International Journal of Innovative Research in Advanced Engineering (IJIRAE)
Volume 1 Issue 1 (March 2014)

Survey: Elliptic Curve Cryptography using Scalar Multiplication Algorithms

[1]kaalidoss Rajamani, **[2] Dr.A.Arul L.S**
[1]Research Scholar/CS, Bharathiar University, India
[2]Department of Computer Science, Bharathiar University, India
[1]Kalidoss.shc@gmail.com, [2]aarul72@hotmail.com

Abstract- Stopping unauthorized access to corporate information systems is crucial for many organizations. In which Communication security is playing one of the key area of interest to protect the sensitive/valuable data. The data used in communication is very sensitive/valuable and needs to be protected and made abstract from intruders of system or over the network. The recent way to provide precious security mechanism of Network security is Cryptography using Elliptic Curve architectures which is based on the arithmetic of elliptic curves and discrete logarithmic problems. ECC schemes are public-key based mechanisms that provide Cipher text (Encryption), digital signatures and key exchange algorithms. The most crucial operation in the cryptosystem is the scalar multiplication operation. In this paper, we study various scalar multiplication algorithms with respect to the efficiency, weight and features etc. This paper gives an idea about algorithms and the areas where we need to researchers can proceed further in the computation of cryptosystem.

Keywords - ***Encryption, Authentication, Cryptography, scalar multiplication***

1. INTRODUCTION

In the year 1985, Miller [1] and Koblitz [2] separately proposed elliptic curve cryptography (ECC) and it is widely accepted in cryptosystem as an alternative tool to the conventional cryptosystems like RSA [3], DSA [4] and DH [5]. The primary reason of the attractiveness of ECC over the conventional systems is that it offers equivalent and enough security for smaller key sizes. For example, 160-bits of ECC and 1024-bits of RSA/DSA/DH offer the same level of security which is merit to become beneficial in applications where bandwidth, processing capacity, power availability and storage is limited. Such applications are smart cards, cellular phones, PDA, sensor networking and beepers (refer [6] for more details). Elliptic curve based protocols are dominating other cryptosystem in cryptography such as Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Integrated Encryption Scheme (ECIES). In all ECC protocols scalar multiplications are playing as the key role in the calculation in cryptography. The speed of scalar multiplication plays an important role in deciding an efficiency of the whole system in those protocols. In particular, fast multiplication is more crucial in some environments such as e-commerce servers where the large number of key agreements or signature generations occurs, and handheld devices with low computational power are required. There has been extensive research to compute scalar multiplication efficiently based on various representations of the multiplier [7].

Cryptography: Cryptography is a foundation of the modern electronic security technologies used today to protect valuable information resources on intranets, extranets, and the Internet so that cryptography is the essential science of providing security for information.

Objectives of Cryptography: The objective of cryptography is providing security to protect information resources by making unauthorized acquisition of the information or tampering with the information more costly than the potential value that might be gained. Because the value of information usually decreases over time, good cryptography based security mechanism to protect information until its value is significantly less than the cost of illegitimate attempts to obtain or tamper with the information. Good cryptography, when properly implemented and used, makes attempts to violate security cost-prohibitive. Another objective of all information security systems, including cryptography-based mechanism security systems are to protect information resources at less cost than the value of the information that is being protected

Security Functions of Cryptography

Cryptography is most often associated with the confidentiality of information that it provides. However, cryptography can offer the following four basic functions.

Confidentiality: Assurance that only authorized users can read or use confidential information. For an example, unauthorized users might be able to intercept information, but the information is transmitted and stored as cipher text and is useless without a decoding key that is known only to authorize users.

Authentication*:* Verification of the identity of the entities that communicate over the network. For example, online entities can choose to trust communications with other online entities based on the other entities ownership of valid digital authentication credentials.

Integrity: Verification that the original contents of information has not been altered or corrupted. Without integrity, someone might alter information or information might become corrupted, and the alteration could be undetected. For example, an intruder might covertly alter a file, but change the unique digital thumbprint for the file, causing other users to detect the tampering by comparing the changed digital thumbprint to the digital thumbprint for the original contents.

Non repudiation: Assurance that a party in a communication cannot falsely deny that a part of the actual communication occurred. Without non repudiation, someone can communicate and then later either falsely deny the communications entirely or claim that it occurred at a different time.

For example, without non repudiation, an originator of information might falsely deny being the originator of that information. Likewise, without non repudiation, the recipient of a communication might falsely deny having received the communication. Communications with other online entities based on the other entities ownership of valid digital authentication credentials.
Integrity: Verification that the original contents of information has not been altered or corrupted. Without integrity, someone might alter information or information might become corrupted, and the alteration could be undetected. For example, an intruder might covertly alter a file, but change the unique digital thumbprint for the file, causing other users to detect the tampering by comparing the changed digital thumbprint to the digital thumbprint for the original contents.
Non repudiation: Assurance that a party in a communication cannot falsely deny that a part of the actual communication occurred. Without non repudiation, someone can communicate and then later either falsely deny the communications entirely or claim that it occurred at a different time. For example, without non repudiation, an originator of information might falsely deny being the originator of that information. Likewise, without non repudiation, the recipient of a communication might falsely deny having received the communication. The rest of the paper is organized as follows. In section-1, the objective of the cryptography, the terminology is used in encryption and decryption and security function of the cryptography are discussed. Insection-2 mathematical background of ECC is discussed. Insection-3 the Elliptic Curve Digital Signature Algorithm, Elliptic Curve Diffie Hellman and Elliptic Curve Integrated Encryption are discussed. In section-4 we have discussed various Algorithms for Elliptic Scalar Multiplication which are used to find the computation of kp value. In section5, the comparision among the alogrithms are discussed and listed out its different features of algporithms. In chapter-6 the conculsionhas been discussed to express the best algoritm which is proposed to find out the value of kp which is being used in the cryptography alogrithms.

