ISSN:0975-9646

Kirti Chawla et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3159 - 3164

Analysis of Multibase Scalar Point Multiplication Scheme in ECC

Kirti Chawla[1], Om Pal Yadav[2]

[1]*M-Tech student, CDAC-Noida affailated to GGSIPU Delhi*
[2] *Sr. Tech. Officer in CDAC- Noida*

Abstract: Development and research in cryptography has shown that RSA and Diffie-Hellman has is becoming more and more unsafe and Elliptic curve Cryptography is becoming a new trend in future for public key cryptosystem. The safety level of ECC with small size key is same as that of earlier cryptosystem with large size key. In this paper Nicolas Meloni's,2 2012 springer algorithm for addition of points on elliptic curve is combined with multibase concept and set generation given in "on-the-fly multi-base recoding for ecc scalar multiplication without pre-computations" by thomas chabrier and arnaud tisserand to improve the speed of the scalar multiplication.In this paper by combining the multibase and Zeckendorf concept number of multiplications and squarings are reduced on the cost of addition. Comparative analysis of proposed algorithm and some previous approaches is also discussed in last section.

Index Terms— Elliptic curve, Public Key Cryptosystem, Scalar Point multiplication, Zeckendorf representation

1 INTRODUCTION

Elliptic curve Cryptography was first introduced by Neal Koblitz and Victor Miller independently in 1985 their papers [1] and [2]. These years, research was done to improve the efficiency of ECC by improving the efficiency of scalar point multiplication which is main operation in ECC. Scalar point multiplication means computing the point nP=P+P+…+P (n times), where n is a positive integer called scalar and P is a point on elliptic curve .Elliptic Curve Cryptography has made the great progress in field of cryptography and public key cryptosystems. In ECC we use points on elliptic curve public keys [19]. It is based on scalar point multiplication instead of multiplication of large prime numbers.The key length of ECC is small as compared to RSA for same level of security. In section 2 preliminaries are discussed. In section 3 some related work is discussed. In section 4 proposed combined algorithm is discussed and in section 5 comparisons of previous approaches and proposed approach is discussed with tables and figures.

2 PRELIMINERIES

2.1 Elliptic Curve

Elliptic Curve Cryptography (ECC) is based on a finite group of points on an elliptic Curve. The equation for elliptic curve over infinite fields [8][17][18].

$y^2=x^3+ax+b$.

2.2 Point Addition in Elliptic Curve

Point addition is defined as taking two points along a curve *E* and computing where a line through them intersects the curve. We use the negative of the intersection point as the result of the addition [8][12].

The operation is denoted by P+Q=R

It can be calculated as:-

$m=y_2-y_1/x_2-x_1$

$x_3=m^2-x_1-x_2$

$y_3 = -y_1+m(x_1-x_3)$

Where x_3, y_3, x_2,y_2 x_1 ,y_1 are coordinates of R,Q,P respectively. According to formula cost of point addition is 2M+1S+1I+6AS where M is multiplication S is squaring I is inverse and AS is addition/subtraction.

2.3 Point Doubling in Elliptic Curve

Point doubling is similar to point addition, except we take the tangent of a single point and find the intersection with the tangent line. This is represented by R= 2P [8][12]

$m=3x_1^2 + a/2y_1$

$x_3=m^2-2x_1$

$y_3=-y_1+m(x_1-x_3)$

According to formula cost of point doubling is 5M+2S+1I+4AS where M is multiplication S is squaring I is inverse and AS is addition/subtraction.

2.4 Zeckendorf Representation

Zeckendorf theorem states that a number can be represented as sum of fibonacci numbers.

Example:- 16 is not in Fibonacci series.

16 can be written as 13+3 .Here 13 and 3 are in the fibonacci series.

