Int. J. Advanced Networking and Applications
Volume: 03 Issue: 05 Pages: 1329-1337 (2012)

Elliptic Curve Point Multiplication Using MBNR and Point Halving

G.N.Purohit
Department of Mathematics, Banasthali University
Jaipur, Rajasthan,304022, India
Email: *gn purohitjaipur@yahoo.co.in*

Asmita Singh Rawat
Department of Computer Science, Banasthali University
Jaipur, Rajasthan,304022, India
Email: singh.asmita27@gmail.com

Manoj Kumar
Department of Computer Science, KNIT, UPTU
Sultanpur, 228188 Uttar Pradesh, India.
Email: rajrajmanoj@gmail.com

----------ABSTRACT----------
The fast implementation of elliptic curve cryptosystems relies on the efficient computation of scalar multiplication. As generalization of double base number system of a number k to multi-base number system (MBNR) provides a faster method for the scalar multiplication is most important and costly operation (in terms of time) in ECC, there is always a need of developing a faster method with lower cost. In this paper we optimize the cost of scalar multiplication using halving and add method instead doubling and tripling methods. The cost is reduced from 40% to 50% with respect to the other fastest techniques

Keywords: Double base number system, Elliptic curve cryptography, multi-base number system, point halving, W SN.

Date of Submission: November 12, 2011 Date of Acceptance: January 18, 2012

1. INTRODUCTION

Public-key cryptography has been widely studied and used since 1975 when Rivest, Shamir, and Adleman invented RSA public key cryptography. This system heavily depends on integer factorization problem (IFP) using big key bits such as 1024 bits and 2048 bits. Later on Deffie- Hellman in [8] developed the public key exchange algorithm using the discrete logarithm problem (DLP). El Gamal also used DLP in encryption and digital signature scheme. Koblitz and Miller [13,14], independently used EC in cryptography using elliptic curves discrete logarithm problem (ECDLP) . ECC provide a high level of security with much smaller keys in comparison to other popular cryptosystems based on integer factorization. Improving the efficiency of scalar multiplication in EC is one of the main interests of many researchers in the field of cryptography. For this reason, ECC offers a security level equivalent to RSA and DSA while using a much smaller key size.

In any implementation of ECC primitives, scalar multiplication is the computationally dominant operation. Several methods have been proposed in the literature to speed-up point multiplication, which use various representations of the base point (affine coordinates, projective coordinates), various representations of the scalar (binary, ternary, NAF, w-NAF), and various curve operations (additions, doublings, halving, tripling). The computational cost (timing) of these curve operations depends on the cost of the arithmetic operations that have to be performed in the underlying field.

Many researchers have given more attention to develop the proposed ECC algorithms and improve their efficiency. Improving the efficiency of scalar multiplication in EC is one of the main interests of many researchers in the field of cryptography. Computationally the most expensive operation in ECC is Scalar Multiplication namely given an integer k, and a point P on an elliptic curve curve, the computation of $kP = P + \cdots + P$ is called scalar multiplication of point P by scalar k. A key factor for its fast implementation is how to compute the scalar multiplication kP efficiently. Generally the integer k is represented in binary form and the double and add method is applied to calculate kP. It is computed by series of doubling (ECDBL) and addition (ECADD) operation of the point P. A point multiplication is the first sequence of additions, several multiplications, squaring and inversion on a finite field. A strategy that has gained lots of attention in recent years is the use of representations of number k based on double-base and multi-base chains. The use of the so-

called Double Base Number System (DBNS) for cryptographic applications was first proposed by Dimitrov et al. in [3]. In the setting of ECC, double-base chains were first applied to the computation of scalar multiplication by Dimitrov et al. [4], and later extended to multi base chains by P.Longa et.al [11]. Dimitrov, Imbert, and Mishra [6] introduced double base chains $d_i 2^{ai} 3^{bi}$, where $d_i \in \{-1,1\}$ with the new restrictions $a_1 \geq a_2 \geq a_3 \geq \ldots$ and $b_1 \geq b_2 \geq b_3 \geq \ldots$ allowing a Horner-like evaluation of *kP* with only a_i doubling and b_i tripling. The double base number system is highly redundant.