II. MATHEMATICAL BACKGROUND OF ECC

An elliptic curve can be presented as the set of solutions for the equation $y^2 \equiv x^3 + ax + b \pmod p$ where a, b є Zp such that $4a^3 + 27b^2$ # 0, including the point infinity O. The efficiency of elliptic curve cryptography algorithm is decided by various factors like selecting finite field (prime / binary); coordinate representations (Affine, Projective, Jacobian, Chudnovsky Jacobian and Modified Jacobian coordinate), elliptic curve arithmetic, scalar representation etc. More details about the mathematical aspect of ECC are available in [25, 26].The most important basic elliptic curve arithmetic operations are elliptic curve addition (ECADD) and elliptic curve doubling (ECDBL).The formula to compute the same is given in the Table-1 with respect to the affine coordinate system. Let P = (x1, y1) & = (x2, y2). Point addition will be made when P # Q, and if P = Q, then point doubling operation will be carried out.

Operation	Formula(Affine Co-ordinate System)
ECADD	$x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$,and d $= \frac{y_2 - y_1}{x_2 - x_1})$
ECDBL	$x_3 = \lambda^2 - 2x_1$, $y_3 = \lambda(x_1 - x_3) - y_1$,and d $\lambda = \frac{3x1^2 - \alpha}{2y_1}$

Table-1

The result of “addition” (P+Q) or “doubling” (2P) of points on the elliptic curve will always be one more point on the curve and one elliptic curve addition operation over E (Fp) takes one inversion, two multiplications, one squaring and six addition. Similarly, doubling requires one inversion, two multiplications, two squaring and eight addition operations. The number of arithmetic operations is varying for each coordinate system. Let us consider the elliptic curve over Fp where a = 1, b = 6, p = 11 with the equation $y^2 \equiv x^3 + x + 6 \pmod{11}$. The set of solutions are E = {(2,4), (2,7), (3,5), (3,6), (5,2), (5,9), (7,2), (7,9), (8,3), (8,8), (10,2), (10,9), O}, including the point infinity O. Choose P = (2, 4) and Q = (10, 9) and the elliptic curve point addition is performed as follows. λ = (9-4)/ (10-2) mod 11= 2 P+Q = (2, 4) + (10, 9), $x_3 = 2^2 – 2 – 10 = –8 = 3$, $y_3 = 2 (2-3) – 4 = -2 – 4 = -6 = 5$ P+Q = (2, 4) + (10, 9) = (3, 5),Select the point P = (8, 8) and the doubling operation is done as follows such as $\lambda = (3 * 8^2 +1)/ (2 * 8) \bmod 11 = (6 / 5) \bmod 11= (50 / 5) \bmod 11 = 10$, $x3 = 10^2 – 2 * 8 = 84 \bmod 11= 7$, y3 = 10(8-7) -8 = 10 – 8 = 2 2P = (8, 8) + (8, 8) = (7, 2).The outcome of addition and doubling is (3, 5) and (7, 2), because the elliptic curve points are Abelian group. In case of affine coordinate system, every ECADD/ECDBL requires an inversion operation which is expensive than other operations such as addition, subtraction and multiplication. The projective coordinate requires only one inversion at last but it takes some extra memory for storing temporary values. In scalar multiplication the ECADD and ECDBL are important factor in any operation to calculate kp value.

III. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography (ECC) is a public key cryptography. In public key cryptography each user or the device taking part in the secure communication basically have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations for communication. Only the particular user knows the private key whereas the public key is distributed to all users or device taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices or users taking part in the communication and domain parameters in ECC are the example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography.

3.1 ECDSA - Elliptic Curve Digital Signature Algorithm

For authenticating a device or a message sent by the device, Signature algorithm is used. For example consider two devices A and B. To authenticate a message sent by A and signs the message using its private key. The device A sends the message and the signature to the device B. This signature can be verified only by using the public key of device A for communication. Since the device B knows A's public key, it can verify whether the message is indeed send by device A or not. ECDSA is a variant of the Digital Signature Algorithm (DSA) that operates on elliptic curve groups for sending a signed message from A to B and both have to agree up on Elliptic Curve domain parameters. The domain parameters are defined in section Elliptic Curve. Sender 'A' have a key pair consisting of a private key dA (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key QA = dA* G (G is the generator point, an elliptic curve domain parameter).

3.2 ECDH - Elliptic Curve Diffie Hellman

ECDH is a key agreement protocol that allows two parties for communication to establish a shared secret key that can be used for private key algorithms. Both parties exchange some public information to each other for secure communication. Using this public data and their own private data these parties calculates the shared secret. Any third party, who doesn't have access to the private details of each device, will not be able to calculate the shared secret from the available public information to misuse. An overview of ECDH process is defined below. For generating a shared secret between A and B using ECDH, both have to agree up on Elliptic Curve domain parameters. The domain parameters are defined in section Elliptic Curve. Both end have a key pair consisting of a private key d (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key Q = d * G (G is the generator point, an elliptic curve domain parameter).

