# A new Scalar Point Multiplication Scheme in ECC based on Zeckendorf Representation and Multibase Concept

**KIRTI CHAWLA**

**Kirtichawla17@gmail.com**

**OMPAL YADAV**

**CDAC NOIDA**

**Ompal.cdac@gmail.com**

*Abstract-With the fast development of cryptography research and computer technology, the cryptosystems of RSA and Diffe-Hellman are getting more and more unsafe, and Elliptic Curve Cryptosystem is becoming the trend of public cryptography in the future. Scalar Point Multiplication Scalar multiplication is the time consuming operation in elliptic curve based cryptosystem. In this paper, Nicolas Meloni1,2 2012 springer algorithm for addition of points on elliptic curve is used along with multibase concept to improve the speed of the scalar multiplication. Comparative analysis of proposed approach and some previous approaches is also discussed in last.*

***Keywords-Elliptic curve; Scalar multiplication ;NAF Representation ;multi base NAF Representation; Zeckendorf Representation ;***

## I. INTRODUCTION

Elliptic Curve is always regarded as a joint point of algebra geometry, number theory and a purified discipline. Nowadays, Elliptic Curve Cryptosystem (ECC) has made great progress not only in theory but also in practice. The key length of classical cryptosystem, such as RSA and Diffe-Hellman, is 516 bits, but with the rapid development of cryptography theory and computer technology, this key length is getting more and more unsafe. To reach the safe level of symmetric cryptosystems with key length of 128 bits, NIST recommends that the key length must be 3072 bits. It is obvious that such increasing of key length is a heavy burden for RSA because the speed of RSA has been very slow, and this phenomenon will be kept for a long time. Compared with classical public cryptographies, ECC has made great progress in algorithm efficiency. The crisis of RSA results from its existence of sub-exponential-time attack, and as for

ECC, it generally has no such attacks, therefore its key length can be cut greatly. The safe level of ECC with the key length of 256 bits is similar with the safe level of symmetrical cryptosystems with the key length of 128 bits. In 1985, Neal Koblitz and Victor Miller independently used elliptic curves in cryptography in their papers [1] and [2],proposed Elliptic Curve Cryptography (ECC). These years, researchers always pay more attention to improve the efficiency of ECC, the main operation is scalar multiplication which 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, so that a fast and secure scalar multiplication algorithm is required. In section I some introduction is given about ECC. In section II preliminaries are discussed. In section III some related work is discussed. In section IV proposed approach is discussed and in section V Comparison of previous approaches and proposed approach is discussed.

## II. PRELIMINERIES

### *A. Elliptic Curve*

Elliptic Curve Cryptography (ECC) is based on a finite group of points on an elliptic Curve. An elliptic curve is a plane curve defined by an equation of the form. This equation is for elliptic curve over infinite fields[8].

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

### *B. Elliptic Curve Point Addition*

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].

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.

### C. *Elliptic Curve Point Doubling*

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]

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

$x_3=m^2-2x_1$

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

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.

### D. *Zeckendorf Representation*

Zeckendorf theorem states that a number can be represented as sum of some 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

To obtain the Zeckendorf representation of any number n we first find the fibonacci series starting from 1 upto maximum number less than equal to n Then we can use a greedy algorithm to generate the representation.

**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

## III. RELATED WORK

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 was n-1 where n is number of 1's in binary representation of scalar and number of doublings required was 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

### A. 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 negative number too, in the representation then there exist infinite number of representations for a number having different lengths and density. By density we mean 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. 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. In fact it can be proved that the NAF representation contains minimum number of non zero digits. Thus NAF representation requires minimum (or in some case it can be just one more than the minimum) computations. The reduction in this example does not seem to be much significant, but in actual implementations, the scalar k is very large and there the reductions can be more significant.[3]

**Advantage:-**

- It reduces the density of non zero numbers So reduced the number of additions.

**Disadvantage:-**

- It sometimes increases the length of representation So it increase the number of doublings.

**Example** is 15 its binary representation will be (1111) .If we represent it in NAF form it will be $(1000\text{-}1)_{NAF}$
It will reduce the number of additions by 2 but increase the number of doubling by 1

## B. W-NAF Form 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 digit. This idea is further extended in w-NAF representation [4] 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 representations uses the digit set $B = \{ 0,\pm1,\pm 3,\pm 5,\pm 7, \ldots.. \pm 2^{w-1}-1\}$

This requires $2^{w-2}$ pre computed points.

**Advantage:**

- It reduces the density of non zero numbers. So reduced the number of additions
- It reduces the length of representation .So Number of doublings also get reduced.

**Disadvantage:-**

- It has overhead of pre computed entries of pre computed entries.

**Example** is 28 its binary representation will be (11100) .If we represent it in 4NAF form it will be $(700)_{7NAF}$

It will reduce the number of additions by 2 and decrease the number of doubling by 2. But it will require 4 pre computed entries.

