I. J. Computer Network and Information Security, **2016, 4, 56-61**
Published Online April 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.04.07

Efficient Scalar Multiplication over Elliptic Curve

Deepika Kamboj
Dept. of Computer Science and Engineering
Dr. B R Ambedkar NIT Jalandhar
E-mail: Er.deepikakamboj@gmail.com

Assoc. Prof. D.K.Gupta
Dept. of Computer Science and Engineering
Dr. B R Ambedkar NIT Jalandhar
E-mail: guptadk@nitj.ac.in

Asst. Prof. Amit Kumar
Dept. of Computer Science and Engineering
Dr. B R Ambedkar NIT Jalandhar
E-mail: amitkumar62003@gmail.com

Abstract—Elliptic Curve Scalar multiplication is the method of adding a point on the curve to itself every time[1]. In recent years, research in Scalar Multiplication over Elliptic curves (EC) over *a finite fields* attracted many researchers, working in field of cryptography, to find out how elliptic curves cryptography (ECC) can be implemented and how to reduce its complexity [4]. The efficient techniques used in Elliptic curve cryptography are Elliptic curve scalar multiplication using *point-halving algorithm* [2], then *double-base (DB) chain algorithm*, and after that *step multi-base representation (SMBR),* but these techniques have their drawbacks. So, it become imperative to find out a new approach which can be efficiently used for implementation of ECC and further reducing its complexity. The paper proposes a new algorithm *Treble algorithm for affine coordinates.* We continued doing work using the binary concept or *double and add* operation with the help of treble approach to make it more efficient which relates to the use of all input values in producing any type of output, including how much time and energy are required The results show that our contribution can significantly enhance EC scalar multiplication.

Index Term—Elliptic curve cryptography, double and add operation, affine coordinates.

I. Introduction

Elliptic Curve Cryptography (ECC) was proposed as an alternative mechanism for implementing public- key cryptography in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) [1]. Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic protocols based on algorithms that needs two separate keys, out of these two, one is secret (or private) and another of public. Elliptic Curve Cryptography is based on discrete logarithms that are more computationally expensive to invert at equivalent key lengths [8]. For general elliptic curves, we will present an improved version of scalar multiplication. We'll present a treble concept as well with its basic operations doubling and addition. Treble method provides fast computation as compare to doubling and addition. Its computation is effective than first applying doubling and after that addition. Besides that we implemented a recursive approach as well where time complexity is reduced but number of functions calling in both approaches remain same but less than the existing approach. The primary advantage promised by ECC is a smaller key size, reducing storage and transmission requirements, i.e. an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key – eg, a 384- bit ECC public key should provide comparable security to a 3072-bit RSA public key.

As the size of key increases, the difference in equivalent key size also increases. Modular arithmetic used in DH provides less security as compare to algebraic curves used in Elliptic curve cryptography. The approximate equivalence in security strength for RSA, DH and Elliptic curve cryptography is given in tables 1 and 2 shows that EC is better to use if key size is large.

Table 1. Key Strength of Elliptic Curve

Key length in bits	**DH security and EC security ratio**
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Table 2. NIST Recommended Key Sizes

Symmetric Key length in bits	Key length of RSA and DH in bits	Key length of EC in bits
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

II. Mathematics of Elliptic Curve Cryptography

We can define Elliptic curves over both prime field and binary field. If P is a prime number or prime power, then Fp will denote the field that has exactly P points on the curve [6]. When the greatest common divisor of P and b is 1, the elliptic curve over the field Fpis expressed by the following equation:

$$E: Y^2 = X^3 + aX + b \tag{1}$$

Whereas, b are the coefficients in Fp and $4a^3 + 27b^2 \neq 0$. While the general curve equation is defined as:

$$E: Y + XY = X + aX + b \tag{2}$$

Coordinates (X, Y) are used to define the affine coordinates which satisfy the curve equation. Coordinates for adding two different points