3.3 ECIES -Elliptic Curve Integrated Encryption System

Integrated Encryption Scheme (IES) is a hybrid encryption scheme which provides enough security against an adversary who is allowed to use chosen-plaintext and chosen-Ciphertext attacks. The security of the scheme is based on the Diffie–Hellman problem. The elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based on encryption algorithm. It is called integrated, since it is a hybrid scheme that uses a public key system to transport a session key for use by a symmetric cipher. ECIES is a public-key encryption algorithm [11].

IV.ALGORITHMS FOR ELLIPTIC SCALAR MULTIPLICATION

In all the protocols that are discussed (ECDH, ECDSA, ECAES), the most time consuming part of the computations are scalar multiplications. The calculations of the form which is shown as follows. Q= k P = P + P + P… k times, here P is a curve point, k is an integer in the range of order of P (i.e. n). P is a fixed point that generates a large, prime subgroup of $E(F_q)$, or P is an arbitrary point in such a subgroup. Elliptic curves have some properties that allow optimization of scalar multiplications. The following sections describe some efficient algorithms for computing kP.

4.1 Non Adjacent Form (NAF)

This is a much efficient method used in the computation of kP. Here, the integer k is represented as $k = \sum_{j=0}^{l-1} k_j 2^j$, where each kj $\in$ {–1, 0, 1}. The weight of NAF representation of a number of length l is $l/3$. Given below is an algorithm-1 for finding NAF of a number[20].Computation of the NAF of a scalar is shown in Table-2.

```
Algorithm-1: Non Adjacent Form (NAF)
NAF(k)Comment: Returns u[] which contains the NAF representation of k
Begin
        c ← k, l ← 0
        While (c > 0)
        BeginWhile
                If (c is odd)
                BeginIf
        u[l] ← 2 – (c mod 4), c ← c – u[l]
                Else
                        u[l] ← 0
                EndIf
                c ← c/2, l ← l + 1
    EndWhile
    Return u
    End
```

No of iterations	C	*L*	U
1	8	0	0
2	4	1	0
3	2	2	0
4	1	3	1

Therefore, the value of 8 in NAF form is (1 0 0 0). (Note that no two consecutive digits are non-zero) and note that in this method when pass odd number as an input, the outcome of the NAF is not similar to binary representation of that number. Where as while pass even number, the outcome of the NAF is similar to binary representation of that number. The above table is indicating the procedure to obtain the kP from the Non Adjacent Form(NAF). In this method almost hamming code is minimum when u calculate using binary method.

4.2 Binary Method

In Scalar multiplication is the computation of the form Q= k P where P and Q are the elliptic curve points and k is an integer. This is achieved by repeated point addition and doubling operations. To calculate the above, integer k is represented as $k=k_{n-1}2^{n-1}+k_{n-2}2^{n-2}+ . . . + k_1+k_0$ where $k_{n-1}=1$ and $k_i \in \{0, 1\}$, i = 0, 1, and 2…, n-1.This method is called binary method [8], which scans the bits of *k* either from left-to-right or right-to-left. The Algorithm-2 given below illustrates the computation of KP using binary method. The cost of multiplication depends on the length of the binary representation of k and the number of 1s in this representation. The number of non-zero digits is called the Hamming Weight of scalar. In an average, binary method requires (n-1) doublings and (n-1)/2 additions. For each bit '1', we need to perform ECDBL and ECADD, if the bit is '0', we need only ECDBL operation. So if we reduce the number of 1s in the scalar representation or hamming weight, the speed of elliptic curve scalar multiplication will improve. The binary representation of 687 is= $(1010101111)_2$. , the hamming weight is 7. By NAF hamming weight of k is reduced from 7 to 5 such as tow addition operations are going to be saved.

4.3 Addition-Subtraction Method

```
Algorithm-2: Binary Method
Input: Binary representation of k and point P
Output: Q =kP
Q=P
    For i = n-1 to 0 do
        Q = 2Q (Doubling)
        If k_i = 1 then
            Q = Q + P (Addition)
Return Q
```

In 1951, Booth [9] proposed a new scalar representation called signed binary method and later Rietweisner [10] proved that every integer could be uniquely represented in this format. The property of this representation is that, of any two consecutive digits, at most one is non-zero. Here the integer k is as $k=\sum_{j=0}^{l-1} k_j 2^j$ where each $k_j \in \{1, 0, 1\}$ Rietweisner's canonical representation is called non-adjacent form (NAF) [11]. NAF of a positive integer is at most one bit longer than the binary representation of the same. The **Algorithm-3 is used for converting an integer k into the signed binary representation of the same**.

```
Algorithm-3: Computation of NAF of an integer
Input     : Positive integer k
Output: s (NAF representation of k)
c = k ;l= 0
While(c>)
If (c is odd)
s[l] =2 – (c mod 4) c =c[l]
Else
s[l] = 0
End If
      c = c/2;
      l = l + 1
End While
Return s
```

The average hamming weight of signed binary representation is (n/3) and it has the lower hamming weight than the binary representation. For example, the binary representation of 1527 is $(10100000111)_2$ and the hamming weight is 5 and NAF of 1527 is $(10\text{-}100000\text{-}100\text{-}1)_2$and the hamming weight is only 4. The hamming weight of k is reduced from 5 to 4 which lead to the improvement in the scalar multiplication. One notable property of elliptic curve group is that the inverse of a point can be computed virtually free. This is the reason why a signed representation of scalar is meaningful. The binary method is revised accordingly and the new algorithm-4 is called addition-subtraction method [12, 13] given below. The algorithm-4 performs *n* doublings and *n/3* additions in an average. The disadvantage of the addition- subtraction method is that it is necessary to complete the recoding and stores them before starting left-to-right evaluation stage. Hence it requires additional *n*-bit memory for the right-to-left exponent recoding.

```
Algorithm-4: Addition-Subtraction method
Input: k and P
Output: Q = kP
s[ ] =NAF(k) /* The NAF of k is stored in s */
Q = P
  For j = n–1t 0 Q = 2Q
  If ($s_j$ = 1)
Q = Q + P
Else If ($s_j$ = –1)
Q = Q – P
End If
Return Q
```