## C. Multibase Non-Adjacent Form (*mb*NAF)

The non-adjacency property (as can be found in NAF) allows the insertion of consecutive "zero" terms in the expansion of an integer, which effectively minimizes the expected nonzero density. A key observation is that, by combining this property with the use of several bases[5], one is able of flexibly inserting consecutive "zero" terms using an extended set of bases (i.e., 3, 5, 7 and so on, besides 2), which can be expected to reduce further the nonzero density in the representation of integers.

**Advantage:-**

- It increases the density of zeros , So reduced the number of additions.
- It reduces the length of representation So Number of doublings also get reduced

**Disadvantage:-**

- It has overhead of intermediate multiplications
- It also require some precomputation depending upon the base set used.

Example is 28 its binary representation will be (11100) .If we represent it in (2,3,5,7)NAF form it will be $1^{(2)}0^{(7)}0^{(2)}0^{(2)}$
It will reduce the number of additions by 2 and decrease the number of doubling by 2. but increase overhead of intermediate multiplication by 7 (for this example)

### D. 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 is added with this algorithm.

## IV. NEW SCALAR MULTIPLICATION ALGORITHM

In proposed approach Zeckendorf representation with multibase concept is used.First by using Algorithm 1 Sets are generated. 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.

**Some Notations used:-**

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

Set B which 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) |

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

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

**Algorithm 2(a)**

**Algorithm to obtain Zeckendrof Representation**

**zeckendorf (int n)**

Input : scalar n=$(bs^{bi})$

Output: Zeckendorf  representationof 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

Example : 4P

| **Iteration** | **Bit** | **U** | **V** | **(U,V)** |
|---|---|---|---|---|
| 1 | 0 | 2P | P | (2P,P) |
| 3 | 1 | 3P<br>4P | P<br>3P | (3P,P)<br>(4P,3P) |

## V. COMPARISON

### Comparative Analysis of proposed approach with previous approaches

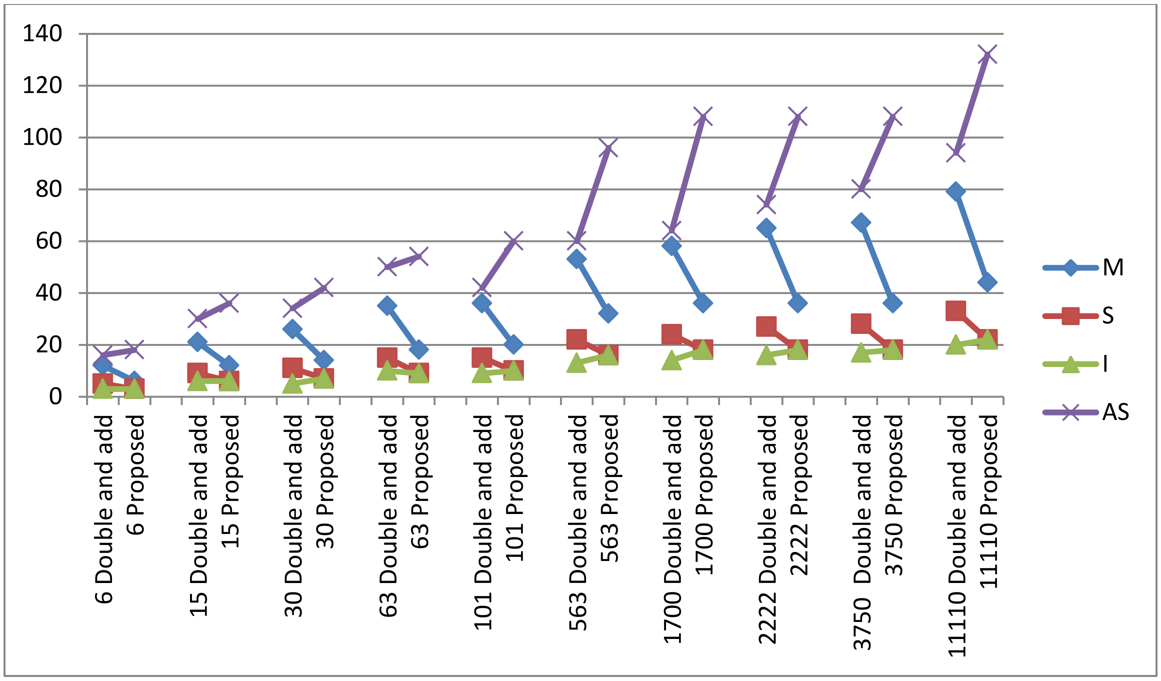
In this section proposed approach is compared with previous approaches. It requires some formulae to be described first as below. Here cost is computed for 10 examples. The cost obtained for different examples is given in table and cost comparison is shown in graph

**Comparison between Simple Double and Add and Proposed Approach:-**

Here cost is computed for 10 examples in case of double and add and proposed approach. The cost obtained for different examples is given in table.