Example:-4

Fibonacci series 1,2,3
3<4 So 3 will be used. Set bit corresponding to 3 =1
Now 4-1 =1 is left
2>1 So bit corresponding to 2 set to 0
1=1 so bit corresponding to 1 set to 1
Representation of 4 will be = 101

3 BACKGROUND

Scalar point multiplication is the main operation in ECC. Initially it was done by double and add algorithm. It was using binary representation of number. For calculating kP only doublings and additions were required. Eg for calculating 5P= ((2(2P)) +P) 2 doublings and 1 addition are required.

Number of additions required according to double and add were n-1 where n is number of 1's in binary representation of scalar and number of doublings required were L-1 where L is length of binary representation.

Various representations were introduced to reduce the cost of scalar multiplication. Some of these are discussed in this section.

3.1 NAF Representation

We know that the binary representation of any number is unique and consists of two digits 0 or 1 [3]. However, if we use negative number too, in the representation then there exist infinite number of representations for a number having different lengths and density.Density means the number of non - zero digits. Inclusion of negative digits in the representation leads to requirement of inverse. In case of Elliptic curves inversion of a point is very simple, i.e. just the negation of the Y- co-ordinate, in case of primary field or addition of X and Y coordinate in case of binary fields. These operations are very low cost and can be neglected.

Out of all such representations, there exist exactly one representation in which there are no consecutive non zero digits [9]. This representation is known as the NAF representation and is important because it puts an upper bound on the density of any l- bit scalar k. The Non Adjacent Form (NAF) representation of a number consists of three digits 0, 1 or -1. The representation ensures that there cannot be any two or more contiguous non zero digits in the representation. As an example, suppose k = 15, in the computation of kP. Binary representation of $(15)_{10}$ is $(1111)_2$, while if we permit negative numbers then k can be represented as either of these: $(100\text{-}11)_2$or $(10\text{-}111)_2$, $(1000\text{-}1)_2$, and so on. Of these forms, $(1000\text{-}1)_2$ satisfies the condition that there are no two consecutive non zero digits. Thus, it is a NAF representation for k. It can be noticed that in this representation, four doubling and only 2 addition operations are required, while in case of binary representation, 3 doubling and 4 addition operations would be required. Thus, NAF representation can reduce the computational cost. [3]

3.2 w-NAF Representation

The NAF representation ensures that there can be no two consecutive non zero digits. Or in other ways, NAF representation ensures that in any two consecutive digits, there can be at most one non- zero digits. This idea is further extended in w-NAF representation [4][9] that ensures that there can be at most one non zero digit in any consecutive w digits in the representation. w-NAF representation is also a radix-2 representation system and was given by Cohen, Miyaji and Ono. Thus for NAF representation, width of the window can be considered to be equal to 2. With increase in w, the density of non- zero digits decreases, and thus, the number of additions also decreases.

A width w-NAF representation uses the digit set B =
$\{0, \pm 1, \pm 3, \pm 5, \pm 7, ... \pm 2^{w-1}-1\}$
This requires 2^{w-2} pre computed points.

3.3 Multibase Non-Adjacent Form (*mb*NAF)

The NAF representation ensures that there can be no two consecutive non zero digits. This idea was further extended using base set instead of using single base.

This further reduces the length of representation and density of non-zero digits. This reduced the cost of scalar point multiplication[5].

3.4 New Point Addition Formulae for ECC Applications by Nicolas Meloni1,2

In this paper a new representation is used for representing a number called Zeckendorf Representation. For calculating kP Zeckendorf representation of k is calculated, then algorithm discussed in reference [6] is used.

This algorithm is used in calculating intermediate multiplication in proposed approach.

In proposed approach multibase concept [20] is combined with this algorithm.

4 PROPOSED ALGORITHM

In proposed approach Zeckendorf representation is combined with multibase concept.First by using Algorithm 1 Sets are generated [20]. After generation of sets point multiplication is computed by Algorithm 2. Algorithm 2 will call two algorithms 2(a) and 2(b). Algorithm 2(a) is used to obtain the Zeckendorf Representation and 2(b) is used to calculate intermediate point multiplication using only point addition [6].

