Speeding up Computation of Scalar Multiplication in Elliptic Curve Cryptosystem

H. K. Pathak
S.o.S in Computer science & IT
Pt. Ravishanker Shukla University
Raipur - 492010 (C.G.) India

Manju Sanghi
Department of Applied Mathematics
Rungta College of Engineering &Technology
Bhilai- 490024 (C.G.) India

Abstract. The basic operation in elliptic curve cryptosystem is scalar multiplication. It is the computation of integer multiple of a given point on the curve. Computation of scalar multiple is faster by using signed binary representation as compared to binary representation. In this paper 'Direct Recoding Method' a new modified algorithm for computation of signed binary representation is proposed. Our proposed method is efficient when compared to other standard methods such as NAF, MOF and complementary recoding method.

Keywords. Elliptic curve cryptography, Scalar multiplication, Signed binary method, NAF, MOF, Complementary recoding.

1. Introduction

Elliptic curve Cryptography was indepen dently introduced by Miller [1] and Koblitz [2] in 1985. Since then it has gained wide acceptance mainly due to its smaller key size and greater security. Scalar multip lication is the central operation of elliptic curve cryptosystem. It involves comput ation of kP where k is the secret key (scalar) and P a point on the elliptic curve. Significant methods to optimize ECC operations have been proposed. In this paper we introduce 'Direct Recoding' an efficient method to compute KP efficiently. For

$$2^{(p+1)} > k > 2^p,$$

$$k = (2^{p+1})_2 - (2^{p+1} - k)_2.$$

As this computation uses only bitwise subtraction it gives the signed binary representation with the lowest hamming weight. The rest of the paper is organized as follows. We start with the introduction of Binary method along with the computation of scalar multiplication in section 2. Signed binary NAF and MOF methods with their algorithms for computation of scalar multiplications are presented in Sections 3 and 4 respectively. In Section 5 we explain the Complementary recoding method based on NAF and finally in Section 6 we explain our proposed method with examples. Also the run time of various scalar multiplication algorithms are presented in this section.

2. Binary Method

Scalar point multiplication is the main cryptographic operation in ECC which computes Q = kP, a point P is multiplied by an integer k resulting in another point Q on the elliptic curve. Binary method [3] is the traditional scalar multiplication method based on the binary expansion of the scalar k using (0, 1). If k has binary representation $(k_{l-1}, k_{l-2}, \ldots\ldots, k_0)_2$ where $k_i \in (0, 1)$ then $k = \sum_{i=0}^{l-1} k_i 2^i$. Given an elliptic point P, $kP = \sum_{i=0}^{l-1} k_i 2^i P$

$= k_0 P + k_1 2^1 P + k_2 2^2 P + \ldots.. + k_{l-1} 2^{l-1} P.$

$= k_0 P + 2(k_1 P + 2(k_2 P + \ldots. + 2(k_{l-2} P + 2(k_{l-1} P) \ldots..).$

i.e., it uses repeated elliptic curve point addition and doubling operations. This method scans the bits of k either from left to right or right to left. Algorithm for the computation of KP is given below.

Algorithm 1. Left-to-right binary method for point multiplication.

Input: Binary representation of k and point P.

Output: Q = kP.

1. Q = ∞.
2. For i = l-1 to 0 do
   2.1 Q = 2Q (Doubling).
   2.2 If k_i = 1 then Q = Q + P (Addition).
3. Return Q.

The running time of an algorithm is determined as how many operations are performed throughout its

execution. If k_i is 1 then a point addition is performed and the expected number of ones (hamming weight) in the binary representation of k is half of its length i.e., l/2. Finally a doubling is performed for each value of I i.e., l times. Therefore the expected running time is l/2 additions + l doublings denoted as $\frac{l}{2}$ A + l D.

Example 1. Let k = 26 and P a point on the elliptic curve E. Given the binary expansion of k as

$$26 = 2^4 + 2^3 + 2^1 = (11010)_2.$$

The scalar multiplication denoted by 26P by using Algorithm 1 would be as follows:

$$26\,P = (\,2\,(\,2^{\,2}\,(\,2P + P) + P\,).$$

i.e., it requires 2 additions and 4 doublings.