| S no | Value | Cost by using Double and Add<br>D=5M+2S+1I +4AS<br>A=2M+1S+1I+6AS | Cost by using proposed<br>Base set (2,3)<br>A=2M+1S+1I+6AS |
|---|---|---|---|
| **1** | 6 | 2D+1A=12M+5S+3I+16AS | 3A=6M+3S+3I+18AS |
| | | | |
| **2** | 15 | 3D+3A=21M+9S+6I+30AS | 6A=12M+6S+6I+36AS |
| | | | |
| **3** | 30 | 4D+3A=26M+11S+5I+34AS | 7A=14M+7S+7I+42AS |
| | | | |
| **4** | 63 | 5D+5A=35M+15S+10I+50AS | 9A=18M+9S+9I+54AS |
| | | | |
| **5** | 101 | 6D+3A=36M+15S+9I+42AS | 10A=20M+10S+10I+60AS |
| | | | |
| **6** | 563 | 9D+4A=53M+22S+13I+60AS | 16A=32M+16S+16I+96AS |
| | | | |
| **7** | 1700 | 10D+4A=58M+24S+14I+64AS | 18A=36M+18S+18I+108AS |
| | | | |

| | | | |
|---|---|---|---|
| **8** | 2222 | 11D+5A=65M+27S+16I+74AS | 18A=36M+18S+18I+108AS |
| | | | |
| **9** | 3750 | 11D+6A=17I+28S+67M+80AS | 18A=18I+18S+36M+108AS |
| | | | |
| **10** | 11110 | 13D+7A=79M+33S+20I+94AS | 22A=44M+22S+22I+132AS |




**Cost Comparison of Double and Add and Proposed Approach**

The above graph is showing cost comparison between double and add and proposed approach.

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

Blue line is showing multiplication. Number of multiplication is decreasing from double and add to proposed approach. For example number of multiplication at 6 double and add is 12M which is decreased to 6M at 6 Proposed. This decrease is shown by negative slope of blue line.

Similarly Red line is showing decrease in number of squarings. For 6 double and add number of squaring is 5S which is decreased to 3S in 23 proposed.

Purple line is showing increase in number of addition and subtraction. For 6 double and add number of addition and subtraction is 16AS which are increased to 18AS in 6 proposed.

Green line is showing trend in number of inverse. For 6 double and add number of inverse is 3I which is same in proposed. In some cases number of inverse is decreasing, in some cases number of inverse is increasing and in some cases number of inverse remains same.

So total decrease is 8 (6 in multiplication,2 in squaring )

Total increase is 2 (2 in addition and subtraction)

Here for 8( total decrease) is more than to 2 (total increase ).

In most of the cases total decrease will be found large as compared to total increase.

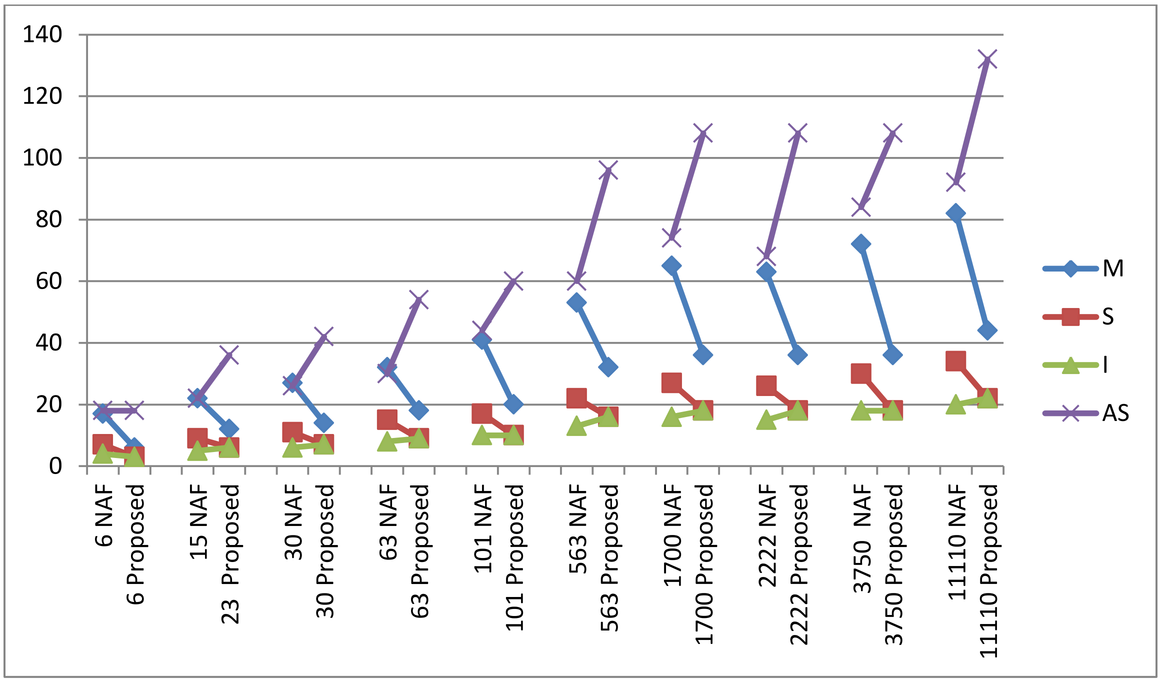
This decrease in proposed approach is based on the number of computations. In some cases number of computations in proposed approach will increase but these are additions and subtractions. Since addition and subtraction take small time as compared to multiplication in processors, so this approach will remain efficient in most of cases.