$$X3 = A^2 – X1 – X2 \bmod P \tag{3}$$

$$Y3 = A(X1 – X3) – Y1 \bmod P \tag{4}$$

Where

$$A = (Y2-Y1)/(X2-X1) \bmod P \tag{5}$$

Coordinates for doubling a point:

$$X4 = A^2 – 2X1 \bmod P \tag{6}$$

$$Y4 = A(X1 – X4) – Y1 \bmod P \tag{7}$$

$$\text{Where } A = (3X1^2 + a)/2Y1 \tag{8}$$

III. Related Work

Many researchers presented different works related to the elliptic curve scalar multiplication in Cryptography. Some of this work is summarized below in form of table with their specified problems.

Table 3. Literature Work with Specified Problem

Literature methods	Specified Problems
Elliptic curve Diffie Hellman	Highly computation-intensive nature of its underlying cryptographic operations, causing long execution times and high energy consumption
Lem-Lee technique for scalar multiplication	Computing k1 ** (2192 x P) ++ k2 ** P all at the same time using the double and add process. Thus computation is expensive.
Public key cryptography	Data throughput rates of most popular public-key encryption methods are several orders slower
Halve-and-add scalar multiplication	The point halving method is faster than the doubling method if it is implemented using affine coordinates
Parallel Scalar Multiplication with Pre- computation	The only drawback is the energy consumption since nodes have to communicate with each other for task distribution and result retrieval
Elliptic Curve Scalar Multiplication with a Bijective Transform	mapping of points is required
Non-Adjacent Form (w-NAF) method	Drawback of wNAF is that it is not possible to merge the exponent recoding and the evaluation stage and it seems impossible to compute wNAF left-to-right. However, in connection with memory constraint devices left-to-righter-coding schemes are by far more valuable

IV. Binary Method Scalar Multiplication

In Scalar Multiplication operation there is a point P and we are interested in computing k multiplied by P [6]. The value of k is written in binary format call it k_0, $k_1$and till k_{i-1}. So, k_i is the first time when 1 is encountered; previous to that, everything was 0.

```
Require $k = (k_{i-1}, k_{i-2}, \ldots, k_0)_2$, $k_i = 1$
Compute Q=kP

Q(X,Y)    ECC_Point(X,Y)
For M = i-2 to 0
    Q(X,Y)    ECC_double(X,Y)
    if $k_m = 1$ then
        Q(X,Y) = ECC_add(Q(X,Y), P(X,Y))
    end if
end for
Return Q
```

V. Proposed Method

This research aims is to find the scalar multiplication taking less time than basic addition and doubling. our operations is more economical than all the previously proposed ones. The new treble concept is applied here on affine coordinates. The proposed algorithm is faster than the previously known results. Our approach reduces lower complexity which contributes to an efficient implementation of elliptic curve cryptosystems. The fast implementation of elliptic curve cryptosystems relies on the efficient computation of scalar multiplication. This algorithm is based on the base 3 concept where value of k is derived in base 3. After that based upon the most significant bit, we decide whether we should double the point or triple that point. The algorithm is based off of an operation that allowed for the tripling of a point and is derived from point doubling and addition operations. Inspite of one doubling operation and one addition operation in calculating the 3P we can directly calculate the triple of that point. The purpose of our research is to takes account of speedups such as A − M tradeoffs where A and M is Addition and Multiplication respectively. Optimized pre computations and treble method save time for single-scalar multiplication in affine coordinates. Our problem of formula optimization is to reduce the number of field operations to compute point multiplication. The fact that tripling a point is cheaper than a double and add using our techniques suggests using the operation of tripling more often while performing scalar multiplication of a point on an elliptic curve.

- Weierstrass equation is given as input through which we can find out the points on the elliptic curve.
- The points will be calculated based upon the value of the prime field so that we can have the finite number of points.
- A curve going through these points will be drawn so that we can show its cubic nature as well means if we draw a line going through two points then it will intersect the curve at the third point.
- Scalar value i.e. value of key ‘k’ provided by user will be converted in base 3 form to make this scalar multiplication more efficient as compare to the earlier one.
- Based upon the most significant bit we complete our scenario whether to use the doubling operation, or tripling operation, or addition.

