Fast Scalar Multiplication in ECC Using The Multi Base Number System

G. N. Purohit[1], Asmita Singh Rawat[2]

[1] Aim & Act, Department of Mathematics, Banasthali University
Jaipur, Rajasthan,304022, India

[2] **Aim & ACT,** Department of Computer Science, Banasthali University
Jaipur, Rajasthan,304022, India

Abstract

As a generalization of double base chains, multibase number system is very suitable for efficient computation of scalar multiplication of a point of elliptic curve because of shorter representation length and hamming weight. In this paper combined with the given formulas for computing the 7- Fold of an elliptic curve point P an efficient scalar multiplication algorithm of elliptic curve is proposed using 2,3 and 7 as basis of the multi based number system . The algorithms cost less compared with Shamirs trick and interleaving with NAFs method.

Keywords: *Scalar multiplication, Elliptic curve, Double base number system, Multibase number system, Double chain, Septupling.*

1. Introduction

Public key cryptography has been widely studied and used since Rivest, Shamir and Adleman invented the cryptography or cryptosystem RSA [1] in 1975. The system heavily depends on integer factorization problem [IFB] using large key bits of the order 1024 bits or 2048 bits . Later on Diffie- Hellman [2] developed the public key exchange algorithm using the discrete logarithmic problem [DLP]. Elgammel also used DLP in encryption and digital signature authentication [DSA] scheme. However, these conventional public key cryptographic systems, such as RSA and DSA are impractical in WSNs due to low processing power of sensor nodes. Koblitz [3] and Miller [4] independently used elliptic curves for cryptography using Elliptic curve Discrete Logarithmic Problem [ECDLP] and provided elliptic curve cryptographic [ECC].

In recent years ECC has received increased acceptance and has been included in standards room bodies such as ANSI, IEEE, ISO and NIST. Compared to traditional cryptographic systems like RSA, ECC offers smaller key sizes and more efficient arithmetic, which results in faster computation, lower power consumption as well as memory and band width savings. Thus ECC is especially useful for mobile constrained devices like WSN, which enables wireless mobile devices to perform secure communication efficiently and establishes secure end to end connections.

In ECC, points on elliptic curves over finite fields are used to generate finite abelian groups to implement public key cryptographic primitives. Cryptosystems in ECC are based on the group of points on an elliptic curve over a finite field. They rely on the difficulty of finding the value of a scalar, given a point and the scalar multiple of that point. This corresponds to solving the discrete logarithm problem. However, it is more difficult to solve the Elliptic curve DLP than its original counterparts. Thus elliptic curve cryptosystems provide equivalent security as the existing public key cryptosystems, but with much smaller key lengths. In addition another benefit is that each user may select a different curve E even though the underlying field *K* remains the same for all users. Thus the hardware which depends on the field remains the same and the curve E can be changed periodically for extra security. Traditionally ECCs has been developed over finite fields which have either prime order or binary fields of order 2^m. The fundamental operation for generating a finite abelian group over an elliptic curve is the addition of two points on it. If point P on EC is added to itself *(k-1)* times then we obtain a new point *kP* on elliptic curve and *kP* is termed as the scalar multiplication of point *P* by scalar *k.* Among the many arithmetic operations like addition, inversion, scalar multiplication involved in ECC, the scalar multiplication is the most important, energy and time consuming operation. A key factor for its fast implementation in ECC is to

compute the scalar multiplication efficiently, when *k* is a large integer. Various fast algorithms have been proposed for this purpose. Traditionally the integer *k* is represented in binary form and the double and add method is applied to calculate *kP*.

In this paper we first compute the 7-fold of an elliptic curve point P, i.e. 7P. The formulas of doublings (2P), tripling(3P), triple and add(3P)+P, quadrupling(4P) , quadruple and add(4P)+P and quintupling(5P) are available in literature. Double base number representation of an integer in bases {2,3}, {2,5} and {3,5} and their generalizations to triple base representation base {2,3,5} was recently reported in [5].

In this paper, an efficient scalar multiplication algorithms of a point P on an elliptic curve is proposed using triple base representation of the scalar using 2,3 and 7 as basis of the multibase number system. We obtain a sparser representation of the scalar, and the present algorithm costs less compared to the existing algorithms. We restrict our work on non super-singular elliptic curves defined over the field F_2m , however this can be suitably modified for any other type of elliptic curve.