**Comparison between NAF approach and Proposed Approach:-**

Here cost is computed for 10 examples in case of double and add and proposed approach. The cost obtained for different examples is given in table.

| S no | Value | Cost by using NAF<br>D=5M+2S+1I +4AS<br>A=2M+1S+1I+6AS | Cost by using proposed<br>Base set (2,3)<br>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 |
| | | | |
| **3** | 63 | 6D+1A=32M+15S+8I+30AS | 9A=18M+9S+9I+54AS |
| | | | |
| **4** | 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 |




Cost Comparison of NAF and Proposed Approach

The above graph is showing cost comparison between NAF and proposed approach.

Horizontal axis 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 in proposed . In some cases number of inverse is decreasing, in some cases number of inverse 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.

**Comparison between wNAF approach 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 $\{\pm 1P, \pm 2P, \pm 3P \ldots \pm .2^{w-1}P-1\}$.

So for w=4 Pre computed enteries will be $\{\pm 1P, \pm 2P, \pm 3P, \pm 5P, \pm 7P\}$

It will require 1D and 3Afor computation.

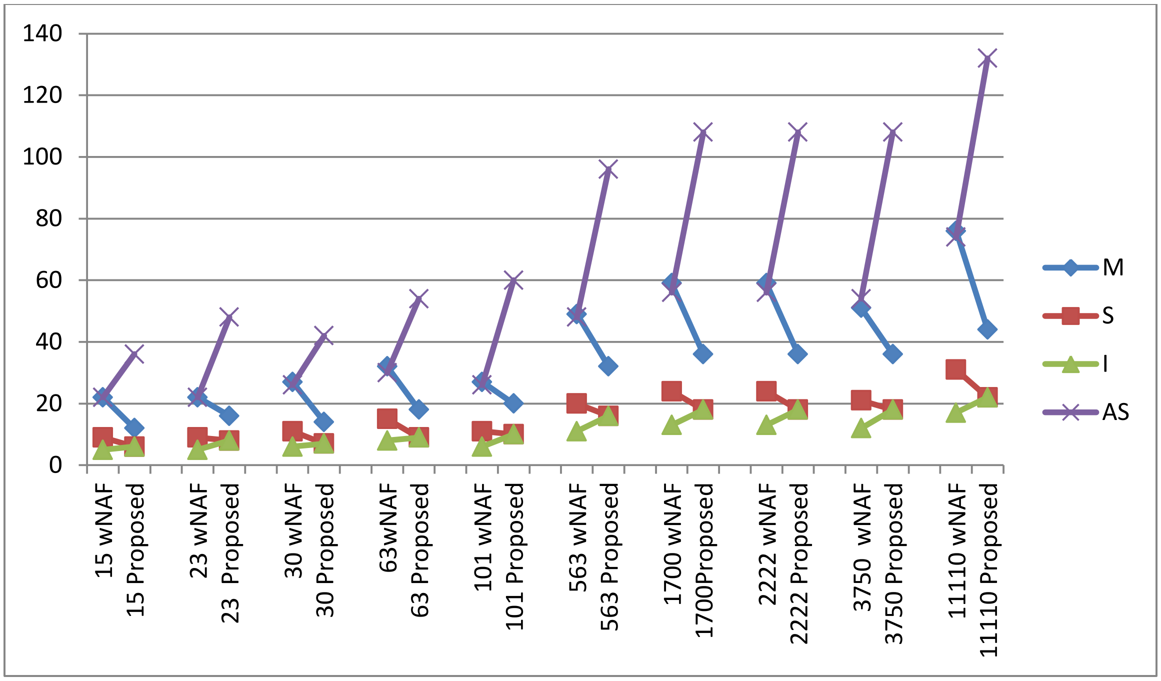
1D+3A=5M+2S+1I+4AS+3(2M+1S+1I+6AS)=11M+5S+4I+22AS

First table and graph is showing cost without adding pre computation cost.

Second Table and graph showing cost with precomputation cost added.

*Table 1*

| S no | Value | Cost without precomputation cost by using wNAF w=4<br>D=5M+2S+1I +4AS<br>A=2M+1S+1I+6AS | Cost by using proposed Base set (2,3)<br>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 |