Points for addition:

$$X_3 = A^2 – X_1 – X_2 \tag{9}$$

$$Y_3 = A(X_1 – X_3) – Y_1 \tag{10}$$

Points for doubling:

$$X_3 = A^2 – 2X_1 \tag{11}$$

$$Y_3 = A(X_1 – X_3) – Y_1 \tag{12}$$

Points for tripling:

$$X_4 = (A_2 - A_1)(A_1 + A_2) + X_1 \tag{13}$$

$$Y_4 = (X_1 - X_4) A_2 - Y_1 \tag{14}$$

The line going through points (X1, Y1) and (X2, Y2) will intersect the curve at the third point which will give the (X3, Y3). After getting all points within the finite field we can use any of the point as the base point and can find any other point existing over the curve in a very less time so that encryption and decryption can take place in efficient manner.

Proposed algorithm

Scalar Multiplication Using proposed algorithm

If k_i = 1

```
Require $k=(k_i,k_{i-1},k_{i-2},\ldots,k_0)_3$, $k_i=1$
Compute Q=kP
ECC_Point Q(X,Y) =ECC_ point P(X,Y)
For M = i-1 to 0
    If $k_m$ = 0 then
        ECC_ point Q(X,Y) = ECC_triple Q(X,Y)
    Else if $k_m$ = 1 then
        ECC_point Q(X,Y) = ECC_add(ECC_triple Q(X,Y), ECC_point Q(X,Y))
    Else if $k_m$ = 2 then
        ECC_point Q(X,Y)=ECC_add(ECC_triple X,Y), ECC_doubleQ(X,Y))
    End if
    End if
    End if
End for
Return Q
```

If k_i = 2

```
Require $k=(k_i,k_{i-1},k_{i-2},\ldots,k_0)_3$, $k_i=2$
Compute Q=kP
ECC_Point Q(X,Y) =ECC_Double P(X, Y)
For M = i-1 to 0
    If $k_m$ = 0 then
        ECC_point Q(X, Y) = ECC_triple (ECC_doubleQ (X, Y))
    Else if $k_m$ = 1 then
        ECC_point Q(X, Y) = ECC_add (ECC_triple (ECC_ Double Q(X, Y)) ECC_point Q(X, Y))
    Else if $k_m$ = 2 then
        ECC_point Q(X, Y) = ECC_Add (ECC_triple (ECC_double Q(X, Y)), ECC_double (ECC_ Point Q(X,Y)))
    End if
    End if
    End if
End for
Return Q
```

Treble method reduces the cost of computation as in spite of calculating double and addition sometimes it can work alone. The below table shows how efficient our approach is:

Table 4. Differentiation between Binary and Proposed Method Based Upon Function Calling

key value k	Total function call in bin- ary approach	Total funct- ion call in tre approach
100	8	6
200	9	8
250	12	7
300	11	7

Example: Let Values of coefficients a, b and prime field p are respectively 2, 2, 17.

Then using Weierstrass equation Points on elliptic curve: (0, 6), (0, 11), (3, 1), (3, 16), (5, 1), (5, 16), (6, 3), (6, 14), (7,6), (7,11), (9, 1), (9, 16), (10, 6), (10, 11), (13, 7), (13, 10), (16, 4), (16, 13)

Let base point is selected: (5, 1)
And private key k is: 71
Now k is converted in base 3 forms as: 2122

Step 1: MSB is 2, hence doubling of (5, 1) is performed and get the point (6, 3)

Step 2: Next digit is 1, hence first we performed tripling of (6, 3) and then we added (5, 1) to it and thus we get (0, 6).

Step 3: Now we have digit 2, hence we performed tripling of (0, 6) and added double of (5, 1) i.e. (6, 3). And thus we get point (3, 1).