Here we discuss the multi base representations (MBNR) which are even shorter and more redundant than the DBNS. The number of representations grows very fast with the number of base elements. The new Multi base Non-Adjacent Form (mbNAF) method was introduced and shown to speed up the execution of the scalar multiplication with an efficient use of multiple bases to represent the scalar. We present new improvements in the point operation formulae. Specifically, we reduce further the cost of composite operations such as quintupling in MBNR base {2, 3, 5 } and septupling in MBNR base
{2, 3, 7} of a point, which are relevant for the speed up of multi base methods.

In this paper, we propose a new multi-base chain representation for scalars to achieve faster scalar multiplication. In an earlier papers we have discussed a multi-base representation of a scalar in the base elements 2,3 and 7 [15] . The authors in [10,12,15] have proposed an halving method instead of doubling and quadrupling. In this method the scalar multiplication is done with a faster speed upto 39% [14] are even upto 50%[12] . Adopting this technique we modify the multi-base chain representation of scalar *k* in terms of monotonic decreasing powers of 1/2, 3 and 7. With this method, we remove point doubling and quadrupling and use point halving instead, while maintaining the tripling and septupling point operations.

The paper is organized as follows: In the next section, we recall some related work regarding the point (scalar) multiplication. In Section 3 some basic facts about elliptic curves and their equations developed on different fields. In Section 4 we introduce the double-base number system and double chain along with algorithm for computation. In Section 5 we introduce the concept of multi-base number system using the septupling algorithm and we propose a new scalar multiplication algorithm based on the multi-base number system. In the next Section 6 we have discussed the point halving method for scalar multiplication. We present numerical results and compare our algorithm with other algorithms discussed in [15], [18] .

2. RELATED WORK

Doche et al. [6] introduced a new method that also finds double-base chains without using the "Greedy" algorithm, although using a somewhat more complex search-based approach in comparison with the basic Multi-base NAF. The cost of this method is comparable to (2,3)NAF method, but slightly higher than that achieved by (2,3,5)NAF. More important, the proposed Multi-base NAF method presents even lower costs in all the cases, with bases (2,3) , (2,3,5) and (2,3,7). The improvement is especially significant in the case without pre-computations, which makes this method especially interesting for applications on constrained devices. M. Ciet et.al [1], discussed a ternary / binary approach making use of the efficient triple (3*P*) and double (2*P*) of point *P* for fast scalar multiplication. A similar idea was suggested in V.S Dimitrov et.al [7]. On the other hand, point halving was proposed independently by Knuden [12] and Schroeppel [17]. They suggested that point doubling in the double-and-add method can be replaced by a faster point halving operation. The idea was to replace almost all point doublings in double-and-add methods with a potentially faster operation called point halving. Knudsen [12] presented some rough analysis which suggests that halving methods could be 39% faster than doubling methods ([14] claims a 50% improvement), but these claims have not been supported by experimental evidence or by detailed analysis.

3. ELLIPTIC CURVE

EC-based cryptosystems can attain equivalent security levels to RSA with significantly smaller cryptographic parameters. For instance, it is widely accepted that 160-bit ECC offers security equivalent to 1024-bit RSA. This significant difference makes ECC especially attractive for applications in constrained environments as shorter key sizes are translated to less storage requirements and reduced computing times.

An elliptic curve over a field F_p is defined in terms of the solutions to an equation in F_p. The form of the equation defining an elliptic curve over F_p differs depending on whether the field is a prime finite field or a characteristic 2 finite field.

An elliptic curve over a finite field GF field K (Galois field) 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 Weierstrass equation of non- super singular curve can be simplified to the form.

$$y^2 + xy = x^3 + ax^2 + b \quad (2)$$

Where $a, b \in F_2m$ and $\Delta = b \neq 0$.