The rest of the paper is organized as following: In the next section we report the related work. In Section-3 we evaluate sep-tupling $7P = (x_7, y_7)$ of a point $P = (x, y)$ and calculate its cost in terms of multiplications, squaring and inversions. The costs of addition and subtraction are ignored which are negligible in comparison to other costs. The triple base representation of an integer is in section 4. Multi base number representation (MBNR) and multi base chain representation and their implementation in scalar multiplication are discussed in section 5 and 6 respectively. Concluding remarks are given in the end.

3. Related Work

The classical approach of representing the integer k in binary form and then performing the scalar multiplication by a standard double and add method has efficient triple (3P) and double(2P) of point P, a ternary (binary approach for fast scalar multiplication is presented in [6]. For general curves a DBNS representation of the scalars using 2 and 3 as bases has been proved quite efficiently [7]. For last couple of years double base number system [DBNS] has been proposed to be used in this context by several authors [8, 9, 10, 11]. In search of sub linear scalar multiplication algorithms, authors of [8] have been used complex bases, 3 and τ for Koblitz curves. As a new approach for fast scalar multiplication, point halving was proposed independently by Knudsen [12] and Schroeppel [13]. They suggested that point doubling in the double and add method can be replaced by a faster point halving operation. A detailed analysis of the speed advantage of employing point halving instead of point doubling is available in [9]. Further point halving can be combined with frobenius endomorphism so as to speed up the corresponding operation in Koblitz curve by 25 percent [14, 15]. In yet another development the double base number representation of integer was generalized to multibase number representation with 2, 3 and 5 as basis elements and is included in [16,17]. The efficient scalar multiplication using multibase number representation included in [16] which also includes quintuple formula. Multibase multiplication using MBNR is included in [17], Scalar multiplication combining MBNR with point halving is discussed in [18].

Our contribution in this paper is computing 7 fold (7P) of an elliptic curve point P for a curve over binary field and using the same in scalar multiplication. The scalar multiplication uses the representation of the scalar as sum/ difference of product of powers 2, 3 and 7.

4. Septupling

In this section we consider Sep-tupling (7P) of a point P on an elliptic curve. We begin with a discussion of an elliptic curve.

4.1: Elliptic Curve

An elliptic curve over a finite field GF (Galois field) K is defined by an equation

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

where $a_1, a_2, a_3, a_4, a_5, a_6 \in K$ are the parameters of the curve and $\Delta \neq 0, \Delta$ being the discriminate of the curve *E.*

In the case of binary field $K = F_2m$, the non- super singular curves are used for cryptography, whose Weierstrass equation can be simplified to the form.

$$y^2 + xy = x^3 + ax^2 + b \quad \text{Where} \quad a, b \in F_2m \text{ and } \Delta = b \neq 0. \quad (2)$$

If $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are two points on E (F_2m) then their sum $P + Q$ is also a point (x_3, y_3) on E , where x_3 and y_3 are given by.

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \quad (2)$$

where $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$

Further double of P i.e. $2P$ is also a point (x_4, y_4) on curve E, where

$$x_4 = \mu^2 + \mu + a = x_1^2 + \frac{b}{x_1^2}$$

$$y_4 = x_1^2 + \mu x_4 + x_4,$$

$$\mu = x_1 + \frac{y_1}{x_1}$$

The usual scalar multiplication kP of P by scalar k is obtained using the above described two operations add and double. For example 25P is calculated as

$$25P = 2(2(2(P + 2P)) + P$$

or

$$2(2(2(2P) + P)) + (2(2P) + P)$$

These group operations in affine coordinates required field inversion besides multiplication and squaring. We denote by i 's and m the cost of one inversion , one squaring and one multiplication respectively. The cost of additions of two points $P + Q$ and of double of a point P, $2P$ are equal and equals to $i+2m$. However we shall neglect the cost of field additions in case of elliptic curves over binary fields. It may be noted that cost of squaring in case of binary fields is almost free. The cost of a repeated doubling $w - DBLP = 2^w P$ is $+(4w-2)m$ as reported in [8]. The costs of : (*i*)double and add, $DA(P,Q) \to 2P \pm Q,$

repeated tripling, $wTPR(P) \to 3^w P$ and *iii)* triple and add $TA(P,Q) \to 3P \pm Q$ are given in [8] as *i) i+am, ii) i+7m and (iii) 2i+am*, respectively.