Cost Comparison of wNAF without Pre computation cost and Proposed Approach

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

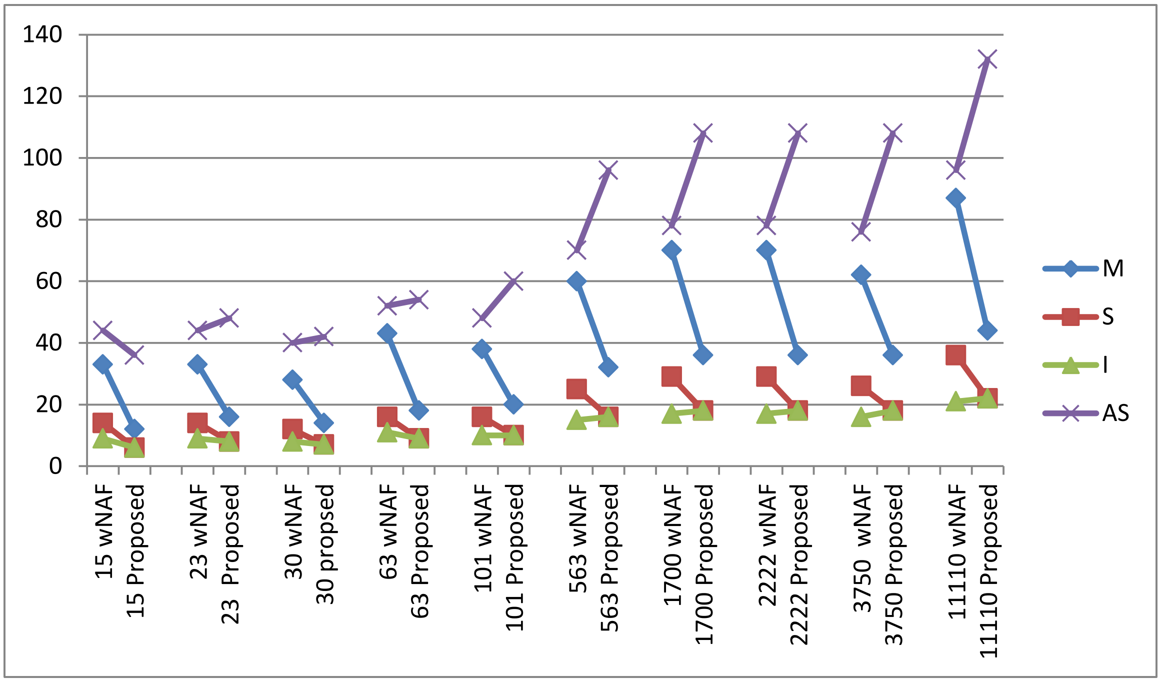
Horizontal axis showing examples and vertical axis is showing the cost.

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

*Table 2*

| S no | Value | Cost with precomputation cost by using wNAF w=4<br>D=5M+2S+1I +4AS<br>A=2M+1S+1I+6AS | Cost by using proposed Base set (2,3)<br>A=2M+1S+1I+6AS |
|---|---|---|---|
| **1** | 15 | 5D+4A=33M+14S+9I+44AS | 6A=12M+6S+6I+36AS |
| | | | |
| **2** | 23 | 5D+4A=33M+14S+9I+44AS | 8A=16M+8S+8I+48AS |
| | | | |
| **3** | 30 | 4D+4A=28M+12S+8I+40AS | 7A=14M+7S+7I+42AS |
| | | | |
| **4** | 63 | 7D+4A=43M+16S+11I+52AS | 9A=18M+9S+9I+54AS |
| | | | |
| **5** | 101 | 6D+4A=38M+16S+10I+48AS | 10A=20M+10S+10I+60AS |
| | | | |
| **6** | 563 | 10D+5A=60M+25S+15I+70AS | 16A=32M+16S+16I+96AS |
| | | | |
| **7** | 1700 | 12D+5A=70M+29S+17I+78AS | 18A=36M+18S+18I+108AS |
| | | | |
| **8** | 2222 | 12D+5A=70M+29S+17I+78AS | 18A=36M+18S+18I+108AS |
| | | | |
| **9** | 3750 | 10D+6A=62M+26S+16I+76AS | 18A=18I+18S+36M+108AS |
| | | | |
| **10** | 11110 | 15D+6A=87M+36S+21I+96AS | 22A=44M+22S+22I+132AS |
| | | | |




**Cost Comparison of wNAF with Pre computation cost and Proposed Approach**

The above graph is showing cost comparison between wNAF and proposed approach with pre computation cost added in the cost.

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

Blue line is showing multiplication. Number of multiplication is decreasing from wNAF to proposed approach. For example number of multiplication at 11110 wNAF is 87M which is decreased to 44 M at 11110 Proposed. This decrease is shown by negative slope of blue line.

Similarly Red line is showing decrease in number of squaring. For 11110 wNAF number of squaring is 36S which is decreased to 22S in 11110 proposed.

Purple line is showing trend in number of addition and subtraction. For 11110 wNAF number of addition and subtraction is 96AS which are increased to 132AS in 11110 proposed.

Green line is showing trend in number of inverse. For 11110 NAF number of inverse is 21I which is increased to 22I. In some cases number of inverse is decreasing, in some cases number of inverse is increasing  and in some cases number of inverse remain same.

So total decrease is 57 (43 in multiplication ,14 in squarings )

Total increase is 36(36 in addition and subtraction)

Here for 57( total decrease) is large as compared to 36 (total increase ).

In case of wNAF if pre computation cost is considered then its cost came out large in most of the cases as compared to proposed approach.

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.

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.

