 001),(011, 011),(100, 110)) ➔

M=010000000001011011100110

An example of BBS operation [8] is used. where, n = 192649 = 383 x 503 and the seed s = 101355

M=010000000001011011100110

Bi=11001110000100111010……
C=100011100000101010…….. *Cipher text send using secure channel*

5. Decryption Process

C=100011100000101010 …… *Cipher text received from secure channel*
Bi=11001110000100111010……
M=010000000001011011100110 ➔(010, 000),(000, 001),(011, 011),(100, 110)) ➔ ((2,0),(0,1),(3,3),(4,6)) ➔Pk-1= 4 2 1 3 ➔**((4,6),(0,1),(2,0), (3,3))** ➔(1,6),(4,5) ➔ Dictionary ➔ M=***"the meeting at 9 o'clock, good luck***".

8. Conclusions

From the new elliptic curve protocol, we reached to the following conclusions:

a. Solve the coding secret massage and using same length code for all transfer massages, by using addition elliptic curve algebra table. the protocol succeed to prevent the attacker be useful form the information about the letter frequency for massages and any depended information between the massages itself.
b. Use the elliptic curve addition table, work wall to find new method for transfer the same massage with different coding by useful the many addition choices between P(x, y) and Q(x, y) points that give unique R(x, y) point.
c. The proposed methods for coding used the sentences instead of the word in order reduce the code length to make the system more secure.
d. The proposed protocol succeeds to produce elliptic curve as one key cryptography system instead of two-key system.
e. The proposed protocol be more secure when large prime number is used with different values for a, b parameters over EP (a,b).

References

[1] Kristin Lauter, "***The Advantages of Elliptic Curve Cryptography for Wireless Security***", Microsoft Corporation, IEEE Wireless Communications , February 2004.
[2] Murat Fiskiran," ***Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments"***, Proc. IEEE Intl. Workshop on Workload Characterization, pp: 127-137, 2002.
[3] De Win, E. and B. Preneel, ***"Elliptic curve public-key cryptosystems - an introduction. State of the Art in Applied Cryptography"***, LNCS 1528, pp: 131-1411,1998.
[4] Aydos, M., E. Savas and C.K. KoV, "***Implementing network security protocols based on elliptic curve cryptography***". Proc. fourth Symp. Computer Networks, pp: 130-139,1999.
[5] Fernandez, A. "***Elliptic Curve Cryptography***." Dr. Dobb's Journal, December ,1999.
[6] Auerbach ,***"Handbook on Local Area Network***s", published in September ,1998.
[7] BLUM86 Blum, L.; Blum, M.; and Shub, M. "A ***Simple Unpredictable Pseudo-Random Number Generator."*** SIAM Journal on Computing, No. 2, 1986.
[8] Peter Wayner," **Disappearing *Cryptography: Information Hiding: Steganography & Watermarking"***, 3 Edition, 2008 .
[9] William Stallings, "**Cryptography and network security**", 2nd edition, Prentice Hall publications.
[10] Kolhekar & Jadhav, "**IMPLEMENTATION OF ELLIPTIC CURVE CRYPTOGRAPHY ON TEXT AND IMAGE** "International Journal of Enterprise Computing and Business Systems ISSN (Online) : 2230-8849 http://www.ijecbs.com,Vol. 1 Issue 2 July 2011