Step 4: Again for digit 2 we performed tripling of (3, 1) and added (6, 3) to it and finally we get point (9, 1).

VI. Performance of Proposed Algorithm

Figure 1 shows the performance between binary and proposed approach with respect to the value of k and number of function calling.

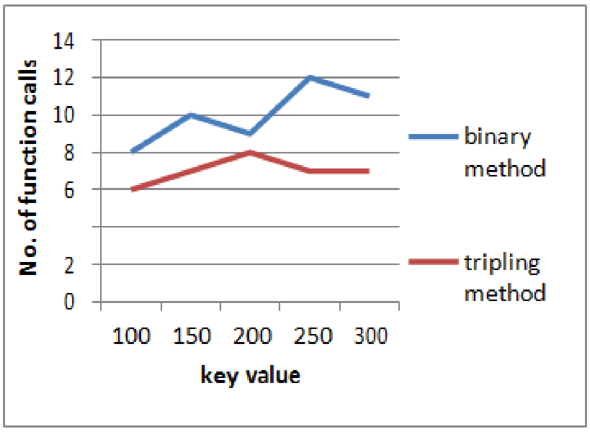
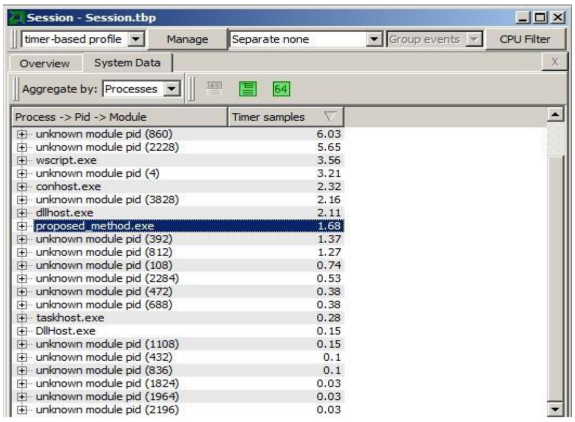


Fig.1. Graph Showing Function Call Required For Binary and Proposed Method with Respect to Key Value

This graph shows that the performance of proposed method is better than that of binary method. A profiling tool AMD code analyst is used to profile the code.

In AMD code analyst we used time based profiler, which shows the time taken by both approaches. Proposed_method.exe is our new approach file which is taking 1.68 milliseconds.

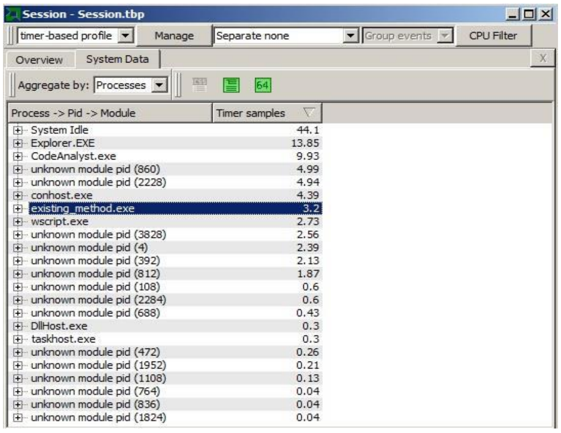


timer-based profile
Separate none
Aggregate by: Processes

Process -> Pid -> Module	Timer samples
unknown module pid (860)	6.03
unknown module pid (2228)	5.65
wscript.exe	3.56
unknown module pid (4)	3.21
conhost.exe	2.32
unknown module pid (3828)	2.16
dllhost.exe	2.11
proposed_method.exe	1.68
unknown module pid (392)	1.37
unknown module pid (812)	1.27
unknown module pid (108)	0.74
unknown module pid (2284)	0.53
unknown module pid (472)	0.38
unknown module pid (688)	0.38
taskhost.exe	0.28
DllHost.exe	0.15
unknown module pid (1108)	0.15
unknown module pid (432)	0.1
unknown module pid (836)	0.1
unknown module pid (1824)	0.03
unknown module pid (1964)	0.03
unknown module pid (2196)	0.03