Some Notations used:-

Bases the multi-base set S with n base elements (bs1, bs2, bs3… bsn) (co-prime integers)

Set B this is union of terms in form of (d, b1, b2, b3….bn)

Where n is number of bases.

Algorithm 1
Generate_set (k,S)

Input : k ,base set S=(bs1,bs2,bs3… bsn)
Output: B

1. B=Null
2. While k>1
3. {
4. If(k%bs1=0 or k%bs2=0….. or k%bsn=0)
5. d=0
6. else
7. d=1 k=k-1
8. for(j=1 to n)// n is number of bases
9. {
10. bj=0
11. while(k%bsj==0)
12. {
13. bj= bj+1
14. k=k/bj
15. }
16. B=B union (d,b1,b2,…bn)}}

Example:- K=101 S=(2,3)

Iteration	K	Term
1	101	(1,2,0)
2	25	(1,3,1)

Table 1 Generation of terms

B={(1,2,0),(1,3,1)}

Algorithm 2
Computation of multiplication
Generation_multiplication (B,P)

Input:- Set B and Point P
Output : kP

1. Q=0
2. For each term in B
3. {

4. Q= Q + d*P
5. For j=1 to n
6. {
7. Arr[]=Zeckendorf(bsj bj)
8. P=fib_add(Arr,P)
9. }
10. }
11. Q=Q+P
12.

Iteration	Term	Q	P
1	(1,2,0)	P	P=4P
2	(1,3,1)	5P	P=8(4P)=32P P=3*(32P)=96P
		96P+5P=101P	

Table-2 Computation of Multiplication

Algorithm 2(a)
Algorithm to obtain Zeckendrof Representation
Zeckendorf (int n)
Input: scalar n= (bs bi)
Output: Zeckendorf representation of scalar n
Var j, s, F [1000] , bit[n] n is number of bases,sum
1. Initialize F[1]= 1
2. F[2]=2, j=2
3. Sum=2 s=1
4. While (F[j]+F[j-1]<=n and n>2)// Generating Fibonacci series upno number <=n
5. {
6. sum= F[j]+F[j-1]
7. j=j+1
8. F[j]=sum
9. }
10. for(k=j;k>=1;)
11. {
12. If(n==F[k])
13. {
14. s=s+1
15. bit[s]=1
16. for(ss=k-1;ss>=1;ss--)
17. s=s+1 bit[s]=0
18. k=k-1
19. }
20. Else if(n>F[k])
21. {
22. n=n-F[k]
23. s=s+1
24. bit[s]=1
25. k=k-1
26. }
27. Else
28. {
29. k=k-1
30. s=s+1
31. bit[s]=0}}
32. Return bit array

Example :-4
Representation of 4 will be = 101

Algorithm 2(b)
Fib_add(Zeckendorf representation of b,P)
Input : Zeckendorf representation of b and P
Output: bP
1. For(i=n-2 to 0){
2. If bit[i]=1
3. (U,V)=(U+P,V)
4. (U,V)=(U+V,U)
5. Else
6. (U,V)=(U+V,U)
7. Return U}

Above algorithm will require L-1+n-1 additions where L is the length of representation and n is number of 1.

5 COMPARISON

5.1 Comparative Analysis of proposed approach with previous approaches

In this section proposed approach is compared with previous approaches. Here cost is computed for 10 examples. The cost obtained for different examples is given in table and cost comparison is shown by graph.