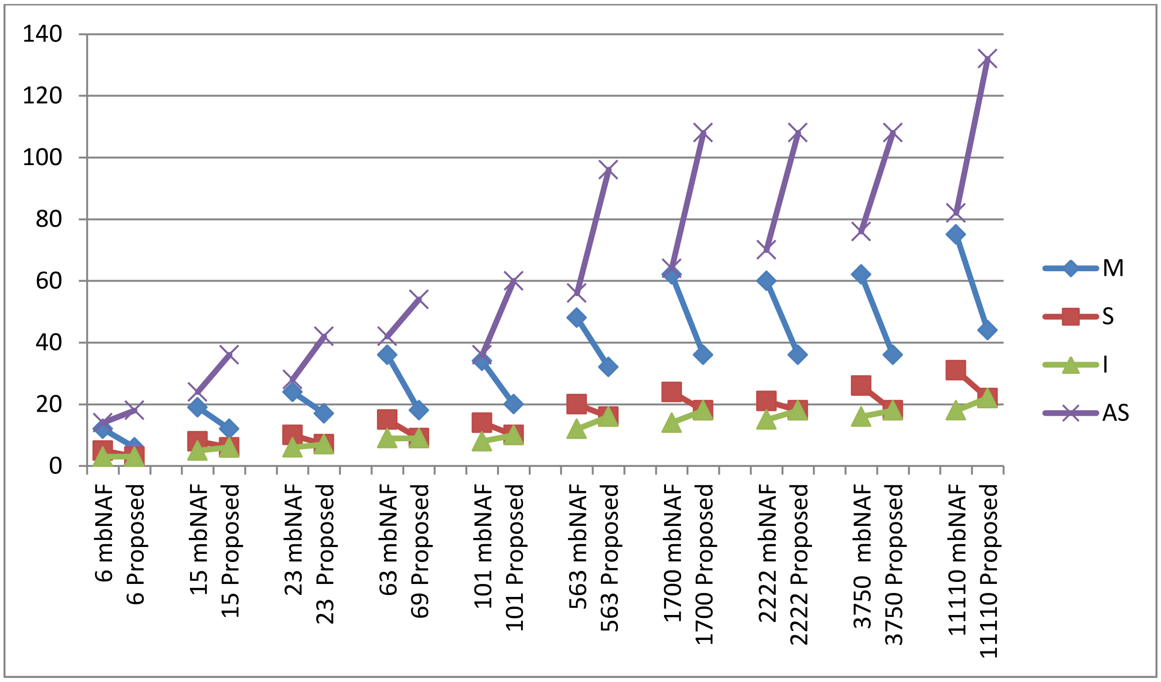
**Comparison between mbNAF approach and Proposed Approach:-**

In mbNAF we use a base set

Here Base set (2,3) is used

| S no | Value | Cost using mbNAF Base set (2,3)<br>D=5M+2S+1I +4AS<br>A=2M+1S+1I+6AS | Cost by using proposed Base set (2,3)<br>A=2M+1S+1I+6AS |
|---|---|---|---|
| **1** | 6 | 2D+1A=12M+5S+3I+14AS | 3A=6M+3S+3I+18AS |
| | | | |
| **1** | 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 |




**Cost Comparison of mbNAF and Proposed Approach**

The above graph is showing cost comparison between mbNAF and proposed approach.

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

Blue line is showing multiplication. Number of multiplication is decreasing from mbNAF to proposed approach. For example number of multiplication at 63 mbNAF is 36M which is decreased to 18 M at 63Proposed. This decrease is shown by negative slope of blue line.

Similarly Red line is showing decrease in number of squarings. For 63 mbNAF number of squaring is 15S which is decreased to 9S in 63proposed.

Purple line is showing trend in number of addition and subtraction. For 63mbNAF number of addition and subtraction is 42AS which are increased to 54AS in 63 proposed.

Green line is showing trend in number of inverse. For 63mbNAF number of inverse is 9I which is same in proposed. In some cases number of inverse is decreasing, in some cases number of inverse is increasing and in some cases number of inverse remains same.

So 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.