3. Non Adjacent Form

The density of the binary expansion can be effectively reduced with a signed binary representation [7] that uses elements in the set (-1, 0, 1). Signed binary representation was first proposed by Booth [4] in 1951. Later Reitweisner [5] gave a constructive proof that every positive integer can be uniquely represented with fewest number of non-zero digits (minimum hamming weight) which is called *Non Adjacent Form* or NAF. In this form integer k is represented as

$k = \sum_{i=0}^{l-1} k_i 2^i$ where K_i Є (-1, 0, 1).

Algorithm 2. Computation of NAF of an integer k.

Input: A Positive integer k.

Output: NAF of k $(k_{l-1} \ldots\ldots k_2 k_1 k_0)_{NAF}$.

1. i = 0.
2. While k ≥ 0 do
   2.1 If k is odd $k_i = 2 - (k \bmod 4)$, $k = k - k_i$;
   2.2 Else $k_i = 0$.
   2.3 k = k/2, i = i+1.
3. Return $(k_{i-1} k_{i-2} \ldots k_1 k_0)$.

NAF method uses both addition and subtraction operations [6] but subtraction of points on elliktic curves is similar to addition operation. Hence running time of NAF is $\frac{l}{3}$ A + lD i.e., it reduces the hamming weight from $\frac{l}{2}$ to $\frac{l}{3}$.

Example 2. NAF of k = 687 is

$687 = 2^{10} - 2^8 - 2^6 - 2^4 - 2^0$

$= (10\text{-}10\text{-}10\text{-}1000\text{-}1)_{NAF}$

$= 1024 - 256 - 64 - 16 - 1,$

the hamming weight is 5 i.e. it uses 5 addition (subtraction equivalent to ad dition) operations while the binary representation of 687 is

$$2^9 + 2^7 + 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = (1010101111)_2.$$

the hamming weight is 7. By NAF hamming weight of k is reduced from 7 to 5 i.e. 2 addition operations have been saved.

4. Mutual Opposite Form (MOF)

MOF Mutual Opposite form is an efficient left to right recoding scheme proposed by Okeya [7] that satisfies the following properties:

1. The signs of adjacent non-zero bits (without considering 0 bits) are opposite.

2. The most nonzero bit and the least nonzero bit are 1 and -1 respectively.

Converting binary string to MOF:

The n-bit binary string k can be converted to a signed binary string by computing

mk = 2k - k where - stands for a bitwise subtraction.

$2k = k_{n-1}\ k_{n-2}\ \ldots.\ k_{i-1}\ldots..k_1\ k_0.$

$-k = k_{n-1}\ldots\ k_i\ \ldots...k_2\ k_1\ k_0.$

$mk = k_{n-1}\ k_{n-2} - k_{n-1}\ldots k_{i-1} - k_i\ldots.k_1 - k_2 k_0 - k_1 k_0$

Algorithm 3: Left to right generation from Binary to MOF.

Input: A non-zero n-bit binary string

$$k = k_{n-1}\ k_{n-2}\ldots..k_1 k_0.$$

***Output*:** MOF of k $(mk_n\ldots...mk_1\, mk_0)$.

1. $mk_n = k_{n-1}$
2. For i = n -1 to 0 do
   2.1 $mk_i = k_{i-1} - k_i$.
   2.1 $mk_0 = -k$.
3. Return $mk_n, mk_{n-1}, \ldots mk_1, mk_0$.

Example 3. Let k = 27, MOF of k is

$$2^5 - 2^3 + 2^2 - 2^0 = (10\text{-}110\text{-}1).$$

Like binary method MOF scans the bits either from left to right [8] or from right to left.

5. Complementary Recoding Technique

Given the binary representation of a scalar $k = (k_{l-1}\ldots k_1 k_0)_2$ the procedure for converting binary string into signed binary string using complementary method ([9], [10]) is given below:

$$K = \sum_{i=0}^{l-1} k_i 2^i = (1000...0)_{(l+1)\text{ bits}} - \bar{k} - 1$$

where $\bar{k} = \bar{k}_{i-1}\ \bar{k}_{i-2}\ \bar{k}_0$

and $\bar{k}_i = 0$ if $k_i = 1$

$\bar{k}_i = 1$ if $k_i = 0$ for i = 0, 1, ..., l-1.