Let F_p be a prime finite field where p is an odd prime number, and $a, b \in F_p$ satisfying

Volume: 03 Issue: 05 Pages: 1329-1337 (2012)

$$4a^3 + 27b^2 \neq 0 \pmod{p}. \quad (3)$$

Then an elliptic curve $E(\mathrm{F}\ p)$ over F p with parameters $a, b \in F_p\ p$ consists of set points $P = (x,y)$ satisfying the equation:

$$y^2 = x^3 + ax + b \pmod{p} \quad (4)$$

Together with an extra point O called the point at infinity. The equation $y^2 = x^3 + ax + b \pmod{p}$ is called the defining equation of $E(\mathrm{F}\ p)$. For a given point $P = (x_p, y_p)$, x_p is called the x-coordinate of P, and y_p is called the y-coordinate of P. The addition of two points on the curve generates a third point on the curve.

3.1.Arithmetic Operations in an EC (Elliptic Curve)

3.1.1.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. Point addition is the addition of two points P(x_1, y_1) and Q(x_2, y_2) on an elliptic curve to obtain another point R(x_3, y_3) on the same elliptic curve. We can obtain point addition R as $P+Q$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$
$$\lambda = (y_1 + y_2)/(x_1 + x_2)$$

3.1.2 Point Doubling

Point doubling is the addition of a point on the elliptic curve to itself to obtain another point on the same elliptic curve(i.e. $Q=2P$). If P is (x_1, y_1) and Q=2P= (x_3, y_3) then coordinates of Q are given as .

$$x_3 = \lambda^2 + \lambda + a$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$
$$\lambda = x_1 + y_1 / x_1$$

3.1.3 Point Halving

Point halving can be seen as the reverse operation of point doubling [2]. We can define the elliptic curve point halving as: Let $P = (x, y), Q = (u, v)$ be points that belong to the curve, then we need to compute P such that $Q = 2P$ using the following equations:

$$\lambda = x + y / x \quad (5)$$
$$u = \lambda^2 + \lambda + a \quad (6)$$
$$v = x^2 + u(\lambda + 1) \quad (7)$$

The point $P = (x, y)$ is computed by solving the Eq.(6) for λ, Eq.(7) for x, and finally Eq.(5) for y. Point multiplication methods based on point halving share strategy with τ -adic methods on Koblitz curves in the sense that point doubling is replaced by a potentially faster operation. As with the efficiently computable endomorphism in, the improvement is not as dramatic as that obtained with methods for Koblitz curves, although halving applies to a wider class of curves. We restrict our attention to elliptic curves E over binary fields F_2m defined by the equation $y^2 + xy = x^3 + ax^2 + b$ where $a, b \in F_2m$, $b = 0$. The algorithm for computing ½ P (x, y) from P (u, v) is given below

Algorithm

Input: $2P = (u, v)$
Output: $P = (x, y)$

Solve $\lambda^2 + \lambda = u + a$ **for** λ
find $T = u + v(\lambda + 1)$
If $Tr(T) = 1$ **then** $\lambda = \lambda$, $x = \sqrt{T}$ **to** t **do.**
else
$\lambda = \lambda + 1$, $x = \sqrt{T + u}$
find $y = \lambda x + x^2$
return (x, y).

4. DOUBLE BASE CHAIN (DBC)

In this section, we present the main properties of the double-base number system, along with some numerical results in order to provide the reader with some intuitive ideas about this representation scheme and the difficulty of some underlying open problems. Double-base number systems have been suggested as a way to speed up scalar multiplication on elliptic curves. The double-base number system (DBNS) is a representation scheme in which every positive integer, n, is represented as the sum or difference of 2-integers, that is, numbers of the form $2^a 3^b$.

A double-base chain for k is an expansion of the form

$$k = \sum_{i=1}^{n} d_i 2^{a_i} 3^{b_i} \quad (8)$$

With where n is the length of the expansion $d_i \in \{-1, 1\}$ and such that the exponents (a_i, b_i) which is non negative integer $(a_i, b_i \geq 0)$ number decrease for the product order.

This Double-Base Chain (DBC) representation is highly sparse and, consequently, permits to reduce the Hamming weight of the expansion for the scalar. With the introduction of efficient tripling formulae, these representations using ternary bases greatly reduce the execution time of scalar multiplication.

