Estimation of Various Scalar Multiplication Algorithms in ECC

Vimal Gaur, Bineet Singh, Deepak, Megha Agrawal, Nimisha Mishra

Abstract:In this modern era of security, public key cryptography is quite popular and holds a great significance. Various public key cryptosystems are available in today's environment such as RSA and ECC. Elliptic Curve cryptography is beneficial in a lot of aspects which includes shorter key as compared to other cryptosystems, high security, fast processing speed, low storage, low bandwidth, small software print, low hardware implementation costs, high performance. The main and the costliest step in ECC is the Scalar Multiplication. In scalar multiplication, integer multiple of an element in additive group of elliptic curves is calculated. In this paper, we compare various available algorithms for the scalar multiplication used in ECC.

Index Terms:Comparison of Scalar Multiplication, ECC, Elliptic Curve Cryptography, Scalar Multiplication.

I. INTRODUCTION

1.1 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography works like other public-key encryption techniques which uses elliptic curve theory. It is a faster and efficient method for the creation of cryptographic keys. It was developed by V. S. Miller[1] and N. Koblitz[2]. They also cited the advantages of Elliptic curve cryptography over traditional techniques such as RSA[3] in their paper.

The purpose of keys is to encrypt and decrypt data. The two type of keys are Public key and Private key. The generation of these keys is a very important part for cryptography.

ECC does not uses the traditional methods for the generation of keys such as the multiplication of huge prime numbers as in RSA, rather it uses characteristics of elliptic curve equation.

Elliptic curves have properties that are very important for cryptography, i.e. they are comparatively easy to evaluate and very hard to reverse, hence it acts as a trapdoor function

1.1.1 Advantages of ECC

Elliptic curve cryptography provides same level of safety with 164 bit key that others provide with 1024 bit key, hence making it 15 times stronger than the other cryptosystems. ECC is considered much more suitable over other public-key cryptosystems as it consumes low computing power and performs better than the other cryptography algorithms.

1.2 Elliptic Curves

Elliptic curve is a type of algebraic curve of form:
$y^2 = (x^3 + ax^2 + b)\text{mod } p$
where, a,b belongs to $Z_p$such that $4a^3 + 27b^2 \neq 0$

Elliptic curves are represented by Weierstrass equation. It operates within a finite field.

These curves are non-singular, that is the curve do not contain cusps or self-intersections. It belongs to abelian group also known as commutative group. The curve has a point O, which acts as its identity and is often taken as the curve's 'point at infinity' in the projection plane. This point cannot be visualized in two-dimensional plane. More details are available in [4] and [5].

Elliptic curves include a vast area of current research, it includes number theory, Elliptic Curve Cryptography and factorization of integers.

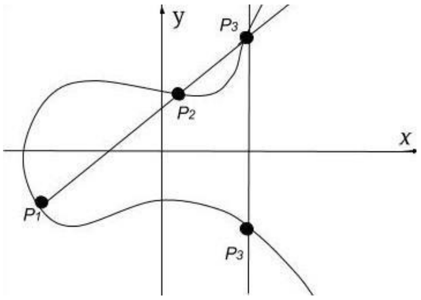

Fig 1: Example of Elliptic Curve

1.2.1 Properties of elliptic curves

1. Point at infinity - This is the point on the curve which is the identity of the curve and is represented by O.

2. Point addition - Let there is two points P and Q on the curve, point addition is defined as negative of the point obtained by intersection between curve and the line formed by joining the two points. (P+Q=R)

3. Point doubling - When P and Q are same, i.e. they are the same coordinates then it is called Doubling. It is same as addition except there is no well-defined line between P and Q, hence, we create a tangent at point to curve. (P+P= 2P=R)

1.3 Scalar Multiplication

Scalar multiplication is the most essential and time consuming step of ECC. It is also known as Elliptic Curve Point Multiplication. Hence efficiency of the scalar multiplication algorithm used is essential.

Revised Manuscript Received on April 12, 2019.

Vimal Gaur,Guide Reader, CSE Department, Maharaja Surajmal Institute of Technology, Delhi, India. (Email: vimalgaur@msit.in)

Bineet Singh,Student, CSE Department, Maharaja Surajmal Institute of Technology, Delhi, India (Email: bineetsingh30@gmail.com)

Deepak,Student, CSE Department, Maharaja Surajmal Institute of Technology, Delhi, India (Email: deepak00231996@gmail.com)

MeghaAgrawal,Student, CSE Department, Maharaja Surajmal Institute of Technology, Delhi, India (Email: megha9673@gmail.com)

Nimisha Mishra,Student, CSE Department, Maharaja Surajmal Institute of Technology, Delhi, India (Email: nimisha.mi296@gmail.com)

DOI: 10.35940/ijitee.F1036.0486S419

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Scalar multiplication used in ECC can also be defined as inverse of the ECDLP(Elliptic Curve Discrete Logarithm Problem), i.e. given n and nP and to compute n.