4.2 Point Septupling

Let P be *(x,y)* be a point on an elliptic curve given by equation () over a binary field. We shall calculate the 7-fold of P given by, $7P = (x_7, y_7)$, that is we shall obtain expression for x_7 and y_7 in terms of x and y .

For non-super singular curves over a binary field, the division polynomials are given by

$$\Psi_1 = 1$$

$$\Psi_2 = x$$

$$\Psi_3 = x^4 + x^3 + a = A$$

$$\Psi_4 = x^6 + ax^2 = x^2(A - x^3) = B$$

The higher degree division polynomials are obtained using the following recurrence relations:

$$\Psi_{2n+1} = \Psi_{n+2}\Psi_n^3 - \Psi_{n-1}\Psi_{n+1}^3$$

$$\Psi_2\Psi_{2n} = \Psi_{n+2}\Psi_n\Psi_{n-1}^2 - \Psi_{n-2}\Psi_n\Psi_{n+1}^2$$

Using these relations we obtain

$$\Psi_5 = \Psi_4 x^3 - \Psi_3^3 = Bx^3 - A^3 = C$$

$$\Psi_6 = \frac{\Psi_5\Psi_3 x^2 - \Psi_3\Psi_4^2}{x} = \frac{CAx^2 - AB^2}{x} = D$$

$$\Psi_7 = \Psi_3 + 2\Psi_3^3 - \Psi_2\Psi_4$$
$$= A + 2A^3 - xB^3 = E$$

$$\Psi_8 = \frac{\Psi_6\Psi_4\Psi_3^2 - \Psi_2\Psi_4\Psi_5^2}{\Psi_2}$$
$$= \frac{DBA^2 - xBC^2}{x} = F$$

For any point $P(x,y)$ on E, its n-fold $n(3P)$ is given by

$$[n]P = (x + \frac{\Psi_{n+1}\Psi_{n-1}}{\Psi_n^2}, y + x + \frac{\Psi_{n+1}\Psi_{n-1}}{\Psi_n^2} + \frac{\Psi_{n+1}^2\Psi_{n-2}}{\Psi_2\Psi_n^3} +$$

$$(x^2 + y)\frac{\Psi_{n+1}\Psi_{n-1}}{\Psi_2\Psi_n^2})$$

So the value of (x_7, y_7) for the 7-fold over binary field. Thus one can be computed from the above equation as follows:

$$x_7 = x + \frac{\Psi_8\Psi_6}{\Psi_7^2}$$

$$x_7 = x + \frac{FD}{E^2}$$

$$y_7 = x + y + \frac{\Psi_8\Psi_6}{\Psi_7^2} + \frac{\Psi_8^2\Psi_5}{\Psi_2\Psi_7^3} + (x^2 + y)\frac{\Psi_8\Psi_6}{\Psi_2\Psi_7^2}$$

$$y_7 = x + y + \frac{FD}{E^2} + \frac{F^2C}{xE^3} + (x^2 + y)\frac{FD}{xE^2}$$

The cost of evaluating various polynomials defined above:

Polynomials	Operations
$A = x^4 + x^3 + a$	*2[s]+ 1[m]*
$B = x^6 + ax^2 = x^2(A - x^3)$	*1[m]*
$C = \Psi_5 = \Psi_4 x^3 - \Psi_3^3 = Bx^3 - A^3$	*1[s] +2[m]*
$D = \frac{CAx^2 - AB^2}{x}$	*4[m] +1[i] +1[s]*
$E = A + 2A^3 - xB^3$	*1[m] + 1[s]*
$F = \frac{DBA^2 - xBC^2}{x}$	*3[m] + 1[s]*
$\frac{FD}{E^2}$	*1[m] + 1[s] + 1[i]*
$\frac{F^2C}{xE^3}$	*1[s] + 4[m] +1[i]*

Thus the total cost of the hepta tupling is *3[i] + 7[s] + 18[m]*. Neglecting the cost of squaring (in case of EC over binary fields) the total cost turns out to be *3[i] + 18[m]*.We can also compute 7P as 2(2P)+3P or 2(3P)+P. Using the generic method the costs of TPL(P) and DBL(P) are respectively i+7m and i+2m. Further the costs of DA(P, Q) are 2i+9m. Hence the total cost *7P=2(3P)+P=4i+18m*. If we consider *7P* as *2(2P)+3P* then the total cost is *5i+20m.* Hence cost calculated by us is the least.