For example, a double-base chain computing 4,012,174 is given as follows. In order to compute 4,012,174P, one uses a Horner-like algorithm by considering the differences between consecutive pairs of exponents and

Volume: 03 Issue: 05 Pages: 1329-1337 (2012)

by applying doubling and tripling and additions and subtractions accordingly. Ciet *et al* [1] have proposed a ternary / binary approach for fast ECC scalar multiplication. It makes use of the efficient doubling (2*P*), tripling (3*P*), and quadrupling (4*P*) of a point *P*.

ALGORITHM: Double-Base Scalar Multiplication

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

And such that $a_1 \geq a_2 \geq a_3 ... \geq a_m \geq 0$, $b_1 \geq b_2 \geq b_3 ... \geq b_m \geq 0$, and a point $P \in E(F_2 m)$.

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

$Z \leftarrow d_1 P$

for $i = 1,..., m-1$ do

$u \leftarrow a_i - a_{i+1}$

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

if $u = 0$ then

$Z \leftarrow 3(3^{v-1} Z) + d_{i+1} P$

Else

$Z \leftarrow 4^{\frac{(u-1)}{2}} Z$

If $u \equiv 0 \pmod{Z}$ then

$Z \leftarrow 4Z + d_{i+1} P$

else

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

return Z

Example:

$$4{,}012{,}174 = 2^{11}3^7 - 2^9 3^6 - 2^8 3^5 - 2^7 3^5 + 2^5 3^2 + 2^4 3^1 - 2^1 3^0$$

Later, [9] extended this approach, called Extended DB, to applications that can afford pre computations. In this case, d_i in (6) is allowed to have any value from a set of pre computed digits, where the elements are prime numbers other than 3. Finding short expansions using $\{2^{a_i} 3^{b_i}\}$ terms has been defined as a difficult problem on its own. [4] proposed to solve that problem by establishing "efficient" maximum bounds a-max and $b-$max for the powers of 2 and 3, respectively, and then executing an exhaustive search for closest terms $\{2^{a_i} 3^{b_i}\}$ (referred to as "Greedy" algorithm). In the next section multi-base (Triple base) number system is shown using the radix 7 to the previous approach and removes the shortcoming of the double base (DB).

Algorithm

GREEDY AlGORITHM

while $k > 0$

let z be the largest integer $2^a 3^b$

Output(a ,b)

replace k by k-z

$k - z \leftarrow 0$

else

end.

5. MULTIBASE NUMBER SYSTEM (MBNS)

As a generalization of double base chains, multi-base number system is very suitable for efficient computation of scalar multiplications of elliptic curves because of shorter representation length and less Hamming weight. In this paper, combined with the given formulas for computing the 5-fold of an elliptic curve point P, an efficient scalar multiplication algorithm of elliptic curve is proposed using 2, 3 and 7 as bases of the multi-based number system. The algorithms cost less compared with Shamir's trick and interleaving with NAF method.

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, 7}. 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}.

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

$$k = \sum_i d_i 2^{a_i} 3^{b_i} 7^{c_i} \quad (9)$$

Where $d_i \in \{-1,1\}$ and the terms of the form $2^{a_i} 3^{b_i} 7^{c_i}$ will be termed as 3-integers. A general multi-base representation although very short is not suitable for a scalar multiplication algorithm. So we include a special representation with restricted exponents

ALGORITHM: Multi-Base Scalar Multiplication

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

And such that

$a_1 \geq a_2 \geq a_3 \ldots \geq a_m \geq 0$,

$b_1 \geq b_2 \geq b_3 \ldots \geq b_m \geq 0$,

$c_1 \geq c_2 \geq c_3 \ldots \geq c_m \geq 0$ and, and a point $P \in E(F_2 m)$.

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

$Z \leftarrow d_1 P$

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

$u \leftarrow a_i - a_{i+1}$

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

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

if $u = 0$ then

$Z \leftarrow 7^w Z$

if $v \neq 0$ then

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

else

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

else

$Z \leftarrow 7^w Z$

$Z \leftarrow 3^v Z$

$Z \leftarrow 2^{u-1} Z$ //

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

DA is used

Return Z

Using this algorithm we have developed triple base representation of some numbers as examples.