Example 4. For $k = 687 = (1010101111)_2$, by the above method

$K = (100.....0)_{(10+1)\text{ bits}} - (0101010000) - 1$

$= (10\text{-}10\text{-}10\text{-}1000\text{-}1)$

i.e., it gives the same output as NAF but by using the complement of k.

6. Proposed method (Direct Recoding method)

According to our proposed method the procedure for converting the scalar k into signed binary representation is as follows:

For any scalar k where

$$2^{p+1} > k > 2^p, \text{ we have}$$

$$K = (2^{p+1})_2 - (2^{p+1} - k)_2.$$

Since this method uses only single operation of bitwise subtraction with $0 - 1 = \bar{1}$ it gives the signed binary representation with the lowest hamming weight and in the least possible time. Hence this method can be called as Direct recoding method.

The output of this method is also similar to other standard recoding methods such as NAF, MOF, and complementary recoding.

Algorithm 4: Scalar multiplication using Proposed method.

Input: Signed binary representation using proposed method.

Output: Q = kP.

1. Q = 0.
2. For i = n-1 to 0 do
   2.1 Q = 2Q.
   2.1 If $k_i = 1$, Q = Q + P;
   2.2 Else If $k_i = -1$, Q = Q – P.
3. End If.
4. Return Q.

Example 5. For k = 686

(i) By binary method, we have

$$686 = (1010101110)_2.$$

Clearly, the hamming weight of 686 is 6.

(ii) By NAF we find that

$$686 = (10\text{-}10\text{-}10\text{-}100\text{-}10).$$

In this case, the hamming weight of 686 is reduced from 6 to 5.

(iii) By complementary recoding we have

$$686 = (10000000000) - (0101010001) - 1$$

$$= (10\text{-}10\text{-}10\text{-}1000\text{-}1) - 1$$

The hamming weight is 6 (5 internal and 1 external).

(iv) By our proposed method, for

$$2^{10} > 686 > 2^9 \text{ we have}$$

$$686 = (2^{10})_2 - (2^{10} - 686)_2$$

$= (10000000000) - (101010010)$

$= (10\text{-}10\text{-}10\text{-}100\text{-}10).$

Thus the hamming weight of 686 is 5 but by using only single operation of bitwise subtraction.

Example 6. For k = 240

(i) By binary method we find that

$$240 = (11110000)_2.$$

Clearly, the hamming weight of 240 is 4.

(ii) By complementary recoding, we have

$$240 = (100000000) - ((00001111) - 1)$$

$$= (10000\text{-}1\text{-}1\text{-}1\text{-}1) - 1.$$

Here, the hamming weight of 240 is increased to 6 (5 internal and 1 external).

(iii) By our proposed method for

$$2^8 > 240 > 2^7 \text{ we have,}$$

$$240 = (2^8)_2 - (2^8 - 240)_2$$

= (100000000) - (10000) = (1000-10000).

Here, the hamming weight of 240 is reduced from 4 to 2 i.e., the least hamming weight when compared to all other existing methods.

We know that one addition operation requires 2 squaring, 2 multiplications and 1 inversion.

Hence our proposed method saves computational cost and time for performing 4 squaring, 4 multiplications and 2 inversions.

TABLE 1

COMPARISION OF RUN TIMES

The following table gives the comparison of run time of various signed binary represen tations NAF the *Non Adjacent form*, MOF *the Mutual Opposite form* , CRM the *Complementary Recoding method* and DRM *Direct Recoding*, the proposed method in seconds.

Bit size	Signed binary representations			
	NAF	MOF	CRM	DRM
25	15.80	13.78	11.16	9.0
37	19.42	17.56	15.09	12.46
44	21.76	19.26	17.29	15.51
52	23.28	20.71	19.36	17.51

We implemented our algorithm on Intel p4 dual core processor 1.6 GHz and 782 MHz and 504 MB of memory using Matlab.