5.1.2 Comparative Analysis of NAF and proposed approach.

S no	Value	Cost by using NAF D=5M+2S+1I +4AS A=2M+1S+1I+6AS	Cost by using proposed Base set (2,3) A=2M+1S+1I+6AS
1	6	3D+1A=17M+7S+4I+18AS	3A=6M+3S+3I+18AS
2	15	4D+1A=22M+9S+5I+22AS	6A=12M+6S+6I+36AS
3	30	5D+1A=27M+11S+6I+26AS	7A=14M+7S+7I+42AS
4	63	6D+1A=32M+15S+8I+30AS	9A=18M+9S+9I+54AS
5	101	7D+3A=41M+17S+10I+44AS	10A=20M+10S+10I+60AS
6	563	9D+4A=53M+22S+13I+60AS	16A=32M+16S+16I+96AS
7	1700	11D+5A=65M+27S+16I+74AS	18A=36M+18S+18I+108AS
8	2222	11D+4A=63M+26S+15I+68AS	18A=36M+18S+18I+108AS
9	3750	12D+6A=72M+30S+18I+84AS	18A=18I+18S+36M+108AS
10	11110	14D+6A=82M+34S+20I+92AS	22A=44M+22S+22I+132AS

Table- 3 Comparison between NAF and proposed approach

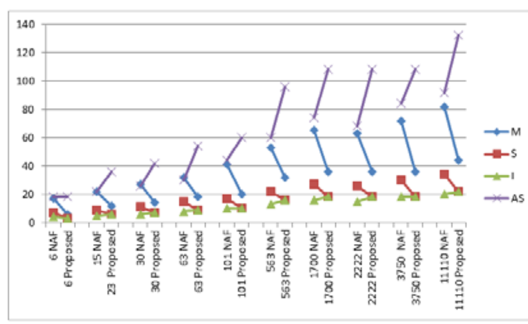

Fig- 1 Comparison between NAF and proposed approach

The above graph is showing cost comparison between NAF and proposed approach.
Horizontal axis is showing examples and vertical axis is showing the cost.
Blue line is showing multiplication. Number of multiplication is decreasing from NAF to proposed approach. For example number of multiplication at 101 NAF is 41M which is decreased to 20 M at 101 Proposed. This decrease is shown by negative slope of blue line.

Similarly Red line is showing decrease in number of squaring. For 101 NAF number of squaring is 17S which is decreased to 10S in 101 proposed.
Purple line is showing increase in number of addition and subtraction. For 101 NAF number of addition and subtraction is 44AS which are increased to 60AS in 101 proposed.
Green line is showing trend in number of inverse. For 101 NAF number of inverse is 10I which is same as in proposed. In some cases number of inverse is decreasing, in some cases numbers of inverses is increasing and in some cases number of inverse remain same.
So total decrease is 28 (21 in multiplication, 7 in squaring)
Total increase is 16 (16 in addition and subtraction)
Here for 28(total decrease) is large as compared to 16 (total increase).
In most of the cases total decrease will be found large as compared to total increase.
This decrease is based on the number of computations. In some cases number of computations will increase but these are additions and subtractions. Since addition and subtraction take small time as compared to multiplication in processors, so this approach will remain effeicient in most of cases.

5.1.3 Comparative Analysis of wNAF and proposed approach.

Here w is taken as 4. In case of w NAF some pre computed multiplications are required. For window size w pre computed entries will be {±1P, ±2P, ±3P…±.2^{w-1}P-1}.
So for w=4 Pre computed enteries will be {±1P, ±2P, ±3P, ±5P, ±7P}
It will require 1D and 3Afor computation.
1D+3A=5M+2S+1I+4AS+3(2M+1S+1I+6AS)
=11M+5S+4I+22AS

Table-4 and fig-2 is showing cost without adding pre computation cost.

S no	Value	Cost without precomputation cost by using wNAF w=4 D=5M+2S+1I +4AS A=2M+1S+1I+6AS	Cost by using proposed Base set (2,3) A=2M+1S+1I+6AS
1	15	4D+1A=22M+9S+5I+22AS	6A=12M+6S+6I+36AS
2	23	4D+1A=22M+9S+5I+22AS	8A=16M+8S+8I+48AS
3	30	5D+1A=27M+11S+6I+26AS	7A=14M+7S+7I+42AS
4	63	6D+1A=32M+15S+8I+30AS	9A=18M+9S+9I+54AS
5	101	5D+1A=27M+11S+6I+26AS	10A=20M+10S+10I+60AS
6	563	9D+2A=49M+20S+11I+48AS	16A=32M+16S+16I+96AS
7	1700	11D+2A=59M+24S+13I+56AS	18A=36M+18S+18I+108AS
8	2222	11D+2A=59M+24S+13I+56AS	18A=36M+18S+18I+108AS
9	3750	9D+3A=51M+21S+12I+54AS	18A=18I+18S+36M+108AS
10	11110	14D+3A=76M+31S+17I+74AS	22A=44M+22S+22I+132AS