$4{,}012{,}174 = 2^1 3^5 7^4 + 2^3 3^4 7^4 + 2^6 3^3 7^3 + 2^8 3^2 7^1 + 2^{10} 3^2 7^0 + 2^{12} 3^1 7^0$

$5{,}288{,}716 = 2^2 3^6 7^3 + 2^6 3^4 7^2 + 2^9 3^4 7^1 + 2^{10} 3^3 7^0 + 2^{11} 3^2 7^0 + 2^{14} 3^0 7^0 + 2^{16} 3^0 7^0$

$6{,}816{,}856 = 2^3 3^5 7^4 + 2^6 3^4 7^3 + 2^9 3^3 7^1 + 2^{12} 3^1 7^0 + 2^{18} 3^0 7^0$

It may be noted that there can be many representation of the same number but one has to be careful in shorting the sparest expansion. However, this algorithm provides a short and sparse triple base representation.

6. PROPOSED APPROACH

We propose a new method to speed up scalar multiplication, which is the most important operation in elliptic curve cryptography. Here we show that the advantage of this representation is that all point doublings and quadrupling can be replaced by faster point halving while maintaining the tripling operations. For binary fields, the approach requires only about half the number of the inversions, one-third of the number of squaring, and a fewer number of multiplications compared with the scalar multiplication using the original MBC(multi-base chain). We modify the mixed powers of 2, 3 and 7 proposed in [15] by representing the scalar by a new multi-base chain involving monotonically decreasing powers of 1/2, 3 and 7. In this paper, we propose a new multi-base chain representation with bases 1/2, 3 and 7 for the incorporation of point halving in scalar multiplication. With this method, we remove point doubling, quadrupling and use point halving instead, while maintaining the tripling and septupling points operations. Implementation of this method of scalar multiplication shows that our approach leads to a lower complexity in computing scalar multiplication. In the next paragraph we provide details of implementation of point halving.

Initially point halving was proposed independently by Knudsen [12] and Schroeppel [17]. It is the reverse operation of point doubling. If all the point doublings required in the traditional double-and-add method are replaced by the faster point halving operation, the computation speed could be faster up to 39% [12] and 50% [17]. A detailed analysis of the computational complexity of point halving was made in [10]. Incorporating these inputs we have proposed a new and much faster algorithm than the other classical approaches. In the following table the operating cost of different mathematical operations performed in an elliptic curve are shown, this we have calculated when

[i] the curve is described on a binary field and

[ii] when the curve is described on a prime field (F_q) of order q.

To denote cost of field operations, we will use [I], [S] and [M] to denote the cost of one inversion, one squaring and one multiplication respectively. We neglect the cost of field additions in comparison to cost of other operations which are much more than the cost of

Volume: 03 Issue: 05 Pages: 1329-1337 (2012)

additions also further we neglect the cost of squaring in case the curve is defined over a binary field.
For computing the scalar multiplications over binary finite fields, the required curve operations can be calculated as a_i doubling, b_i tripling and c_i septupling, using the same. The number of curve additions is the same as the number of terms in the chain. Whenever the components of the binary and ternary are not zero, double-and-add and triple-and-add operations are used instead of curve addition. In some cases the average number of terms in our algorithm is more than the number of terms in the original multi-base algorithm, but our method still costs less than the original algorithm.

Table1. Cost for different Operations

S. No	Operations	Binary field cost	Prime Field Costs
1	$P+Q$	$1I+1S+2M$	$1I+1S+2M$
2	2*P*	1*I* + 1*S* + 2*M*	1*I*+1*S* + 2*M*
3	2*P*+*Q*	1*I*+2*S* + 9*M*	1*I*+2*S* + 9*M*
4	3*P*	1*I*+4*S* + 7*M*	1*I*+4*S* + 7*M*
5	3*P*+*Q*	2*I*+3*S* + 9*M*	2*I*+3*S* + 9*M*
6	4*P*	1*I*+5*S*+8*M*	1*I*+5*S*+8*M*
7	5*P*	1*I*+5*S*+13*M*	10*S*+15*M*
8	7*P*	3*I*+7*S*+18*M*	-

The point halving operation is incorporated in to the new MB (multi-base) chain to achieve faster scalar multiplication. The paper shows that the advantage of this representation is that all point doublings required in the original chain point doubling and quadrupling can be replaced by faster point halving while maintaining all the tripling operations. For binary fields, the approach requires only about half the number of the inversions, one-third of the number of squaring, and a fewer number of multiplications compared with the scalar multiplication using the original DB chain and multi-base (MB) chain representation.