4.4 Repeated Doubling Method

A point on the elliptic curve over F_{2m} is represented in the form of (x, λ) rather than in the form of (x, y) when using the repeated doubling method for scalar multiplication. Every point $P = (x, y) \in E(F_{2m})$, where $x \neq 0$, P can be represented as the pair (x, λ), where $\lambda = x + y/x$.Th algorithm-5 is as given below. Where f is the function for doubling, P is the coordinate to double; n is the number of times to double the coordinate. Example: 100P can be written as 2(2(P+2(2(2(P+2P))))) and thus requires six doublings and two additions. 100P would be equal to f (P, 100). This algorithm requires $\log_2(n)$ iterations of point doubling and addition to compute the full-point multiplication. There are many variations of this algorithm .It can be seen that we save one field multiplication in each of the iterations.

```
Algorithm-5: Repeated Doubling method
Repeated-doubling(P, i)
Comment: Returns Q = $2^i$P
Begin
$\lambda \leftarrow x + y/x$
For j = 1 to i – 1
BeginFor
        $x_2 \leftarrow \lambda^2 + \lambda + a$
        $\lambda_2 \leftarrow \lambda^2 + a + b/( x^4 + b)$
        $x \leftarrow x2, \lambda \leftarrow \lambda_2$
EndFor
        $x_2 \leftarrow \lambda_2 + \lambda + a$
        $y_2 \leftarrow x_2 + (\lambda + 1)x_2$
        $Q \leftarrow (x_2, y_2)$
        Return Q
End
```

4.5 Window method

When we are allowed to use extra memory, the window method further enhances the efficiency of scalar multiplication by using table of pre-computed points. A window is a combination of consecutive columns such that the number of columns is less than or equal to *w* where *w* is the width. In this method *w* consecutive bit of binary representation is scanned and get replaced by pre-computed table value. Let us consider the window method with width *w*=4, the necessary pre-computation is made according to the digits set T= {1, 3, 5, 7} that is 1….2^{w-1}-1. During the recoding stage, the binary exponent is getting replaced as follows: 1|1→0|3, 1|0|1→0|0|5, and 1|1|1→ 0|0|7. The conversion from binary to the window can be performed left-to-right or right-to -left as well. The result may differ syntactically, but there is no change in the non-zero density of those representations. The window based signed binary representation is called *w*NAF, first described in [15] has minimal hamming weight than any other scalar representations is shown in algorithm-6. The property of *w*NAF is that the most significant non-zero bit is positive, among any *w* consecutive digits, at most one is non-zero; each non-zero digit is odd and less than 2^{w-1} in absolute value *w*NAF is computed only from the least significant bit that is right-to -left. So we need to compute and store the *w*NAF representation of the multiplier before starting scalar multiplication. The drawback of *w*NAF is that it is not possible to merge the exponent recoding and the evaluation stage and it seems impossible to compute *w*NAF left-to-right. However, in connection with memory constraint devices left-to-right recoding schemes are by far more valuable

```
Algorithm-7: Scalar multiplication using window method
Pre-computation Stage: Input: point P and width w
Output: Pi = iP; i = 1…$2^{w-1}$– 1
P1 = P, X = ECDBL (P1)
For i = 3 to $2^{w-1}$ - 1, step 2 do
Pi = ECADD(X, $P_{i-2}$), Return Pi; i = 1 … $2^{w-1}$ – 1
Evaluation Stage:
Input: point P and k (k is an integer)
wNAF = $sk_n|sk_{n-1}|...|sk_0$ of $d$
Pi :=iP; i = 1… $2^{w-1}$ - 1
Output X = O
for  i = n down to 0 do X= ECDBL(X)
if $k_i$> 0 then
X= ECADD(X, $P_{ki}$) else if $k_i$< 0 then
X= ECADD(X, $-P_{|kj|}$)  Return X
```

The average density of non-zero bits is asymptotically 1/ (w+1) for $n\rightarrow\infty$, and the digit set equals T= {±1, ±3... ± (2^{w-1}-1)} which seems to be minimal. For example, the binary representation of 2927 is $(101101101111)_2$ and *w*NAF representation of 2927 is $(005005005007)_2$.The hamming are 9 and 4 respectively. So the wNAF is an optimal representation of the scalar [19]. The Algorithm-7, [15] does the necessary pre-computations and then kP is computed using window method.