Table- 4 Comparison between w NAF without pre computation cost and proposed approach

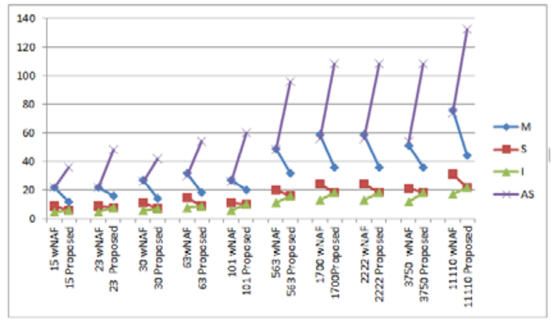

Fig- 2 Comparison between wNAF without precomputation and proposed approach

The above graph is showing cost comparison between wNAF and proposed approach without considering pre computation cost.

Horizontal axis is showing examples and vertical axis is showing the cost.

In case of wNAF if pre computation cost is not considered then its number of computations came out small in many cases as compared to proposed approach. But if pre computed cost is considered it will be high. However the computations which are increased are due to addition and subtractions in place of multiplications. Since multiplication takes more time as compared to addition and subtraction, so the proposed approach will remain better in most of the cases.

Since pre computed cost is only one time cost of a system. If enough storage is available w NAF can be preferred over other approaches

5.1.4 Comparative Analysis of mbNAF and proposed approach

In mbNAF we use a base set.Here Base set (2,3) is used.

S no	Value	Cost using mbNAF Base set (2,3) D=5M+2S+1I +4AS A=2M+1S+1I+6AS	Cost by using proposed Base set (2,3) A=2M+1S+1I+6AS
1	6	2D+1A=12M+5S+3I+14AS	3A=6M+3S+3I+18AS
2	15	3D+2A=19M+8S+5I+24AS	6A=12M+6S+6I+36AS
3	30	4D+2A=24M+10S+6I+28AS	7A=14M+7S+7I+42AS
4	63	6D+3A=36M+15S+9I+42AS	9A=18M+9S+9I+54AS
5	101	6D+2A=34M+14S+8I+36AS	10A=20M+10S+10I+60AS
6	563	8D+4A=48M+20S+12I+56AS	16A=32M+16S+16I+96AS
7	1700	10D+4A=62M+24S+14I+64AS	18A=36M+18S+18I+108AS
8	2222	10D+5A=60M+21S+15I+70AS	18A=36M+18S+18I+108AS
9	3750	10D+6A=62M+26S+16I+76AS	18A=18I+18S+36M+108AS
10	11110	13D+5A=75M+31S+18I+82AS	22A=44M+22S+22I+132AS