6.1 Point Halving Algorithm Implementation for MBNR

A careful analysis of elliptic curve point multiplication methods that use the point halving technique of Knudsen and Schroeppel, and have compared these methods to traditional algorithms that use point doubling and tripling. The performance advantage of halving methods is evident in the case of point multiplication kP .

ALGORITHM: Point Halving Scalar Multiplication

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

$d_i \in \{-1,1\}$

And such that $a_1 \geq a_2 \geq a_3 ... \geq a_m \geq 0$,

$b_1 \geq b_2 \geq b_3 ... \geq b_m \geq 0$,

$c_1 \geq c_2 \geq c_3 ... \geq c_m \geq 0$, and a point

$P \in E(F_2 m)$.

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

$Z \leftarrow d_1 P$

for $i = 1,...,m-1$ do

$u \leftarrow a_i - a_{i+1}$

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

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

if $u = 0$ then

$Z \leftarrow 7^w Z$

if $v \neq 0$ then

$Z \leftarrow 3(3^{v-1} Z) + d_{i+1} P$

// TA used

else

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

else

$Z \leftarrow 7^w Z$

$z \leftarrow 3^v Z$

$Z \leftarrow (\frac{1}{2})^{u-1} Z$

$Z \leftarrow (\frac{1}{2}) Z + d_{i+1} P$

return Z

In order to implement this we first multiply the scalar k with a large power of 2, say, 2^q, where 2^q is considered as a value around the field size. Next we calculate $2^q.k$ remainder $\bmod p$ (modulo) denoted as k' as given in equation (10).

$$k' = 2^q k \bmod p \quad (10)$$

Then we obtain the MB chain of k' with powers of 2 , 3 and 7 in the form of increasing binary exponents but decreasing ternary and septenary exponents. Some more steps (as explained in next paragraph) can yield the following form of representation of k as given by

$$k = \frac{k'}{2^q} = \frac{\sum_{i=1}^{m} d_i 2^{a_i} 3^{b_i} 7^{c_i}}{2^q} = \sum d_i (\tfrac{1}{2})^{(q-a_i)} 3^{b_i} 7^{c_i} \bmod p, \quad (11)$$

where $k' = 2^q k \bmod p$, $d_i \in \{1,-1\}$, $a_1 \le a_2 \le a_3 \ldots$, $b_1 \ge b_2 \ge b_3 \ge \ldots$ and $c_1 \ge c_2 \ge c_3 \ge \ldots$

This method will return the terms in the order from the highest power of 1/2 to the lowest power of 3 and 7. Here we reverse the terms i.e. the last terms becomes the first term, and then the expression become the desired one multi-base chain with the binary ternary exponents. In the following table we have taken different prime numbers. Some examples implementing this algorithm are given in the Table 2 . Two near by prime numbers k and p are chosen, then k is multiplied with a suitable power *q of 2 i.e.* 2^q and finally $k' = 2^q k \bmod p$ is calculated, finally the multi-base representation of k' is obtained using the already described algorithm. It is interesting to observe that in some cases the representation consist of only four terms.

Table.2

q	K	p	$k' = 2^q k$	Multi Base chain of k'
23	4,012,174	4,012,193	552,646	$2^1 3^6 7^3 + 2^3 3^4 7^2 + 2^6 3^3 7^1 + 2^9 3^2 7^0 + 2^{12} 3^0 7^0$
24	4,012,174	4,012,193	1,109,123	$2^0 3^5 7^4 + 2^4 3^4 7^3 + 2^8 3^3 7^1 + 2^{15} 3^0 7^0$
25	4,012,174	4,012,193	2,201,862	$2^1 3^5 7^4 + 2^5 3^4 7^3 + 2^9 3^3 7^1 + 2^{14} 3^1 7^0$
26	4,012,174	4,012,193	407,542	$2^1 3^3 7^4 + 2^7 3^3 7^2 + 2^{11} 3^3 7^1 + 2^{13} 3^1 7^0 + 2^{16} 3^0 7^0$

Finally we multiply the multi-base representation of k' by $\frac{1}{2^q}$, in order to make all the binary exponents negative but with decreasing magnitude. The ternary and septenary exponents are unaffected and are all positive or zero with decreasing magnitude. This is actually a new multi-base chain with decreasing powers of 1/2,3 and 7 with value equal to *k*. This representation for numbers in the column 3 of previous Table2 are given in Table3

Table3.