4.5 Double-And-Add Method

This reduced computational complexity by applying one constant inversion operation, regardless of the number of doubling. The complexity is given as (4r + 1) M + (4r + 1) S + I, where M, S and I denote a multiplication, a squaring and an inversion in Fq, respectively is shown algorithm-8.

```
Algorithm-8 Double-and-add method using left-to-right binary method

Input: $k = (k_{t-1}… k1, k_0)2$, $P \in E(\mathbb{F}_q)$.

Output: $kP$.
$Q \leftarrow \infty$.
For $i$ from $t - 1$ down to 0 do
   $Q \leftarrow 2Q$.
   If $k_i = 1$, then $Q \leftarrow Q + P$.
Return ($Q$)
```

4.6 Mutual Opposite Form (MOF)

The left-to-right recoding method eliminates the need for recoding and storing the multiplier in advance. Joye and Yen [16] first proposed the left-to-right recoding algorithm in the year 2000. In CRYPTO 2004, Okeya [17] proposed a new efficient left -to-right recoding scheme called mutual opposite form (MOF). The properties of new singed binary representation such as signs of adjacent non-zero bits (without considering 0 bits) are opposite, Most non-zero bit and the least non-zero bit are 1 and -1, respectively, All the positive integers can be represented by unique MOF. The following Algorithm-9 is a simple and flexible conversion from n-bit binary string k to (n+1)-bit MOF. This algorithm converts the binary string to MOF from the most significant bit efficiently. MOF representation of an integer is highly flexible because, the conversion from right-to-left and left-to-right is possible. Applying window method on MOF can further minimize the non -zero density of MOF. The *w*MOF is the first window based signed recoding scheme that can be performed from the most significant bit. As in the case of elliptic curve scalar multiplication a left-to-right evaluation is the natural choice, *w*MOF enables to merge recoding and evaluation stage. Hence there is no need to store the recoded scalar earlier. TheAlgorithm-10 will make use of the pre-computed table to generate wMOF of k from left-to-right.

```
Algorithm-9: Left-to-Right Generation from Binary to wMOF
Input: widthw, n-bit binary string $k=k_{n-1}|k_{n-2}|...|k_1|k_0$

Output: wMOF of k ($sk_n|sk_{n-1}|...|sk_0$)
$k_{-1}$ = 0; $k_n$ = 0 i = 0
        while i ≥ w - 1 do
        if $k_i = k_{i-1}$ then
```

```
            $sk_i = 0; i = i - 1$
        else
        {The MOF windows begins with a non-zero digit left-hand
        }
            $(sk_i, sk_{i-1}, ..., sk_{i-w+1}) \leftarrow Table_{w\ SW}(k_{i-1} - k_i, ki-2 - ki-1, ..., ki-w - ki-w+1)$
= i – w
if $i \geq 0$ then
            $(sk_i, sk_{i-1}, ..., sk_0) \leftarrow Table_{i+1SW}(k_{i-1} - k_i, k_{i-2} - k_{i-1}, ..., k_0 - k_1, - k_0)$
        return $(sk_n, sk_{n-1}, ..., sk_0)$
```

```
  Algorithm-10: Left-to-Right Scalar multiplication using wMOF(On the Fly)
  Input     : Point P, n-bit binary string $k = k_{n-1}| k_{n-2}|...|k_1|k_0$
  Output: k P
  $k_{-1}=0; k_0= 0$
    i=e+1 for the largest e with $k_e\#0$
    If $k_{i-2}= 0$ then
            Q = P ; i = i–2;
    Else {$k_{i-2}= 1$}
            Q=ECDBL(P); i=i-2;
    While $i \geq 1$ do
            If $k_{i-1}= k_i$ then Q=ECDBL(Q); i=i-2;
            Else { $k_{i-1}\# k_i$ }
Q=ECDBL(Q)
If $(k_i ,k_{i-2} ) = (1,1)$ then
Q=ECDBL(Q);

Q=ECADD(Q,-P)
Else if $(k_{i,},k_{i-2} ) = (1,0)$ then
Q=ECADD(Q,-P); Q= ECDBL(Q)
Else if $(k_{i,},k_{i-2} ) = (0,1)$ then
Q=ECADD(Q,P); Q= ECDBL(Q)
Else if $(k_{i,},k_{i-2} ) = (0,0)$ then
Q= ECDBL(Q); Q=ECADD(Q,P)
i=i-2
If i=0 then
Q= ECDBL(Q); Q=ECADD(Q, $-k_0$ P)
Return Q
```

The average non-zero density of wMOF is also 1/ (w+1) for $n \rightarrow \infty$. Every non-negative integer *k* has a representation as *w*MOF, which is unique except for the number of leading zeros. The Algorithm-10 merges the recoding and evaluation stage of the scalar multiplication. The advantage of above algorithm is that it reduces the memory requirement since it does not store the converted representation of d; instead it is used directly in the scalar multiplication.

4.7 Shamir Method - Parallel Computation

Some public key cryptographic protocols such as the verification of digital signature, self-certified signature scheme requires the computation of powers of two, three, or more. ECDSA verification requires the computation of aP + bQ, where a and b are integers and P and Q are elliptic curve points. Normally this is done by computing aP and bQ individually and add result finally. Shamir proposed a method [18] to compute aP + bQ simultaneously. The signed binary representations of a pair of integers are written one below another, the number of non-zero columns is defined as the "joint weight". The joint weight of a and b determine the speed of the computation. For example, the joint weight of 57 and 22 in binary expansion is 6, since57= $(111001)_2$ and 22= $(010110)_2$ and the joint weight are 5.

