E - Commerce for Future & Trends

Volume 1, Issue 1
www.stmjournals.com

Efficient Computation of Point Multiplication in the Implementation of Elliptic Curve Cryptography

Bh . Padma*, D.Chandravathi
Department of Computer Applications, GVP College for Degree and PG Courses, Yendada, Rushikonda, Visakhapatnam-45, India

Abstract

In recent times, elliptical curves have been used for public-key cryptographic and signature schemes in information security. The security provided by elliptic curve cryptographic protocols, either using signature or public-key encryption, is entirely based on the discrete logarithm problem. The discrete logarithm problem says that for a given a point Q that is a certain multiple k of a fixed point P, how to find the value of k in a reasonably amount of time. The difficulty of the discrete logarithm problem can only be exploited if scalar multiplications are easy to obtain. But fortunately, a point multiplication can always be computed in linear time, nonetheless this operation needs to be optimized as much as possible. Because of there is much concentration on reducing the speed of the scalar multiplication, several methods have been developed for improvisation. This paper describes implementations and test results of Elliptic Curve Cryptography (ECC). The paper deals with the problem of improving the performance of point multiplication using Binary method and Addition - Subtraction method. These methods reduce the number of point doublings and point additions in the computation. Also this paper insists the application of these two proposed methods for point multiplication.

Keywords**:** *Encryption/decryption, elliptic curve cryptography, elliptic curves, discrete logarithmic problems*

****Author for Correspondence*** E-mail: padma.bhogaraju@gmail.com

INTRODUCTION

Elliptic Curve Cryptosystem was proposed by Miller and Koblitz. The complexity of the system entirely based on the difficulty of elliptic curve discrete logarithm problem (ECDLP)[1]. Most of the e products and standards use RSA for public key encryption. But to get the desired amount of security using RSA , it is needed to increase the key size which causes a heavier processing load on applications that are using RSA. Now a days Elliptic curve Cryptography (ECC) has begun to challenge RSA.

The main attraction of ECC compared to RSA is that it offers equal security for a smaller bit size, and accordingly reduces the processing overhead. Another attractive feature of ECC is there is a possibility of optimizing the arithmetic operations in the underlying binary or primary field. Now ECC is very popular for many information security applications [2].

Elliptic Curve Cryptography (ECC) is a public key technology that offers performance advantages at higher security levels cryptography. In public key cryptography every user taking part will take a pair of keys, a public key and a private key. The private key is known only to the one who owns it and the public keys are distributed to all users taking part in the communication [3].

ELLIPTIC CURVE CRYPTOGRAPHY

In public key cryptographic algorithms, there always exists a set of parameters or a set of predefined constants which are known by all the devices taking part in the communication [4]. In ECC we call these predefined constants as ‘Domain parameters’. All the

mathematical operations for encryption and decryption of ECC is defined over the elliptic curve $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$[5]. When the values of a 'a' and 'b' changes it gives a different elliptic curve.

The set of all the points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the elliptic curve and the private key is a random number generated by the sender or receiver. The sender and receiver generate the public keys by multiplying the private key with the generator point G in the curve[6].

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. So the dominating operation involved in Elliptic curve cryptography is point multiplication. i.e. multiplying a scalar k with any point P that lie on the curve to obtain another point Q on the elliptic curve.

The most important computation in elliptic curve cryptography is the scalar multiplication using a large integer *k*. In elliptic curve cryptosystems, the process of scalar multiplication is involved in main operations such as key agreement, signing/verifying and also encryption and decryption. The speed of scalar multiplication only determines the efficiency of the crypto-system [7].

The Addition-Subtraction Method not only reduces the total number of point addition, but also reduces the number of point doublings [8]. The observations that are found while using both the methods for encrypting and decrypting a file have shown that the Addition-Subtraction method is faster than the widely used original method.

SCALAR MULTIPLICATION

Let us say P is a point on the elliptic curve , and say k is an integer , now the scalar multiplication is finding another point on the curve that is obtained by finding kP = P+P+P+.....+ k times.

The point P is a fixed point that generates a large, prime order subgroup. It is also known as point multiplication and dominates the execution times of elliptic curve cryptographic scheme. This is the original method we generally use while implementing ECC [9].

BINARY METHOD

Elliptic curve cryptosystem will be the cryptosystem for the future. One method to improve the performance of this cryptosystem is to use an efficient and novel method for point multiplication that is Binary Method [10]. The Binary Method considers the binary representation of k[11] for the computation of kP.

$$\text{i.e., } k = \sum_{j=0}^{l-1} k_j \, 2^j \text{ where each } k_j \in \{0, 1\}.$$

then kP can be computed as

$$\text{i.e } kP = \sum_{j=0} k_j \; 2^j * P$$

$$kP = 2(\ldots..2(2k_{l-1} P + k_{l-2} P)+\ldots..)+ k_0 P$$

As for example let k=15.