Scalar multiplication can be defined as the process of consecutive addition of a point along the elliptic curve. Let there be a point P that lies on the curve, and n be some scalar, then nP=P+P…….+P till n times. The security is dependent on the intractability of the determination of n from Q=nP. Given nP and P, but it is almost impossible to track down n, it is computationally intractable. It behaves like a one-way function.

The chosen algorithm for scalar multiplication affects the complexity and efficiency of the process of elliptic curve cryptography majorly. In this paper, we compare different scalar multiplication algorithms.

Right to left binary method, Left to Right Binary method, Addition Subtraction using NAF, Windowed method, Montgomery ladder, and wNAF method are some of the algorithms used for Scalar Multiplication and further compared in this paper.

1.3.1 Left to Right Binary Method

Let there is a point P on the curve and we have to evaluate nP for some integer n. In left to right algorithm for scalar multiplication the integer n is used in binary form and it is processed from left to right. The process involves point doubling and point addition. For the binary representation of n while iterating for 0 and 1 from left to right, we perform point doubling. Point addition is done only when 1 is present in binary representation. Final value of P then represents nP which is send in the encrypted form.

Algorithm 1: - Left to right binary method

Input: k is integer and P is point on curve

Output: Q is final point on curve which we need to be find.

```
    Q=P
    for j = n-1 to 0
        Q = point_double(Q)
    if$k_j$ = 1
        Q = point_add(Q, P)
Return Q
```

1.3.2 Right to left binary method

Let there is point P on elliptic curve and we have to evaluate nP for some integer value of n. In Right to left algorithm for scalar multiplication the integer n is used in the binary form and it is parsed from right to left. The process involves point addition and point doubling. In binary representation of n while iterating for 0 and 1 from left to right first check if value is 1. If it is 1 then point addition is done followed by point doubling. Else only point doubling is done. Final value of P then represents nP which is send in the encrypted form.

Algorithm 2:- Right to Left binary Method

Input: k is scalar and P is point on curve

Output: Q is final point on Elliptic curve which we need to be find.

```
    Q=O(Infinity)
    for j = 0 to n-1 do
    If $k_j$=1 then
        Q=Q+P //Addition of Point P and Q
    P=2P //Doubling of Point P
Return Q
```

1.3.3 Addition Subtraction using NAF (Non-Adjacent Form)

In this method to evaluate nP , n is represented in the non adjacent form[6]. In NAF representation, two consecutive numbers in the series are never non zero numbers. NAF is in the form -1.0,1. It means the value of NAF exists in {-1,0,1}.

Example: -

$(0\ 1\ 1\ 1\ 1)_2 = 0 + 8 + 4 + 2 + 1 = 15$

$(1\ 0\ 0\ 0\ {-1})_2 = 16 + 0 + 0 + 0 -1 = 15$

Algorithm 3.1: - Calculation of NAF of any Integer

Input:- $A = (a_{m-1}\ a_{m-2} \cdots a_1\ a_0)_2$

Output:- $Z = (z_m z_{m-1} \cdots z_1\ z_0)_{NAF}$

```
    i=0
    while 0 < A do
    if A % 2 == 1 then
        $z_i$ =2 − (A mod 4)
        A = A − $z_i$
    else
        $z_i$ = 0
        A = A/2
        i =i + 1
return z
```

In the addition subtraction method addition evaluates a positive point P and subtraction evaluates negation of P so cost of evaluation is reduced since cost of evaluation of addition and subtraction is practically same. It does addition or subtraction based on the sign of current digit of the non-adjacent form (NAF) of the scalar, scanned from left to right.

Algorithm 3.2 :-Addition Subtraction using NAF

Input:- K is integer and P is point on curve

Output:- Q is another point on Elliptic curve which we need to be find.

```
    A[ ] =NAF(k) //Value of NAFof k is stored in array
    Q = P
    for j = n - 1 till 0
        Q = 2Q
    if ( Aj = 1)
        Q = Q + P
    else if (Aj = -1)
        Q = Q - P
    endif
Return Q
```

1.3.4 Window method

When we have extra memory then we have a option to use window method where we can enhance the efficiency of scalar multiplication by using some pre computed points. There is window size of width w, consecutive bits of the binary form is parsed and we replace it by pre-computed table value. We first select a window size w, and then we compute values of cP where c = 0,1,2,.....,2^{w-1}-1. This algorithm uses the representation of k in the base of 2^w. More details about the algorithms can be read at [7].