The signed binary representation of 57 is $(100\text{-}1001)_2$ and 22 is (10-10-10). The computation of aP + bQ using Shamir method is illustrated in the Table-3. It is clear that the number of additions required is depends on the joint weight of a and b and the number of point doublings required is one less than the number of bits in a or b. Thus minimizing joint weight would speed up the computation. This method costs n doublings and 2n/3 additions on average. The simple pre-computations like P+Q and P-Q would improve the speed called fast Shamir method given in Algorithm-11

A	1	0	0	-1	0	0	1
B	0	1	0	-1	0	-1	0
Double	0	2P	4P+2Q	8P+4Q	14P+6Q	28P+12Q	56P+22Q
+P	P						57P+22Q
-P				7P+4Q			
+Q		2P+Q					
-Q				7P+3Q		28P+11Q	

Table–3: Shamir method to compute aP + bQ

```
Algorithm-11: Shamir Method to compute aP + bQ
Input:  NAF (a), NAF (b) and P,Q
Where a,b are integers and P,Q are points
Output:aP + bQ
P1=P,P2=Q,Q=0,R=P1+P2,S=P1-P2
for i=n 1 to 0do Q=2Q
if $(a_i,b_i)$ # (0,0)
if $(a_i,b_i)$ = (1,0) then Q=Q+P1
elseif $(a_i,b_i)$ = (-1,0) then Q=Q-P1
elseif $(a_i,b_i)$ = (0,1) then Q=Q+P2
elseif $(a_i,b_i)$ = (0,-1)  then Q=Q-P2
elseif $(a_i,b_i)$ = (1,1) then Q=Q+R
elseif $(a_i,b_i)$ = (-1,-1) then Q=Q-R
elseif $(a_i,b_i)$ = (1,-1) then Q=Q+S
elseif $(a_i,b_i)$ = (-1,1) then Q=Q-S
End
RetunQ
```

4.9 Joint Sparse Form (JSF)

Solinas [18] presented a right-to-left method called Joint Sparse Form (JSF) for computing the signed binary representation of a pair of integers, which results in minimal joint weight than Shamir's method. The property of JSF is that, the average joint weight among all JSF representations of two n-bit integers is n/2. Of any three consecutive positions, at least one is a double zero Adjacent terms do not have opposite signs, that is X_j,X_{j+1} # -1 and Y_jY_{j+1} # -1 If X , X_{j+1} # - 1, then Y_{j+1} = ±1 and Y_j = 0 If Y_j, Y_{j+1} # - 1, then X_{j+1} = ±1 and X_j = 0 .The Algorithm-10 is used for generating joint sparse form of pair of integers called JSF of integers [18, 19].The output of the algorithm-12 can be used in fast Shamir method, Algorithm-9 to compute aP+bQ and now it costs *l* doublings and *w* additions, where *l* is the length and *w* is the hamming weight.

4.10 Direct Doubling

Among the various elliptic curve arithmetic operations, point doubling is quite costlier than point addition in the scalar multiplication over affine coordinate system. Sakai and Sakurai [20] proposed a scalar multiplication algorithm using direct computation of several doublings (which computes 2^kP directly from P) without computing the intermediate points 2P, 2^2P, $2^3P…2^{k-1}P$. The concept of direct computation of 2^kP was first suggested by Guajardo and Paar[12]. The new doubling formula is re-constructed as below.

$$A_1=x_1$$
$$B_1=3x_1^2 + a$$
$$C_1= -y_1,$$
$$D1=12A_1 C_1^2 - B_1^2$$
$$x_2 = B_1^2 – 8A_1 C_1^2 / (2C_1)^2$$
$$y_2 = 8C_1^4 – B_1 D_1 / (2C_1)^3$$

The computational complexity of this formula is (5S + 5M + I) and the existing method has the complexity of (6S + 4M + I). The complexity of the direct computation versus separate doubling is given in Table-4.From the Table-3, it is found that to compute 2^kP requires at most 4k+1 squaring, 4k+1 multiplication and only one inversion when compared to the separate k doubling which requires 2k squaring, 2k multiplication and k inversions. As we know that the inversion is the costliest arithmetic operation in elliptic curve. The direct computation of 2^k P using affine coordinate system is shown in the following table-4. The result shows that direct computation with 160-bit size takes 18.4 ms to compute 2^kP, but binary method takes 26.8 ms for the same computation. The performance of the direct computation may be further improved by new recoding method such as MOF.

The basic idea of the method is to process on k column by column. First, kd−1P + . . . + kid2^{id}+...+ Kwd−12 is calculated and doubled. Second, kd−2P +. . . + kid−1$2^{id-1}$+. . . +kwd−22 is calculated and added to the results of the first calculation and the final result is doubled [29]. The procedure will be continued like this which is shown in the algorithm-13. Finally, k0P + . . . + kid2^i+ . . . +k

$(w-1)d2^{(w-1)d}$ is calculated and added to the previous sum. The obtained result represents kP. In order to accelerate the computation process, one can compute $[aw-1. . . a2, a1, a0]$ P = $aw-12^{(w-1)d}+. . .+a12^{d}P+a0P$ for all possible values of string (aw−1, . . . , a2, a1, a0). The exact algorithm-14 gives clear picture over that process. Probability of $[k_i^{w-1}, k_i^{w-2}, \ldots, k_i^{2}, k_i^{1}]$ is a one can compute $[aw-1. . a2, a1, a0]$ P = $aw-12^{(w-1)d}+ . . . +a12^{d}P+a0P$ for all possible values of string zero. Hence, it is a non-zero array with a probability of $1 - 1/2^{w}$. The non-infinite addition in line 2.2occurs expectedly $[\![((2]\!]^{\uparrow}(w-1)\ /2^{w}$)*d-1) times (−1 comes from the first infinity addition).Also, in line 2.1 a non- infinity doubling is executed (d − 1) times (−1 comes from the first infinity doubling). In the case of additional memory being available for the algorithm, two columns can be executed simultaneously. The idea is to divide the columns of k into two parts. Apply the previous procedure simultaneously for both sides. This idea is illustrated in Algorithm-15 Similar to Algorithm-14, the expected time of Algorithm-15 can be calculated as $[\![((2]\!]^{\uparrow}(w-1)\ /2^{w}$ *2e-1) A+ $(e-1)^{D}$. On the other hand, Algorithm-15, it requires much storage for pre computation as Algorithm-14. If memory is limited, the two algorithms can be compared for a given fixed amount of pre computation.