Fig.2. Profiling of Proposed Method using AMD Code Analyst

Whereas existing_methods.exe is binary approach file which is taking 3.2 milliseconds for same data. This profiling tool helps in differentiating the time taken between these two approaches.



timer-based profile
Separate none
Aggregate by: Processes

Process -> Pid -> Module	Timer samples
System Idle	44.1
Explorer.EXE	13.85
CodeAnalyst.exe	9.93
unknown module pid (860)	4.99
unknown module pid (2228)	4.94
conhost.exe	4.39
existing_method.exe	3.2
wscript.exe	2.73
unknown module pid (3828)	2.56
unknown module pid (4)	2.39
unknown module pid (392)	2.13
unknown module pid (812)	1.87
unknown module pid (108)	0.6
unknown module pid (2284)	0.6
unknown module pid (688)	0.43
DllHost.exe	0.3
taskhost.exe	0.3
unknown module pid (472)	0.26
unknown module pid (1952)	0.21
unknown module pid (1108)	0.13
unknown module pid (764)	0.04
unknown module pid (836)	0.04
unknown module pid (1824)	0.04

Fig.3. Profiling of Existing Method using AMD Code Analyst

VII. Second Approach to Reduce Complexity

To reduce time complexity of our proposed algorithm, we introduced a different approach again, where we worked on taking bits in multiple of 2 rather than one bit at a time. Means on first time we took one bit, then 2 bits, then 4 bits and so on. Thus this procedure reduces the amount of time taken by an algorithm to run as a function of the length of the string representing the input. This algorithm works in following manner:

120001210110102001121010120

- Here in first step we have taken only one bit that is 2.
- Then we worked on two bits at a time i.e. on 00.
- After that to work on 4 bits at a time we applied a condition here for more than 2 bits we will keep on calling our scalar function.

Thus our new approach gives the complexity O (log n) in spite of O (n) which is really very less. At each step we move on to just twice position of the present one which really makes it very efficient for further point computation. The following graph in figure 4 shows the time complexity of both algorithms.

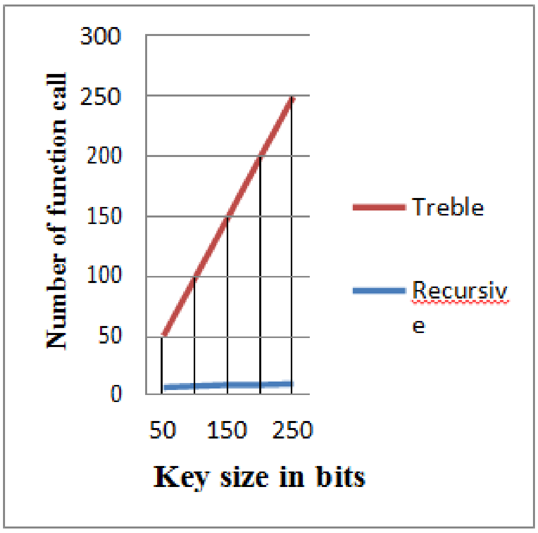


Fig.4. Time Complexity Graph between Treble Method and Recursive Approach

VIII. Conclusion and Future Work

In this paper, the research undertaken for efficient scalar multiplication over elliptic curve presents a novel approach which is effective and economical than previously proposed ones and may offer better performance. The paper presented new algorithm *Treble algorithm for affine coordinates.* The first approach for scalar multiplication with less number of function calling helps us in getting result in less time but with complexity of O(n) whereas our second approach works well when number of bits is very large because it gives O(log n) complexity.

Elliptic Curve Scalar Multiplication is diverse aspect of Cryptography. The work done in this research can be further extended by conducting research in areas mentioned below;

- The proposed alogorithim can be used to apply on on jacobian or projective coordinates.
- Calculating the tripling points in this algorithm is complicated. In future, a simple approach can be determined for computing these points
- Its complexity can be reduced by log(n) using some other concept than recursive calling because recursive call uses a large amount of memory.