Binary representation of K is $(1111)_2$.

Thus Q=15P can be obtained as

Q = 15P = 2*(2*(2*P+P) +P) +P.

similarly the others

Q = 22P = 2* (2* (2* (2*P) +P) +P). - $(10110)_2$

Q=45P= 2* (2* (2* (2* (2*P)+P)+P)+P- $(101101)_2$

Q=63P=2* (2* (2* (2* (2*P+P) +P) +P) +P) +P -

$(111111)_2$.

ADDITION-SUBTRACTION METHOD

An improved method for computing kP can be obtained from the following facts. Let us say k is an integer then k has a unique representation of the form $\sum k_j 2^j$, where each j belongs to { -1,0,+1}, such that no two consecutive digit are nonzero. This representation of any integer is known as non-adjacent form (NAF) [1, 12]. The computation of the negation of a point P=(x,y) takes very less amount of time, and the cost of addition and subtraction will be the same and hence this method does not need extra amount of processing time.

Algorithm for Conversion into NAF Form

1. input n
2. output $U = u_m u_{m-1} \ldots\ldots u_1 u_0$ (NAF form)
3. i = 0
4. while n > 0 do
5. if n is odd then $u_i = -1$
6. else $u_i = 0$
7. $n = (n - u_i)/2$
8. i = i + 1
9. return U

Example

Suppose k=63, at first c=63. Then, by this method firstly we get $u_0 = -1$.(as c is odd). Then, by setting $c = c - u_0$, c becomes even. Then u_1 becomes 0 . The iteration goes on as is specified in the algorithm (with $u_1=0$, $u_2=0$, $u_3=0$, $u_4=0$, $u_5=0$, $u_6=1$) and then finally we get NAF(63)= (100000-1).

Say for example, NAF(23)=10-100-1.

The addition-subtraction method, performs an addition or subtraction depending on the sign of digit of non-adjacent form (NAF) of the number[9]. This algorithm requires doubling and addition on an average. Addition-Subtraction method is also called as Window Method[13] .

Algorithm for addition-subtraction method

Step 1. input $k = \{k_n, k_{n-1}, .., k_0\}$
Step 2.Let y = 1and z = 1
Step 3.for i from length(k) -1 to 0 step-down by 1 DO
Step 4. y = y + y
Step 5. if $k_i = 1$, then
Step 6. y = y + z
Step 7. if $k_i = -1$, then
Step 8. y = y – z
Step 9. return : y

According to the above algorithm ,
NAF(23)=10-100-1, NAF(63)=100000-1, So
23P = 2* (2*(2*(2*(2*P) -P))) –P
63P = 2*(2* (2* (2* (2* (2*P))))) –P

Comparison between the Above Two Optimization Methods

The comparison of CPU Times for encryption times and decryption times in both the scalar multiplication methods with elliptic curve parameters: p=2011, a=9, b=7, G= (2010, 1600) was recorded. The encryption and decryption times are specific to the processor. The above observations are recorded on a machine with 1GB RAM and 1.6 GHz processor speed on Win XP platform. Comparison of CPU time in General Method and Addition Subtraction Method for Scalar Multiplication in ECC (for fixed values of parameters: private key of receiver is 35 and k=1000) for different file sizes.

The graphs below represents the variations in the encryption and decryption times for Binary method, the Addition-Subtraction method and the general method [14] Table 1 shows the CPU time for the encryption times when using the three methods for scalar multiplication.

Table 2 shows the CPU time for the decryption times when using the three methods for scalar multiplication.

***Table 1**: Encryption Times.*

File Size	Addition–Subtraction Method (CPU time in sec)	Binary Method (CPU time in sec)	General Method (CPU time in sec)
1K	35.36	51.30	117.84
2K	66.39	102.85	237.09
3K	106.04	153.49	352.84
4K	132.65	204.47	473.68
5K	163.44	255.69	591.11

Table 2: *Decryption Times.*

File Size	Addition–Subtraction Method	Binary Method	General Method
1K	0.375	0.76	1.25
2K	0.750	1.52	2.56
3K	0.970	2.25	3.88
4K	1.500	3.01	5.17
5K	1.95	3.77	6.45

The efficiency in the computation of the point multiplication using the three methods is compared in Figure 1.