The following table represents the costs of different operations used for the efficient scalar multiplication using the binary field method.

Table 2 : table of costs for different operations

S.No.	Operations	Binary field costs
1	$P+Q$	$1I+1S+2M$
2	2*P*	1*I* + 1*S* + 2*M*
3	2*P*+*Q*	1*I*+2*S* + 9*M*
4	3*P*	1*I*+4*S* + 7*M*
5	3*P*+*Q*	2*I*+3*S* + 9*M*
6	4*P*	1*I*+5*S*+8*M*
7	5*P*	1*I*+5*S*+13*M*
8	7*P*	3*I*+7*S*+18*M*

5.Multibase Number Representation (MBNR)

First we review double base number system (DBNS)

5.1 DBNS

Improving the classical methods of double and add for scalar multiplication a new method (DBNS), using bases besides 2, were introduced [2, 3, 5]. In this system one can represent k as the sum of terms of the form $s_i 2^{bi}3^{ci}$, where $s_i \in \{-1,1\}$, such representation always exists and in fact this number system is quite redundant. One of the most interesting properties of the representation is that among all the possible representation for a given integer, some of them are really sparse, that is to say that the number of non-zero terms is quite low.

To compute DBNS representation of an integer, one usually uses a greedy algorithm. It consists of the following: find the closest integer of the form $2^{bi}3^{cj}$ to *k*, subtract it from *k* and repeat the process with $k' = k - 2^{bi}3^{ci}$ till it is equal to zero. Performing a point scalar multiplication using this number system is relatively easy. Letting k be equal to $\sum_{i=1}^{n} s_i 2^{bi}3^{ci}$ one just needs to compute [$s_i 2^{bi}3^{ci}$] *P* for *i*=1 to *n* and then add all the points.

Example:

$$895712 = 2^{10}3^6 + 2^9 3^5 + 2^8 3^4 + 2^7 3^3 + 2^6 3^2 + 2^5 3^1 + 2^4 3^0 + 2^1 3^0.$$

Even if the number of additions is quite low, in practice such a method requires too many doublings and triplings. For this reason the general DBNS representation has been considered to be not suitable for scalar multiplication.

To overcome this problem the concept of double base chains was introduced in [3]. In this system, an integer k is still represented as $\sum_{i=1}^{n} s_i 2^{bi}3^{ci}$, but with the restriction that allowing a Horner like evaluation of *kP* using only doublings and triplings, however, with significantly increase in the number of point additions.

5.2 Multibase Number Representations (MBNR)

Let $B = \{b_1, b_2, ..., b_l\}$ be set of small integers .A representations of integer *k* as a sum of powers of elements of B of the form $k = \sum_{j=1}^{m} s_j b_1^{cj1} .... b_l^{cjl}, s_j \in \{-1,1\}$ is called a multibase representations of *k* using the base B. The integer *m* is the length of the representation of *k* using the base B. The integer *m* is the length of the representation or Double base number system (DBNS) or double base number representation discussed in previous section.

(DBNR) is a special case with with $|B| = 2$. In this paper we are particularly interested in multibase representation with B= {2, 3, 7}. The multi base representations with B= {2, 3, 5} have been discussed by many authors [4, 6].Authors in [17] combined with MBNR with point halving.

The double base number system is highly redundant. Further these representation are very short in length, a 160bit integer can be represented using around 23 terms using the base B={2,3}. The results on length of DBNS representation are included in [2]. The multi base representation is even shorter and more redundant than the DBNS. The same 160 bit integer can be represented using around 15 terms using a triple base B= {2, 3, 5}.
Example:

$$895712 = 2^4 3^7 5^2 + 2^4 3^5 5^1 + 2^4 3^4 5^0 + 2^1 3^4 5^0 + 2^0 3^2 5^0 + 2^0 3^1 5^0$$

The multi base representation of a number using a triple base B= {2, 3,7} is even shorter and sparse as compared to its representation using the triple base {2,3,5}.