References

[1] Nagaraja Shylashree and Venugopalachar Sridhar, Hardware Realization of Fast Multi-Scalar Elliptic Curve Point Multiplication by Reducing the Hamming Weights OverGF(p),IJCNIS,2014.

[2] E. Knudsen, Elliptic scalar multiplication using point halving. Advances in Cryptology— ASIACRYPT 99, Lecture Notes in Computer Science, 1999.

[3] P. K. Mishra, V. S. Dimitrov. Efficient Quintuple Formulas for Elliptic Curves and efficient Scalar Multiplication Using Multibase Number Representation. Springer-Verlag, 2007.

[4] Mohamed Lehsaini, Chifaa Tabet Hellel, “Improvement of Scalar Multiplication Time for Elliptic Curve Cryptosystems”, 2013 IEEE.

[5] J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary. Wireless sensor network security: A survey. Security in distributed, grid, mobile, and pervasive computing, 2007.

[6] N. Koblitz. Elliptic curve cryptosystems. Mathematics of computation, 1987.

[7] Ram RatanAhirwal and ManojAhke, “Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Securing Hypertext Information on Wide Area Network”, International Journal of Computer Science and Information Technologies, 2013.

[8] D.Hakerson, A. Menezes, and S. Vanston “Guide to Elliptic Curve Cryptography,” Springer-Verlag, (2004).

[9] Ch. Suneetha, D. Sravana Kumar and A. Chandrasekhar, “Secure key transport in symmetric cryptographic protocols using elliptic Curves over finite fields,” International Journal of Computer Applications, November 2011.

[10] D. Sravana Kumar Ch. Suneetha A. Chandrasekhar, “encryption of data using elliptic curve over finite fields”, international journal of distributed and parallel systems, January 2012.

[11] Ch. Suneetha, D. Sravana Kumar and A. Chandrasekhar, “Secure key transport in symmetric cryptographic protocols using elliptic curves over finite fields”, International Journal of Computer Applications, November 2011.

[12] William Stallings, “A text book of Cryptography and Network security”, Principles and practices, Pearson education, fourth edition, 2007.

[13] Wasim A Al-Hamdani, Ph.D., “Elliptic Curve for Data protection”, Kentucky State University 400 East Main, KY 40601 USA.

[14] Cohen, H., Miyaji, A., & Ono, T. (1998). Efficient elliptic curve exponentiation using mixed coordinates. In Advances in Cryptology— ASIACRYPT’98 (pp. 51-65). Springer Berlin/Heidelberg.

[15] Yoshitaka Nagai, Masaaki Shirase and Tetsuya Izu, “Elliptic Curve Scalar Multiplication with a Bijective Transform”, 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.

Authors’ Profiles

Deepika Kamboj was born in 1991 in India. She completed her B.tech from UPTU in Computer Science 2008-12 and currently pursueing M.tech from NIT Jalandhar 2013-15. Her area of interest is cryptography.

Deepak Kumar Gupta, working as an associate professor in National Institute of Technology Jalandhar, received his B.Tech degree from Gulbarga University, Karnataka. He competed his master's degree in computer science and engineering from National Institute of Technology Jalandhar. His research area include Computer Network and Operating System.

Amit Kumar Dogra, working as assistant professor @ Dr. B. R. Ambedkar National Institute of Technology Jalandhar, received his B.Tech degree from university of Jammu. He completed his master's degree in Computer Science and Engineering from Shri Mata Vaishno Devi University, Katra. His research area include Wireless Adhoc and Sensor Networks, Network Security.

How to cite this paper: Deepika Kamboj, D.K.Gupta, Amit Kumar,"Efficient Scalar Multiplication over Elliptic Curve", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.4, pp.56-61, 2016.DOI: 10.5815/ijcnis.2016.04.07