Algorithm 4:- Windowed method

```
    Q = 0 (Point at Infinity)
    for i from m to 0
      for j in 1 to w
      Q = 2Q
    If $d_i > 0$
  Q = Q + $d_i$P //use precomputed $d_i$ value
```

1.3.5 w-ary non-adjacent form NAF method

In this algorithm, we do not compute dP for all the values of d in range 0 to 2^{w-1}-1, rather we only need to compute half the values. It uses Point Subtraction along with Point Addition.

Let us take one example where w=3 the precomputation is done for the digits set T={1,3,5,7} which is 1, 3….2^{w-1}-1. The number d can be converted from binary to w-NAF form using the following algorithm.

Algorithm 5.1:- w-ary non-adjacent form(wNAF)

```
  Input: width w, an integer d of n bits
  Output:wNAF form of a number c
    j = 0
    while (c > 0) do
      if (c % 2) = 1 then
        $c_j$ = c mods $2^W$
        c = c − $c_j$
      else
        $c_j$ = 0
        c = c/2
      j = j + 1
  return $(c_{j-1}, c_{j-2}, \ldots, c_0)_{wnaf}$
```

mods method:

```
  if (c % $2^W$) >= $2^W$−1
    return (c %$2^W$) − $2^W$
  else
  return (c % $2^W$)
```

*Algorithm 5.2:- Scalar Multiplication using wNAF *

```
  Input: P, $(d_{i-1}, d_{i-2}, \ldots, d_0)_{wnaf}$
  Output: Q
    Q = 0 (Point at Infinity)
    for c = i − 1 to 0 do
      Q = point_double(Q)
      if ($d_c$ != 0)
        Q = Q + $d_c$G
  return Q
```

1.3.6 Montgomery ladder method

This method calculated point multiplication in fixed amount of time. This method also used binary representation of scalar **k**. Reference [7] presents more details of this method.

Algorithm 6: Montgomery ladder

```
  Input: Point P, k(scalar) in binary form
  Output:kP
    X = 0 and Y = P
    for i = i-1 to 0 do
      ifki= 0
        Y = point_add(X, Y)
        X = point_double(X)
      else
        X = point_add(X, Y)
        Y = point_double(Y)
  return X
```

II. RELATED WORK

S. Rahimi and A. Mirghadri[8] in their paper entitled 'Classification and Comparison of Scalar Multiplication Algorithms in Elliptic Curve Cryptosystems' presented on comparison of various algorithms used for implementing scalar multiplication in elliptic curve cryptography. They have computed execution time for algorithms such as binary method, Montgomery's ladder, sliding window method etc. They have also discussed about endomorphism, which plays an important part in theory of elliptic curves. They have compared the results obtained from various algorithms.

E.Karthikeyan[9] in his paper entitled 'Survey of Elliptic Curve Scalar Multiplication Algorithms' presented on ECC multiplication algorithms, has firstly discussed about ECC . His paper explains various algorithms available to perform scalar multiplication in ECC and briefly analyses them. He concludes the areas for improvement in scalar multiplication field.

III. OBSERVATIONS& RESULTS

This paper is based on study and comparison of various algorithms used to implement scalar multiplication in elliptic curve cryptography. This cryptography is based on characteristics of elliptic curve.

Consider a point P on the elliptic curve and an scalar value k. A point Q = kP has to be calculated which serves as public key in encryption. Comparative analysis of various algorithms used for scalar multiplications is shown graphically. The algorithms are compared based on various parameters such as time of execution, doubling operations, precomputation required and hamming weight of integer k.

The graphs shown in the following observations are of graph: $y^2 = x^3 + 4x^2 + 3 \bmod 607$

and,

P=(234,121)

k=99999999

w = 4 (in case of windowed and wNAF method)

The methods are compared based on the time needed to compute the Scalar Multiplication, the addition operations performed, the doubling operation performed, precomputation needed, and the hamming weight of scalar.

3.1 Time required

It can be seen that binary right to left and left to right methods have maximum time of execution among others. And windowed NAF algorithm has least time of execution for some value of k. The execution time of windowed method and addition subtraction method is midway of binary and NAF methods. So wNAF method is best in terms of time of execution according to the used case.

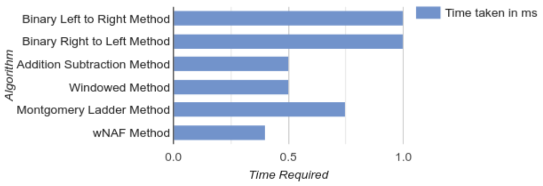

Fig 2: Comparison of Algorithms based on time taken in ms

3.2 No. of addition operations performed

In below chart, we can see that number the number of addition operation in montgomery ladder method is most. The addition operation is performed least in wNAF followed by windowed method and Addition Subtraction method.