Example:

$$895712 = 2^9 3^5 7^1 + 2^7 3^3 7^1 + 2^5 3^1 7^1 + 2^5 3^1 7^0$$

.
In this article, unless otherwise stated, by a multi base representation of *k*, we mean a representation of the form.

$$k = \sum_i s_i 2^{bi} 3^{ci} 7^{di}$$

Where $s_i \in \{-1,1\}$ and the terms of the form $2^{bi} 3^{ci} 7^{di}$ will be termed as 3-integers. A general multibase representation although very short is not suitable for a scalar multiplication algorithm. So we include a special representation with restricted exponents.
Definition: A multi base representation $k = \sum_i s_i 2^{bi} 3^{ci} 7^{di}$ using the base B= {2, 3, 7} is called a step multibase representation (SMBR) if the exponents $\{b_i\}$,$\{c_i\}$ and $\{d_i\}$ form three separate monotonic decreasing sequence.

We consider an example illustrating this definition for the same number.

Example:

$$895712 = 2^5 3^4 7^3 + 2^4 3^2 7^2 - 2^3 3^0 7^2 - 2^3 3^0 7^0$$

An integer *k* has several SMBR, the simplest one being the binary representation. If *k* is represented in SMBR, then we can write it using Horner's rule and an addition chain, like double base chain in [1], for scalar multiplication can easily be developed. In case of our base system {2,3,7}, we require b_1 doublings, c_1 tripling and d_1 sep tuplings. An integer can be converted to a multi base representationwith base {2,3,7} using the Greedy Algorithm as:

GREEDY AlGORITHM:
while $k > 0$
let z be the largest integer $2^b 3^c 7^d$
Output(b,c,d)
replace k by k-z
$k - z \leftarrow 0$
else
end.

In this process the pre-computed points are extensively used to accelerate the scalar multiplication in applications where extra memory is available. So we have used a new method multi base chain representation in which one does not require any pre-computations but in this method the expansion of the scalar reduces the cost of the scalar multiplication making it faster.

The important contribution in [7] was the new ternary-binary method to perform the efficient scalar multiplication. Ciet et al.[7] have proposed a ternary-binary method for fast ECC scalar multiplication. It makes use of efficient doubling (2P), tripling (3P), quadrupling (4P). In this paper a new septenary /ternary /binary approach for fast ECC scalar multiplication is proposed, which makes the use of septupling (7P) for the efficient scalar multiplication.

In this base system only b_1 doublings, c_1 tripling and d_1 sep- tuplings are needed for the scalar multiplication; in the next section we give implementation of this method and develop Septupling, 7*P*, for point *P*.

6.Scalar Multiplication Implementation and Algorithm.

We have already suggested that an integer *k* can be represented in multi base number system as the sum or difference of the mixed powers of 2, 3 and 7, as given in the following equation

$k = \sum_i s_i 2^{bi} 3^{ci} 7^{di}$ with $s_i \in \{-1,1\}$ and $b_i, c_i, d_i \geq 0$

The sequence of the binary and ternary exponents decreases monotonically, i.e.

$b_1 \geq b_2 \geq b_3 \ldots \geq b_m \geq 0$, $c_1 \geq c_2 \geq c_3 \ldots \geq c_m \geq 0$ and $d_1 \geq d_2 \geq d_3 \ldots \geq d_m \geq 0$, and thus a multi base chain is formed.

For implementing the scalar multiplication we use a recursive formula for the fast computation of scalar multiplication using following equation for recursive calculations.

$K_1 = 1$, $K_i = 2^u 3^v 7^w K_{i-1} + s_i$ with $i \geq 2$, $s_i \in \{-1,1\}$,

where *u* is the difference of two consecutive binary exponents, *v* is the difference of two consecutive ternary exponents and *w* is the difference of two consecutive septenary exponents.

To implement it we have used the following algorithm.