$k' = 2^q k$ mo	Point Halving Multi Base chain of k
552,646	$(\frac{1}{2})^{22} 3^6 7^3 + (\frac{1}{2})^{20} 3^4 7^2 + (\frac{1}{2})^{17} 3^3 7^1 + (\frac{1}{2})^{14} 3^2 7^0 + (\frac{1}{2})^{11} 3^0 7^0$
1,109,123	$(\frac{1}{2})^{24} 3^5 7^4 + (\frac{1}{2})^{20} 3^4 7^3 + (\frac{1}{2})^{16} 3^3 7^1 + (\frac{1}{2})^{9} 3^0 7^0$
2,201,862	$(\frac{1}{2})^{24} 3^5 7^4 + (\frac{1}{2})^{20} 3^4 7^3 + (\frac{1}{2})^{16} 3^3 7^1 + (\frac{1}{2})^{11} 3^1 7^0$
407,542	$(\frac{1}{2})^{25} 3^3 7^4 + (\frac{1}{2})^{19} 3^3 7^2 + (\frac{1}{2})^{15} 3^3 7^0 + (\frac{1}{2})^{12} 3^1 7^0 + (\frac{1}{2})^{10} 3^0 7^0$

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

$$K_1 = 1, \qquad K_i = 2^u 3^v 7^w K_{i-1} + d_i \text{ with}$$

$i \ge 2$, $d_i \in \{-1,1\}$

Now we use this recursive formula for implementing the point halving

$$K_i = (\tfrac{1}{2})^u 3^v 7^w K_{i-1} + d_i \text{ with } d_i \in \{-1,1\}$$

As an example for illustration of this algorithm we consider computing 2201862P. We first develop the multi base chain as given below.

$$2{,}201{,}862 = 2^1 3^5 7^4 + 2^5 3^4 7^3 + 2^9 3^3 7^1 + 2^{14} 3^1 7^0$$

Now we develop the halving chain for the same number with the help pf the algorithm. The point halving chain developed is given below.

$$2{,}201{,}862 = (\tfrac{1}{2})^{24} 3^5 7^4 + (\tfrac{1}{2})^{20} 3^4 7^3 + (\tfrac{1}{2})^{16} 3^3 7^1 + (\tfrac{1}{2})^{11} 3^1 7^0$$

After using the recursive formula we can obtain the following equation.

$$\frac{3}{2^{11}}\left[\frac{3^2 7^1}{2^5}\left[\frac{3^1 7^2}{2^4}\left[\frac{3^1 7^1}{2^4} + 1\right] + 1\right] + 1\right]$$

Method of calculating cost of calculating 2201862*P* in different iterations using point halving

Table 4

i	K	d	u	v	w
1	1	1	0	0	0
2	$\frac{21}{16}K_1+1$	1	4	1	1
3	$\frac{147}{16}K_2+1$	1	4	1	2
4	$\frac{63}{32}K_3+1$	1	5	2	1
5	$\frac{3}{2048}K_4+1$	1	11	1	0

With the help of the algorithm developed in this paper we can observe that the cost of the *k.P* is much less, if we compare with the previously developed methods. In this method the number of iterations decreases and is more efficient than the other methods proposed earlier. Implementation show that for binary fields, our approach requires only about half the number of inversions, one-third the number of squaring, and a slightly fewer number of multiplications when compared with the scalar multiplication using the multi-base chain representation. Finally, Table 4 summarizes that the cost of field operations optimize under this technique.