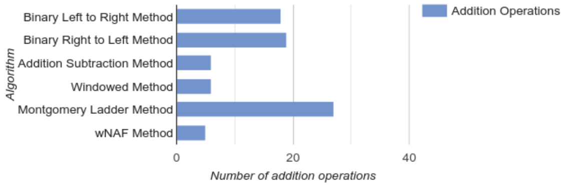

Fig 3: Comparison of Algorithms based on number of addition operations

3.3 No. of doubling operations required

In below chart, we can see that number the number of doubling operations performed by each method are almost similar with least in wNAF.

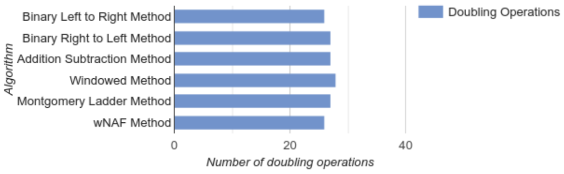

Fig 4: Comparison of Algorithms based on number of doubling operations

3.4 Hamming Weight of k

Hamming weight of k is defined by count of symbols other than zero in symbolic representation of k. For scalar multiplication the integer k may be represented in binary form, NAF or wNAF. In binary form, hamming weight is equal to the number of ones. Most algorithms perform more operations on non-zero element in the form used.

Hence having less hamming weight would reduce the complexity of the problem. As we can see, wNAF representation has least hamming weight in our case, followed by Addition Subtraction method that uses NAF form, followed by other algorithms that uses the binary representation of k.

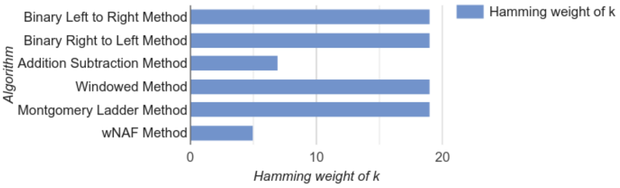

Fig 5: Comparison of Algorithms based on hamming weight of scalar k

3.5 Precomputation Required

Precomputation defines the act of computation done before the algorithm is executed to avoid repeated execution of any task by storing results in lookup table. On comparing the algorithms based on precompution required, it can be observed that binary method, addition subtraction method and montgomery ladder method perform a fixed amount of computation but in case of windowed method, the computation depends on the selected width (2^w) but wNAF requires only half computation as compared to windowed method. It needs 2^{w-1}computations where w is the width.

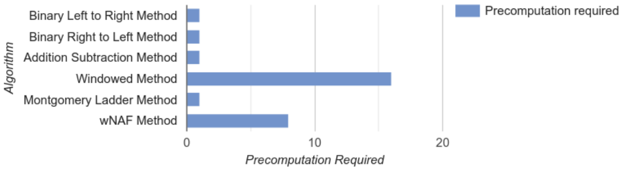

Fig 6: Comparison of Algorithms based on the precomputation require

IV. CONCLUSION

Different algorithms used for Scalar Multiplication have been compared. If precomputation is not an issue, then the algorithm with least number of overall operations can be selected. The hamming weight of scalar k contributes to the time taken by the algorithm. Hence, depending on the requirements and resources available the suitable algorithms can be selected.

REFERENCES

1. V.S. Miller, "Use of Elliptic Curves in Cryptography", Advances in Cryptology — CRYPTO '85 Proceedings, vol. 218, pp.417- 426, 1985.
2. N. Koblitz, "Elliptic Curve Cryptosystem", Mathematics of Computation, vol. 48, no. 177, p. 203, 1987.
3. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
4. M. Rosing, Implementing elliptic curve cryptography, Manning, Greenwich, CT, 1999.
5. D.R. Hankerson, S.A. Vanstone, A.J. Menezes, Guide to elliptic curve cryptography, Springer, New York, 2004.
6. F. Morain and J. Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains," RAIRO - Theoretical Informatics and Applications, vol. 24, no. 6, pp. 531–543, 1990.
7. P. L. Montgomery, "Speeding the Pollard and Elliptic Curve Methods of Factorization," Mathematics of Computation, vol. 48, no. 177, p. 243, 1987.
8. S. Rahimi and A. Mirghadri , "Classification and Comparison of Scalar Multiplication Algorithms in Elliptic Curve Cryptosystems ," International Journal of Computer & Information Technologies (IJOCIT), pp. 2345-3877, 2014.
9. E. Karthikeyan, "Survey of Elliptic Curve Scalar Multiplication Algorithms ," Int. J. Advanced Networking and Applications, vol. 4 no. 2, pp.1581-1590, 2012.