Table- 5 Comparison of mbNAF and proposed approach

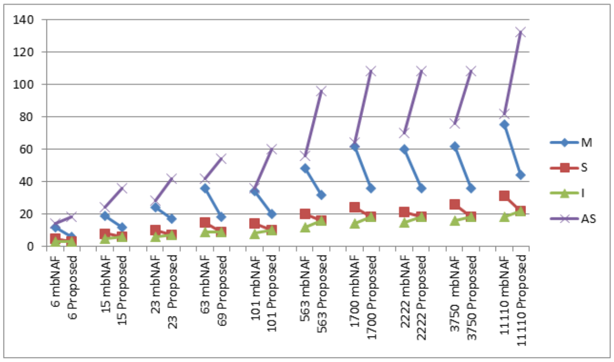

Fig-3 Comparison between mbNAF and proposed approach

The above graph is showing cost comparison between mbNAF and proposed approach.
For example for 63 total decrease is 24 (18 in multiplication, 6 in squarings)
Total increase is 12(14 in addition and subtraction)
Here for 24 (total decrease) is large as compared to 12 (total increase).
This decrease in proposed approach is based on the number of computations. In some cases number of computations in proposed approach will increase but these are additions and subtractions. Since addition and subtraction take small time as compared to multiplication in processors, so this approach will remain efficient in most of cases.

5.1.4 Comparison of proposed approach and Zeckendorf without multibase concept

In this section proposed approach is compared with Zeckendorf without multibase concept.
The algorithm used in proposed approach for calculating intermediate multiplication is used for finding scalar point multiplication in [6].
In table 6 Comparison between Zeckendorf without multibase concept and proposed approach is shown.
In fig-4 Comparison is shown in graphical form.

S no	Value	Cost using simple zeckendorf without multibase A=2M+1S+1I+6AS	Total Computations	Cost byusing proposed Base set (2,3,5) A=2M+1S+1I+6 AS	Total Computations
1	6	4A=8M+4S+4I+24A S	40	3A=6M+3S+3I+ 18AS	30
2	15	6A=12M+6S+6I+36 AS	60	5A=10M+5S+5I +30AS	50
3	30	8A=16M+8S+8I+48 AS	80	6A=12M+6S+6I +36AS	60
4	155	12A=24M+12S+12I+ 72AS	120	10A=20M+10S+ 10I+60AS	100
5	255	13A=26M+13S+13I+ 78AS	130	12A=24M+12S+ 12I+72AS	120
6	610	13A=26M+13S+13I+ 78AS	130	11A=22M+11S+ 11I+66AS	110
7	1545	18A=36M+18S+18I+ 108AS	180	16A=32M+16S+ 16I+96AS	160
8	1700	17A=34M+17S+17I+ 102AS	170	16A=32M+16S+ 16I+96AS	160
9	5355	22A=44M+22S+22I+ 132AS	220	20A=40M+20S+ 20I+120AS	200
10	11110	23A=46M+23S+23I+ 92AS	230	22A=44M+22S+ 22I+132AS	220

Table-6 Comparison between zeckendorf without multibase and proposed approach

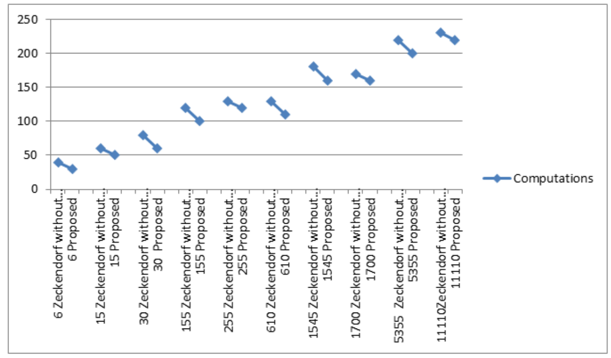

Fig-4 Comparison of proposed with Zeckendorf without multibase

The above graph is showing the decrease in number of computations. If we use simple zeckendorf representation without multibase concept number of computations will be large. However in some cases number of computations came out to be large for proposed approach. This is because of less optimal base set. This is limitation of proposed approach that it is using random base set due to which sometime cost may increase.

5.2 Comparison of single double and multibase versions of proposed approach

In this section computations are computed for single double and multibase. For single base base 2 is used,for double base base set (2,3) is used and for multibase base set (2,3,5) is used.