## COMPARISON OF PROPOSED APPROACH AND ZECKENDORF WITHOUT MULTIBASE CONCEPT

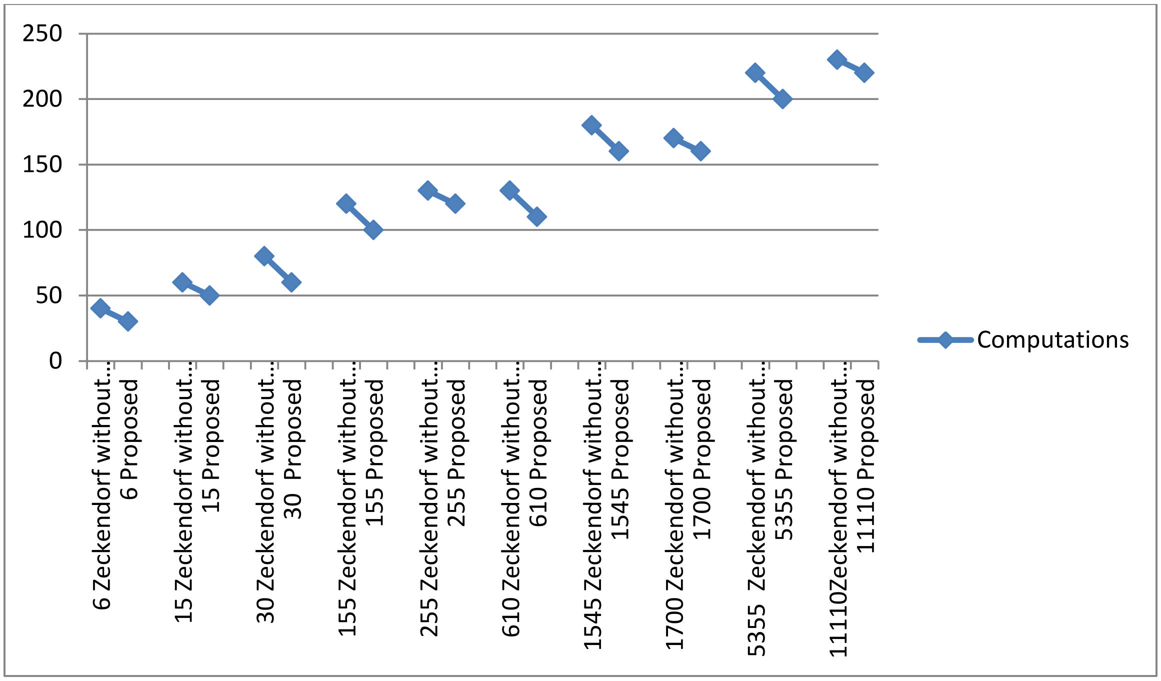
In this section proposed approach is compared with Zeckendorf without multibase concept.

The algorithm used in proposed approach for calculating intermediate multiplication can be used for finding scalar point multiplication.

| S no | Value | Cost using simple zeckendorf without multibase A=2M+1S+1I+6AS | Cost by using proposed Base set (2,3,5) A=2M+1S+1I+6AS |
|---|---|---|---|
| **1** | 6 | 4A=8M+4S+4I+24AS | 3A=6M+3S+3I+18AS |
| | | | |
| **1** | 15 | 6A=12M+6S+6I+36AS | 5A=10M+5S+5I+30AS |
| | | | |
| **3** | 30 | 8A=16M+8S+8I+48AS | 6A=12M+6S+6I+36AS |
| | | | |
| **4** | 155 | 12A=24M+12S+12I+72AS | 10A=20M+10S+10I+60AS |
| | | | |
| **5** | 255 | 13A=26M+13S+13I+78AS | 12A=24M+12S+12I+72AS |
| | | | |
| **6** | 610 | 13A=26M+13S+13I+78AS | 11A=22M+11S+11I+66AS |
| | | | |
| **7** | 1545 | 18A=36M+18S+18I+108AS | 16A=32M+16S+16I+96AS |

| | | | |
|---|---|---|---|
| **8** | 1700 | 17A=34M+17S+17I+102AS | 16A=32M+16S+16I+96AS |
| **9** | 5355 | 22A=44M+22S+22I+132AS | 20A=40M+20S+20I+120AS |
| **10** | 11110 | 23A=46M+23S+23I+92AS | 22A=44M+22S+22I+132AS |

| S no | Value | Total computations using simple zeckendorf without multibase | Total computations using proposed Base set (2,3,5) |
|---|---|---|---|
| **1** | 6 | 4A=40 | 3A=30 |
| **1** | 15 | 6A=60 | 5A=50 |
| **3** | 30 | 8A=80 | 6A=60 |
| **4** | 155 | 12A=120 | 10A=100 |
| **5** | 255 | 13A=130 | 12A=120 |
| **6** | 610 | 13A=130 | 11A=110 |
| **7** | 1545 | 18A=180 | 16A=160 |
| **8** | 1700 | 17A=170 | 16A=160 |
| **9** | 5355 | 22A=220 | 20A=200 |
| **10** | 11110 | 23A=230 | 20A=220 |




Comparison of proposed algorithm 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.