7. CONCLUSIONS

We have presented a scalar multiplication that uses the new point halving method to reduce the number of required terms in the scalar expansion. The scalar multiplication methods based on halving are straight forward to implement, although some extra static storage (per field) is required over methods based on doubling and tripling. The performance advantage of point halving methods is clearest in the case of point (scalar) multiplication *kP* , which is used in speeding up elliptic curve arithmetic. We have presented a scalar multiplication that uses the new multi-base chain method to reduce the number of required terms in the scalar expansion. This method significantly improves the MBNR algorithm and reduce the cost of field operations. The main procedure in the new algorithm used is the EC point halving and halve-and-add operations, which cost less, instead of using elliptic curve point doubling and double-and-add operations.

REFERENCES

[1]. Ciet,M., Joye,M., Lauter,K., and Montgomery, P.L.:(2003),"Trading Inversions for Multiplications in Elliptic Curve Cryptography", Cryptology ePrint Archive, Report 2003/257.

[2]. Darrel Hankerson, Julio Lopez Hernandez, and Alfred Menezes.(2000) "Software implementation of elliptic curve cryptography over binary fields". CHES 2000, 1965:1–24, August.

[3]. Dimitrov, V., Jullien, G., Miller, W.(1997) "Theory and Applications for a Double-Base Number System". ARITH 1997, pp. 44

[4]. Dimitrov, V., Imbert, L., Mishra, P.K.:(2005) "Efficient and Secure Elliptic Curve Point Multiplication using Double-Base Chains". ASIACRYPT 2005, LNCS, vol. 3788, pp. 59{78. Springer, Heidelberg.

[5]. Dimitrov, V., Mishra, P.K.:(2007) "Efficient Quintuple Formulas for Elliptic Curves and Efficient Scalar Multiplication using Multibase Number Representation". ISC 2007, LNCS, vol. 4779, pp. 390{406. Springer, Heidelberg.

[6]. Dimitrov, V., Imbert,L. and Mishra,P.K,.:(2005) "Efficient and secure elliptic curve point multiplication using double-base chains". In ASIACRYPT, pages 59–78, 2005.

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

[8]. Diffie,W., and Hellman, M. E.(1976) "New directions in cryptography", IEEE Trans.Inform. Theory, IT-22(6), November.

[9]. Doche,C., and Imbert,L.:(2006) "Extended Double-Base Number System with Applications to Elliptic Curve Cryptography", in Progress in Cryptology (INDOCRYPT'06*)*, LNCS Vol.4329, pp 335-348, Springer-Verlag

[10]. Fong,K., Hankerson,D., Lopez,J., and Menezes,A.:(2004) "Field Inversion and Point Halving Revisited", IEEE Transactions on Computers, vol. 53, no. 8, pp. 1047 – 1059.

[11]. Longa,P.:(2007) "Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields". Master Thesis, University of Ottawa . Available at http://patricklonga.bravehost.com/publications.html

[12]. Knudsen,E.W.:(1999) "Elliptic Scalar Multiplication using Point Halving", ASIACRYPT'99,LNCS 1716, pp. 135 – 149.

[13]. Koblitz,N.,(1987) "Elliptic curve cryptosystems", Math. Comp., 48(177): 203-209, January .

[14]. Miller,.V,(1986) "Use of elliptic curves in cryptography", in A. M. Odlyzko, editor, Advances in cryptology - CRYPTO 86, volume 263, of lecture notes in computer science. pages 417-426, Spriner-Verlag, 1987. Proceedings, Santa Barbr (USA), August 11-15.

[15]. Mishra,P.K., Dimitrov,V.S.:(2005) ”Efficient Quintuple Formulas for Elliptic Curves and Efficient Scalar Multiplication Using Multibase Number Representation”. Springer-Verlag, 2007, volume 4779, pages 390-406. Springer.

[16]. Morian,F., Olivos,J.:(1990) “Speedong up computation on an elliptic curve using addition subtraction chains”, Information theory applications, vol.24, pp 531-543.

[17]. Schroeppel,R.:(2000) “Elliptic Curve Point Ambiguity Resolution Apparatus and Method”, International Patent Application Number PCT/US00/31014, filed 9 November.

[18]. Wong,K.W., Edward Lee,C.W., Cheng,L.M, Xiaofeng Liao.:(2006) “Fast elliptic scalar multiplication using new double-base chain and point halving”, Applied Mathematics and Computation.