S no	Value	Total computations using base 2	Total Computations using base set(2,3)	Total Computations using base set (2,3,5)
1	45	100	90	80
2	90	110	100	90
3	63	100	90	90
4	139	130	120	110
5	246	130	120	110
6	2223	210	180	170
7	3750	200	180	180
8	11110	250	220	200

Table-7 Comparison between single double and multibase

From the table we can analyze that number of computations are decreasing from single to double base and double to triple base. But in some cases like 3750 number of computations are same for double and triple base. This is due to limitation of the proposed approach that base set is not optimal.

6 Conclusion and Future Work

The proposed approach is using Zeckendorf Representation of number and multibase concept.
It removes the doublings completely. It has no overhead of precomputed enteries.
This decreases the number of multiplications and squarings in most of cases. The limitation of proposed approach is that base set selected is predefined due to which sometimes

cost get increased as compared to previous approach. It can be extended to choose the base set according to the scalar whose point multiplication needs to be calculated such that base set is optimized and number of precomputations can be further reduced.

REFERENCES

[1] N. Koblitz. Elliptic Curve Cryptosystems. Mathematics of Computation. Vol. 48, pp. 203–209, 1987.

[2] V. Miller. Use of Elliptic Curves in Cryptography, Advances in Cryptology. Crypto'85, LNCS Vol. 218, pp. 417-426, Springer, 1986.

[3] G.W. Reitweisner , D. Hankerson, A. Menezes, and S.A. Vanstone, Guide to Elliptic Curve Cryptography Springer-Verlag,2004 pg 98

[4] wNAF an Efficient Left-to-Right Signed Digit Recoding Algorithm Brian King Indiana University Purdue University Indianapolis Springer-Verlag Berlin Heidelberg 2008

[5] New Multibase Non-Adjacent Form Scalar Multiplication and its Application to Elliptic Curve Cryptosystems *eprint.iacr.org 2008* Patrick Longa, and Ali Miri

[6] New Point Addition Formulae for ECC Applications Nicolas Meloni1,2 2012 springer

[7] William Stallings. *Cryptography and Network security: Principles and Practices*. Prentice Hall Inc., second edition, 1999.

[8] Network Security and Cryptography Bernard Menezes IIT Bombay (Book)

[9] Different Number RepresentationTechniques for ECC Optimization Guided By Prof. Bernard Menezes

[10] Setting Speed Records with the Multibase Non-Adjacent Form Method for Efficient Elliptic Curve Scalar Multiplication Patrick Longa and Catherine Gebotys Department of Electrical and Computer Engineering University of Waterloo, Canada 3. New Multibase Non-Adjacent Formeprint.iacr.org/2008/118

[11] A Novel Algorithm for Scalar Multiplication in ECDSA Hui Li, Ruixia Zhang, Junkai Yi, Hongqiang Lv.

[12] A Tutorial on Elliptic Curve Cryptography (ECC) Fuwen Liu.

[13] Vassil S. Dimitrov and Kimmo U. J arvinen and JithraAdikari, Area-E_cient Multipliers Based on Multiple-Radix Representations, IEEE Trans. Computers, vol 60(2), 2011, pg 189-201

[14] Christophe Doche and Laurent Imbert. Extended double-base number system with applications to elliptic curve cryptography. In proceedings of the 7th international conference on Cryptology in India, IN-DOCRYPT'06, pages 335–348, Berlin, Heidelberg, 2006. Springer-Verlag

[15] A New Generic Algorithm for Scalar Multiplication ZHANG Bao-hua YIN Xin-chun 2009

[16] [Thesis] Different Number RepresentationTechniques for ECC Optimization Guided By Prof. Bernard Menezes

[17] Elliptic Curve Cryptography, http://www.isg.rhul.ac.uk/~sdg/ecc.html

[18] Elliptic Curve Basics, http://en.wikipedia.org/wiki/elliptic_curve

[19] Object-Oriented Analysis and Design of Elliptic Curve Cryptosystem Shen Guicheng Zheng Xuefeng Liu Bingwu

[20] On-the-Fly Multi-Base Recoding for ECC Scalar Multiplication without Pre-Computations by Thomas Chabrier , and Arnaud Tisserand