An integer *k*, can be converted to a multi base representation

$k = \sum_i s_i 2^{bi} 3^{ci} 7^{di}$ with $s_i \in \{-1,1\}$ and $b_i, c_i, d_i \geq 0$ using greedy algorithm as already explained in Section.5 . Now we describe the algorithm:

ALGORITHM:

Input: An integer $k = \sum_{i=1}^{m} s_i 2^{b_i} 3^{c_i} 7^{d_i}$, $s_i \in \{-1,1\}$

And such that $b_1 \geq b_2 \geq b_3 \ldots \geq b_m \geq 0$, $c_1 \geq c_2 \geq c_3 \ldots \geq c_m \geq 0$ and $d_1 \geq d_2 \geq d_3 \ldots \geq d_m \geq 0$, and a point $P \in E(F_2 m)$.

Output : the point $kP \in E(F_2 m)$

$Z \leftarrow s_1 P$

for $i = 1, \ldots, m-1$ do

$u \leftarrow b_i - b_{i+1}$

$v \leftarrow c_i - c_{i+1}$

$w \leftarrow d_i - d_{i+1}$

if $u = 0$ then

$Z \leftarrow 7^w Z$

if $v \neq 0$ then

$Z \leftarrow 3(3^{v-1} Z) + s_{i+1} P$ // TA used here

else

$Z \leftarrow Z + s_{i+1} P$

else

$Z \leftarrow 7^w Z$

$Z \leftarrow 3^v Z$

$Z \leftarrow 2^{u-1} Z$ // DA is used here

$Z \leftarrow 2Z + s_{i+1} P$

Return *Z*

As an example for illustration of this algorithm we consider computing 895712P. We first develop the multi base chain as given below.

$895712 = 2^5 3^4 7^3 + 2^4 3^2 7^2 - 2^3 3^0 7^2 - 2^3 3^0 7^0$

and compute 127*P*, 2285*P*, 111964*P* and finally 895712*P* successively.

Table2:Method of calculating 895712*P* in different iterations for sep-tupling

i	K	s	u	v	w
1	1	1	0	0	0
2	$126 K_1 + 1 = 127$	1	1	2	1
3	$18 K_2 - 1 = 2285$	-1	1	2	0
4	^{49} $K_3 - 1 = 111964$	-1	0	0	2
5	$8 K_4 = 895712$	0	3	0	0

This algorithm has used a multibase representation of the scalar with 2, 3 and 7 as the base numbers and it uses group operation like ADD, DBL, w-DBL, DA, TA for efficient computation. The new multi base chain method and proposed septenary/ternary/binary method is much faster than any other methods mentioned above for scalar multiplication for the binary fields without requiring any pre computations.

7. Conclusion

In this paper we have presented fast and secure scalar multiplication algorithms. In our work we have proposed a new algorithm for MBNR representation of an integer and combining with the scalar multiplication. We have shown that the length of the MBNR is shorter than the

DBNR and is also more redundant, since the number of representation grows faster as the number of base element is higher. For the MBNR representation we have used 2, 3 and 7 as the bases which makes the representation sparser

8. References

[1]. Rivest, R. Shamir,A, & Adleman A. (1978). "A method for obtaining digital signature and public key cryptosystems". Communication of the ACM, vol.21, pp120-126

[2]. Diffie, W.Hellman, M.E. (1976). "New directions in cryptography", IEEE Transactions Information theory, IT-22(6).

[3]. Koblitz N.(1987). "Elliptic curve cryptosystems.Maths computing 48", (177) pp. 203-209

[4]. Miller, V. (1986). "Use of elliptic curves in cryptography .Advances in cryptology"- Crypto'85 pp 417-426.

[5] P. K. Mishra, V. S.Dimitrov. "Efficient Quintuple Formulas for Elliptic Curves and Efficient Scalar Multiplication Using Multibase Number Representation". Springer-Verlag, 2007, volume 4779, pages 390-406.

[6].F.Morian, J.olivos, (1990). "Speedong up computation on an elliptic curve using addition – subtraction chains", Information theory applications, vol.24, pp 531-543.