Multiplication methods are included into many standards, e.g. NIST, IEEE, ISO, and ANSI. The methods, studied in the previous two sections, are also applicable to the cryptography in this section. Moreover, generic methods are used with fast computation methods in order to decrease the computation time of the scalar multiplication [32].

```
Algorithm-12: Computation of simple joint Sparse Form
Input     : x and y are integers
Output: JSF of (x and y)
j=0
While x # 0 or y # 0 do
  $x_j$ = x mod 2, $y_j$ = y mod 2
  if$x_j$ =1 and $y_j$ = 1 then

    If(x - $x_j$) / 2 ≡1 (mod 2) then
        $x_j$ = - $x_j$

    end if
    if (y - $y_j$) / 2 ≡1 (mod 2) then
        $y_j$ = - $y_j$
    end if
  else if $x_j$ # $y_j$ then
      if (x - $x_j$) / 2 ≡ (y - $y_j$) / 2 (mod 2) then
        $x_j$ = - $x_j$, $y_j$ = - $y_j$ end if
  end if
 x = (x - $x_j$) / 2, y = (y - $y_j$) / 2 j = j + 1
End while
```

4.11 Fixed-Base Comb Method

This method is also called as Lim-Lee Method. Let d = $\left[\frac{l}{w}\right]$. If it's required, pad on the left side of the representation with zeros and then, separate k into w concatenated parts: k = $K^{w-1}. . .K^{0}$. kP= $\sum_{i=0}^{i-1} k_i 2^{i} P$

= $k0P + k12P + k24P + ... + kt-12^{l-1}P$

= $k0P + ... + kd-12^{d-1}P + k(w-1)d2^{(w-1)d}P+ ... + kwd-12^{wd-1}P$

$k0P + ... + (kd-1P)\ 2d^{-1}+ k\ (w-1)\ d2^{(w-1)d}P+ ... + (kwd-12^{(w-1)d}P)2^{d-1}$.

Algorithm-15.

Fixed-base comb method point multiplication with two tables, Input: w, d= [l/w], e = [d/2], k= $(k_{l-1} \ldots . k_0)2 = k^{w-1} \| \ldots k^{0}$ where k^{i} is d-bit- integer and k_s^{i} is s^{th} bit of k^{i} ,P€E(F_q)

Pre-computed values: [aw−1, a2, a1, a0]P and 2^{e}[aw−1, . . . , a2, a1, a0]P for all

Possible string (aw−1. . . a2, a1, a0) Output: kP

```
1. $Q = \infty$
2. for i from e-1 down to 0 do
 2.2Q=Q+[$K_I^{W-1}, K_I^{W-2}, \ldots . K_I^{2}, K_I^{1}$]+$2^{8}$[$a_{w-1}, \ldots, a_2, a_1, a_0$]P
3. Output (Q)
```

Algoriths Features	is Addition required	Is Subtraction	is doubling required	Is sign of digit required	Are pre-computed points required	Is iteration required	Is squaring required	Is Multipli cation	Is Weight of NAF required	Is inversion required
Non adjacent form (NAF)	✓	✓	×	×	×	✓	×	×	×	✓
Binary Method	✓	×	✓	×	×	✓	✓	✓	×	✓
Addition-subtraction Method	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
Repeating Doubling Mthod	✓	×	✓	×	✓	✓	✓	✓	✓	✓
Window Method	✓	×	✓	×	✓	✓	×	✓	✓	✓
Double and Add Method	✓	×	✓	×	×	✓	✓	✓	×	✓
Mutual Opposite Form	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
Shamir Method-Parallel Computation	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
Joint Sparse Form	✓	×	✓	✓	✓	✓	✓	✓	✓	✓
Direct Doubling	✓	✓	✓	×	×	✓	✓	✓	×	✓
Fixed-Base Comb with Window-NAF Method	✓	✓	✓	×	✓	✓	✓	✓	✓	✓

Table-4:comparison among different alogrithms

5. Comparision

In this paper, various methods has been discussed to calculate kp value for ECC,and aslo the follwing comparision Table-4 express the various features among the discussed alogrithms and also various reperesentation of different alogrithm and how it plays a role in the scalar multiplication are discussed .

VI. CONCLUSION

ECC becomes to be cryptosystem for the future, one way to improve the performance of such cryptosystem is to use an efficient method which is the most time consuming operation. Even though elliptic curve based cryptographic algorithms were widely accepted and used in many applications operation called scalar result and still there are many areas available to improve. In this paper, the dominant multiplication is analyzed in many factors. Since from the inception, so many algorithms were proposed. Various representation of multiplier is presented and how it plays a role in the scalar multiplication is also discussed and at last Fixed-Base Comb with Window-NAF Method is current proposed method for scalar multiplication when we compare to remaining algorithms in this paper.

REFERENCES

[1] V.S. Miller, "Use of Elliptic Curves in Cryptography", Advances in Cryptology, Proceedings of CRYPTO’85, LNCS-218, pp. 417-426, 1986.
[2] N.Koblitz, "Elliptic Curve Cryptosystem", Mathematics of Computation, Vol.48, pp.203-209, 1987.
[3] R.L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public key Cryptosystems", Communications of the ACM, Vol. 21, pp. 120-126, 1978.