From the table we find that our proposed method takes the least time to find the signed binary representation of any integer k when compared to the other known methods (See, for instance, Fig.1 below).

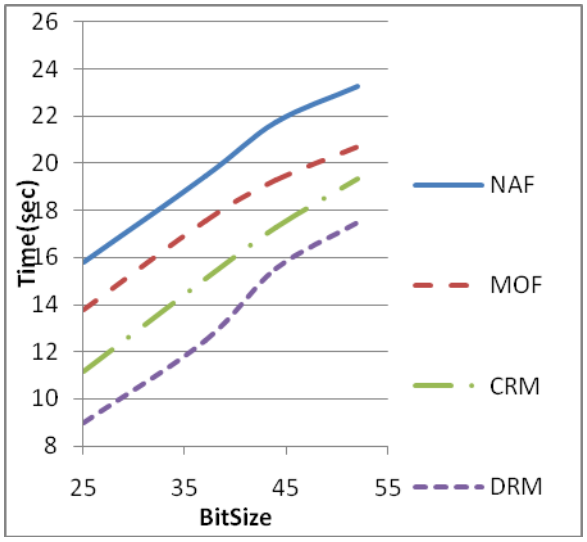

Fig 1 Time ratio of the direct recoding method with other algorithms (NAF, MOF, CRM, DRM).

7. Conclusion

In the implementation of ECC scalar multiplication is not only the basic computation but also the most time consuming operation. Its Operational efficiency directly determines the performance of ECC.

In this paper we proposed a scalar multiplication using direct recoding method. Theoretical tasks and numerical tests reveal that this algorithm can remarkably enhance the computing efficiency of scalar multiplication compared with other traditional algorithms and therefore has practical significance for the implementation of ECC. Moreover, Fig. 1 shown above earnestly justifies our conclusion.

8. References

[1] V. S. Miller, Use of elliptic curves in cryptography, Advances in Cryptology, Proceedings of CRYPTO'85, LNCS, 218 (1986), 417-426.
[2] N. Koblitz, Elliptic curve cryptosystem, Mathematics of Computation, 48 (1987) 203-209.
[3] Standard specifications for Public key cryptography, IEEE Standard 1363, 2000.
[4] A.D. Booth, A signed binary multiplication technique, Journal of Applied Mathematics, 4(2) (1951), 236-240.
[5] G. W. Reitwiesner, Binary Arithmetic, Advances in computers, 1 (1960), 231-308.
[6] F. Morain, J.Olivos, Speeding up the computations on an elliptic curve using addition subtraction chains, RAIRO Theoretical Informatics and Applications, 24 (1990), 531-543.
[7] K. Okeya, Signed binary representations revisited, Proceedings of CRYPTO'04 (2004), 123-139.
[8] M. Joye, S. Yen, Optimal left to right binary signed digit recoding, IEEE Transactions on Computers, 49 (2000), 740-748.
[9] P. Balasubramaniam, E. Karthikeyan, Elliptic curve scalar multiplication algorithm using complementary

recoding, Applied Mathematics and Computation, 190 (2007), 51-56.

[10] P.Balasubramaniam, E. Karthikeyan, Fast Simultaneous scalar multiplication, Applied Mathematics and Computation, 192 (2007), 399-404.

Authors Profile

1. Dr. H. K. Pathak received Post Graduate degree in Mathematics from Pt. Ravishanker Shukla University, Raipur. He was awarded Ph.D in 1988 by the same University. He has published more than 185 research papers in various international journals in the field of non linear analysis-Approximation and expansion, Calculus of variations and optimal controls Optimization, Field theory and polynomials, Fourier analysis, General topology, Integral equations, Number theory, Operations research, Mathematical programming, Operator theory, Sequences, Series, summability. At present he is Professor and Head in S.o.S in Computer science & IT in Pt. Ravishanker Shukla University.

2. Mrs. Manju Sanghi received the post graduate degree in Mathematics from Ravishanker University Raipur in 1996. Since 2001 she has been working as lecturer in Rungta college of Engineering & Technology Bhilai. Currently she is pursuing PhD from School of studies in Mathematics Ravishanker Shukla University Raipur. Her research interests include Cryptography.