[7].M. Ciet, M. Joye, K. Lauter, P.L. Montgomery,(2003) "Trading Inversions for Multiplications in Elliptic Curve Cryptography",Cryptology ePrint Archive, Report 2003/257 . Also to appear in Design, Codes and Cryptography

[8] V. Dimitrov, L. Imbert and P.K. Mishra,(2005) "Efficient and Secure Elliptic Curve Point Multiplication using Double-Base Chains," Advances in Cryptology - ASIACRYPT'05, LNCS Vol. 3788, pp. 59-78, Springer-Verlag, 2005.

[9]. K.W. Wong, Edward, C.W.Lee, L.M.Cheng, Xiaofeng Liao,(2006), "Fast Scalar Multiplication using new Double Base Chain and Point Halving", Applied Mathematics and Computation.

[10].Avanzi, R.M and Sica, F(2006). "Scalar multiplication on Koblitz curves using Double bases". Technical Report Available at http://eprint.iacr.org/2006/067.

[11].V. Dimitrov ;K.V.Jarvanian ,M.J.Jacobian , W.F.Chan and Z.Huang,(2006). "FGPA implementation of point multiplication on Koblitz curves Using Kleinian integers". LNCS 4249 pp. 445-459, Springer Verlag.

[12].C.Doche and L.Imbert. "Extended Double Base Number Systems with applications to elliptic Curve Cryptography" , Available at http://eprint.iacr.org/2006/330.

[13].M.Ciet and F.Sica (2005). "An Analysis of Double base Number system and a sub linear scalar multiplication Algorithm". LNCS Vol.3715 pp. 171-182. Springer Verlag.

[14]. E.W.Knudsen (1999). "Elliptic scalar multiplication using point halving", LNCS Vol.1716 pp 135-149.

[15]. R.Schroeppel.(2000) "Elliptic curve Point Ambiguity Resolution Apparatus and method ." International patent Application Number PCT/US00 31014, field November 9.

[16].R.M.Avanzi,C.Henbergerand,,H.Prodinger(2005)."Minimality of the hamming weight of the τ-NAF for Koblitz Curve and Improves combination with point halving". Cryptology eprint archive, Report 2005/225.

[17].R.M.Avanzi, M. Ciet, F.Sica (2004). "Faster Scalar multiplication on Koblitz Curves Combining Point halving with the Frobenius Endomorphism" , LNCS Vol.2947, pp.28-40.

[18].A.M.Ismail, M.R.MD Said, K.A.Mohd Atan , I.S.Rakhimov(2010). "An Algorithm to enhance Elliptic Curves scalar Multiplication Combinig MBNR with point halving", Applied Mathematical sciences, Vol.4,pp.1259-1272.

[18].D.Bernstein, P.Birkner, P.Longa, C.Peters. "Optimizing Double Base Elliptic Curve Single Scalar Multiplication". LNCS Vol. 4859, pp.167-182, Springer Verlag.

[19].P.Longa (2007): "Accelerating the scalar multiplication on Elliptic curve Cryptosystems over prime fields" . Master thesis University Of
Ottawa, http://patriclonga.bravehost.com/publications.html.

[20] R. Dahab and J. Lopez (1998), "An Improvement of Guajardo-Paar Method for Multiplication on non Super Singular Elliptic Curves".In Proceedings of the XVIII International Conference of the Chilean Computer Science Society (SCCC'98), IEEE CS Press, November 12-14, Anto Fagasta, Chile, pp.91-95.

[21].V.S.Dimitrov,L.Imbert, and P.K.Mishra,(2005) " Fast elliptic Curve Point Multiplication using Double-Base Chain", Cryptology ePrint Archive , Report 2005/069.

.

Prof. G. N. Purohit. He is a Professor in Department of Mathematics & Statistics at Banasthali University (Rajasthan).Before joining Banasthali University, he was Professor and Head of the Department of Mathematics, University of Rajasthan, Jaipur. He had been Chief-editor of a research journal and regular reviewer of many journals. His present interest is in O.R., Discrete Mathematics and Communication networks. He has published more 40 research papers in various journals.

Asmita Singh Rawat received the B.Sc degree from University Of Lucknow and M.C.A degree from U.P Technical University in 2006 and 2009, respectively. She is currently working towards a PhD degree in computer
Science at the Banasthali University of Rajasthan. Her research interests include wireless sensor network security with a focus on the elliptic curve cryptography.