[4] T. ElGamal, "Public key Cryptosystem and a Signature Scheme based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. IT-31, No. 4, pp. 469-472, 1985.
[5] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644-654, 1976.
[6] J. Lopez and R. Dahab, "An Overview of Elliptic Curve Cryptography", Technical Report, Institute of Computing, State University of Combinas, Brazil, 2000.
[7] D.M.Gordon, "A Survey of Fast Exponentiation Methods", Journal of algorithms, vol.27, pp.129-146, 1998.
[8] A.J. Menezes and S.A Vanstone, "Elliptic Curve Cryptosystems and Their Implementations", Journal of Cryptology, Vol.6, No. 4, pp.209-224, 1993.
[9] A.D.Booth, "A Signed binary multiplication technique", Journal of Applied Mathematics, Vol. 4.No. 2, pp.236-240, 1951.
[10] G.W.Reitwiesner, "Binary Arithmetic", In Advances in computers, Academic Press, Vol.1, pp.231-308, 1960.
[11] F.Morain and J.Olivos, "Speeding up the computations on an Elliptic curve using Addition-Subtraction chains", RAIRO Theoretical Informatics and Applications, Vol.24, pp.531-543, 1990.
[12] J.Gujardo and C. Paar, "Efficient Algorithms for Elliptic Curve Cryptosystems", Advances in Cryptology-CRYPTO'97, Springer-Verlag LNCS, vol.1294, pp.342-356, 1997.
[13] IEEE P1363, Standard Specifications for Public-Key Cryptography, 2000.
[14] Hankerson, D.; Menezes, A.; Vanstone, S. Guide to Elliptic Curve Cryptography; Springer: New York, NY, USA, 2004.

[15] K.Koyama and Y.Tsuruoka, "Speeding up Elliptic Cryptosystem by Using a Signed binary Window Method", Advances in Cryptology – CRYPTO '92, LNCS – 740, pp.345-357, 1993.
[16] M.Joye and S.Yen, "Optimal Left-to-Right binary signed digit recoding", IEEE Transactions on Computers, Vol. 49, pp. 740-748, 2000.
[17] K.Okeya, "Signed binary representations revisited", Proceedings of CRYPTO'04, pp.123-139, 2004.
[18] J.A. Solinas, "Low-Weight binary representations for pairs of integers", Technical Report CORR 2001-41, Center for Applied Cryptographic Research, University of Waterloo, Canada, 2001.
[19] Seo, H.; Kim, H.; Park, T.; Lee, Y.; Liu, Z.; Kim, H. Fixed-Base Comb with Window-Non-Adjacent Form (NAF) Method for Scalar Multiplication. Sensors2013, 13, 9483-9512.
[20] Seo H, Kim H, Park T, Lee Y, Liu Z, Kim H. Fixed-Base Comb with Window-Non-Adjacent Form (NAF) Method for Scalar Multiplication. Sensors. 2013; 13(7):9483-9512.
[21] V.S.Dimitrov and G.A. Jullien, "Loading the bases: A new number representation with applications", IEEE Circuits and Systems Magazine, Vol. 3. No. 2, pp.6-23, 2003.
[22] V.S.Dimitrov, G.A. Jullien, and W.C. Miller, "An algorithm for modular exponentiation", Information Processing letters, Vol. 66. No. 3. pp.155-159, 1998.
[23] M.Ciet, M.Joye, K.Lauter, and P.L.Montgomery, "Trading inversions for multiplications in elliptic curve cryptography", Cryptology ePrint Archive, Report 2003/257, 2003.
[24] K.Eisentrager, K.Lauter, and P.L.Montgomery, "Fast elliptic curve arithmetic and improvedweil pairing evaluation", Topics in Cryptology–CT-RSA 2003, vol. 2612, Springer-Verlag LNCS, pp. 343-354, 2003.
[25] MichaelRosing, Implementing Elliptic Curve Cryptography, Manning Publications, 1998.
[26] DHankerson, A. J Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer, 2004.
[27] Cohen, H.; Frey, G.; Avanzi, R.; Doche, C.; Lange, T.; Nguyen, K.; Vercauteren, F. Handbook ofElliptic and Hyperelliptic Curve Cryptography; Taylor and Francis Group, LLC: Boca Raton, FL,USA, 2006.
[28] Seo, Hwajeong; Kim, Hyunjin; Park, Taehwan; Lee, Yeoncheol; Liu, Zhe; Kim, Howon. 2013. "Fixed-Base Comb with Window-Non-Adjacent Form (NAF) Method for Scalar Multiplication." Sensors 13, no. 7: 9483-9512.
[29] m. bellare, j.a. garray, and t. rabin, "fast batch verification for modular exponentiation and digital signatures," advances in cryptology eurocrypto'98, vol. 1403 of lecture notes in computing science, pp 236-250. springer-verlag, 1998.
[30] M. Bellare, J.A. Garray, and T. Rabin, "Fast Batch verification for modular exponentiation and digital signatures," Advances in Cryptology eurocrypto'98, vol. 1403 of Lecture Notes in Computing Science, pp 236-250. Springer-Verlag, 1998.
[31] M. Joye and S. M. Yen, "Optimal left-to-right binary signed-digit recoding," IEE Transactions on Computers, vol.49, and issue: 7):pp.740-748, 2000.
[32] M.Joye and S.Yen, ""Optimal Left-to-Right binary signed digit recoding," IEEE Transactions on Computers, Vol. 49, pp. 740-748, 2000.
[33] Saeed Rahimi,Abdolrasoul ,Mirghadri "Classification and Comparison of Scalar Multiplication Algorithms in Elliptic Curve Cryptosystems" International Journal of Computer & Information Technologies (IJOCIT)