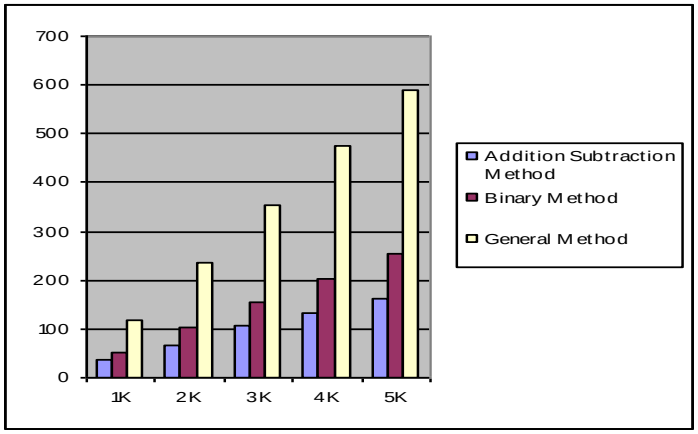

Fig. 1: *Variations in the Encryption Times.*

The following graph represents variations in the decryption times.

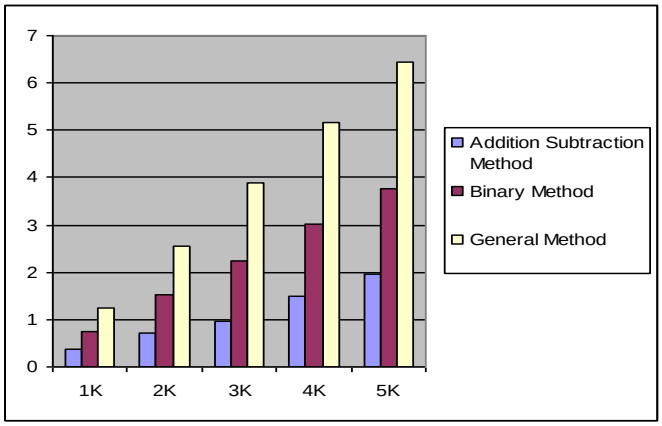

Fig. 2: *Variations in the Decryption Times.*

CONCLUSION

For efficient implementation of ECC, it is desired that the point multiplication algorithm under the underlying field arithmetic either by using binary field or prime field, should be efficient. ECC implementation using the proposed methods i.e., (Addition Subtraction Method and Binary method), on point multiplication has shown considerable improvement in the efficiency of the scalar multiplication compared to the original implementation. This paper concludes that Addition-Subtraction method shows considerable performance compared to the Binary and the General method for point multiplication for ECC.

REFERENCES

1. Lopez J., Dahab R. (2000). An overview of elliptic curve cryptography. *Technical report, IC-00-10, May 22. Available at http:// www.dcc.unicamp.br/ic-main/publication - e.html*.
2. Bhandari, A. K.; Nagraj, D.S.; Ramkrishna, B.; Venkataramana, T. N. (editors). Elliptic Curves, Modular Forms and Cryptography. *New Delhi, India: Hindustan Book Agency,* 2003E..
3. Dummit, David S. and Foote, Richard M. Abstract Algebra. *New York, NY: John Wiley and Sons, Inc.*, 1999.
4. Stalling W. Cryptography andNetwork Security. *Prentice Hall, New Jersey, USA, Third Edition, Chapter 10.*.
5. 2003Solinas J. Efficient arithmetic on Koblitz curves. *Designs, odes and Cryptography*.2000; 19: 195–249p .
6. Miller V.S. Use of elliptic curves in cryptography. *Advances in Cryptology, Proceedings of CRYPTO'85, LNCS, Springer-Verlag*. 1986; 218: 417–426p.
7. Koblitz N. Elliptic curve cryptosystem. *Mathematics of Computation*; 48 (1987): 203–209p.
8. Rivest R.L., Shamir A., Adleman L.M. A Method for obtaining digital signatures and public key cryptosystem. *Communications of the ACM*; 21(1978): 120–126p.
9. Elliptic Curve Cryptography. Standards for Efficient Cryptography Group,. *Working Draft. Available from: http://www.secg.org/ September,*.
10. Md. Rafiqul Islam et al. A New Point Multiplication Method for Elliptic Curve Cryptography Using Modified Base Representation. *International Journal of The Computer, the Internet and Management*. 2008; 16(2).
11. 2000Standard Specifications for Public Key Cryptography, *IEEE Standard 1363*..
12. Al-Daoud et al. A new addition formula for Elliptic curve over GF (2n). *IEEE Transactions on Computers*. 2002; 51(8):972–975p .
13. 2000Diffie W., Hellman M.E. New directions in cryptography. *IEEE Transactions on Information* Theory. 1976; 22 (6): 644– 654p .
14. Menezes A.J., Vanstone S. A. Elliptic curve cryptosystem and their implementations. *Journal of Cryptology*.1993; 6(4): 209–224p .