**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.

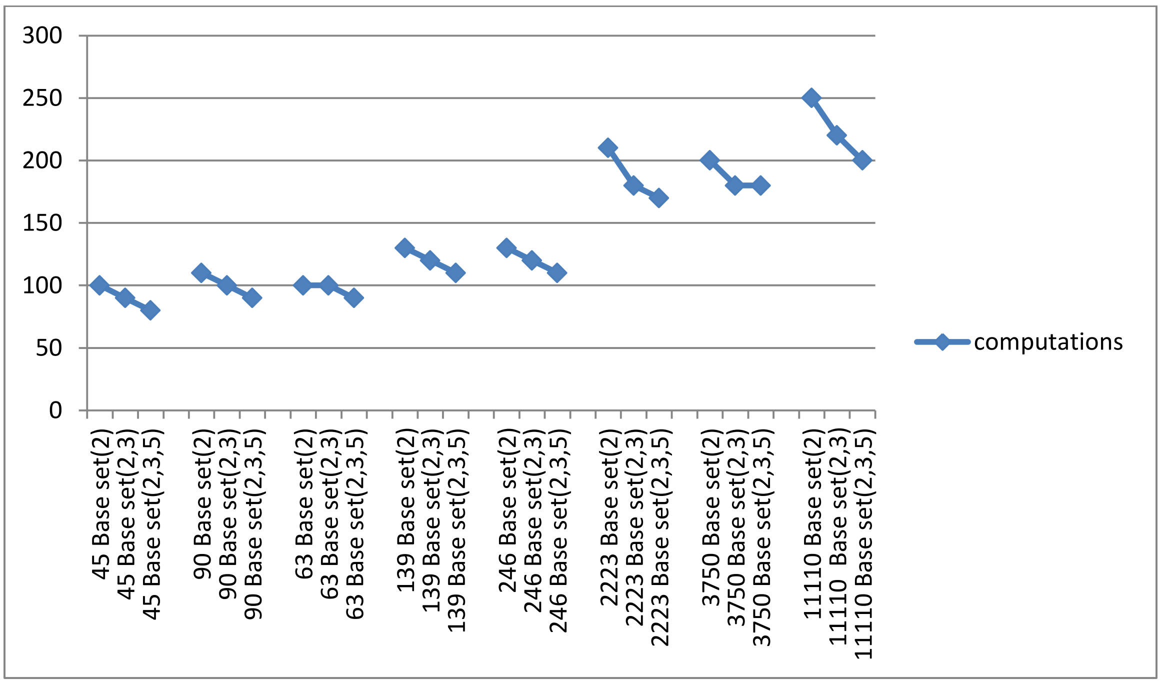
| S no | Value | Cost by using proposed Base set (2) A=2M+1S+1I+6AS | Total computations |
|---|---|---|---|
| **1** | 45 | 10A=20M+10S+10I+60AS | 100 |
| | | | |
| **1** | 90 | 11A=22M+11S+11I+66AS | 110 |
| | | | |

| | | | |
|---|---|---|---|
| **3** | 63 | 10A=20M+10S+10I+60AS | 100 |
| | | | |
| **4** | 139 | 13A=26M+13S+13I+78AS | 130 |
| | | | |
| **5** | 246 | 13A=26M+13S+13I+78AS | 130 |
| | | | |
| **6** | 2223 | 21A=42M+21S+21I+126AS | 210 |
| | | | |
| **7** | 3750 | 20A=40M+20S+20I+120AS | 200 |
| | | | |
| **8** | 11110 | 25A=50M+25S+25I+150AS | 250 |
| | | | |

| S no | Value | Cost by using proposed Base set (2,3) A=2M+1S+1I+6AS | Total computations |
|---|---|---|---|
| **1** | 45 | 9A=18M+9S+9I+54AS | 90 |
| | | | |
| **1** | 90 | 10A=20M+10S+10I+60AS | 100 |
| | | | |
| **3** | 63 | 9A=18M+9S+9I+54AS | 90 |
| | | | |
| **4** | 139 | 12A=24M+12S+12I+72AS | 120 |
| | | | |
| **5** | 246 | 12A=24M+12S+12I+72AS | 120 |
| | | | |
| **6** | 2223 | 18A=36M+18S+18I+108AS | 180 |
| | | | |
| **7** | 3750 | 18A=36M+18S+18I+108AS | 180 |
| | | | |
| **8** | 11110 | 22A=44M+22S+22I+132AS | 220 |

| S no | Value | Cost by using proposed Base set (2,3,5) A=2M+1S+1I+6AS | Total computations |
|---|---|---|---|
| **1** | 45 | 8A=16M+8S+8I+48AS | 80 |
| | | | |
| **1** | 90 | 9A=18M+9S+9I+54AS | 90 |
| | | | |
| **3** | 63 | 9A=18M+9S+9I+54AS | 90 |
| | | | |
| **4** | 139 | 11A=22M+11S+11I+66AS | 110 |

| | | | |
|---|---|---|---|
| | | | |
| **5** | 246 | 11A=22M+11S+11I+66AS | 110 |
| | | | |
| **6** | 2223 | 17A=34M+17S+17I+102AS | 170 |
| | | | |
| **7** | 3750 | 18A=36M+18S+18I+108AS | 180 |
| | | | |
| **8** | 11110 | 20A=40M+20S+20I+120AS | 200 |
| | | | |




**Comparison of single double and multibase versions of proposed algorithm**

From the graph 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.

## VI. CONCLUSION AND FUTUTE WORK

The proposed approach is using Zeckendorf Representation of number multibase concept. It removes the doublings completely. It has no overhead of precomputed enteries. This decreases the number of computations if base set is optimized. This is limitation of proposed approach 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 need to be calculated such that base set is optimized and number of precomputations can be further reduced.

## VII. 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*

[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] Optimization in Elliptic Curve Cryptograpy Thesis report by Deepak Sahu ,Deepak Suyel, Prashant Singh ,Rohit Jaiswal IIT Bombay.

[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 num-ber 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] New Multibase Non-Adjacent Form Scalar Multiplication and its Application to Elliptic Curve Cryptosystems (extended version) Patrick Longa, and Ali Miri