Proceedings of the 2001 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 5–6 June 2001

A Fast VLSI Algorithm for Multiplication on Elliptic Curves

Chang N. Zhang, Hua Li

Abstract— **Large integer modular multiplication are arithmetic primitives required by the public-key cryptosystems including Elliptic Curve Cryptography (ECC). In this paper a fast new algorithm for multiplication of a point on the elliptic curve with a large integer based on non-adjacent form(NAF) Frobenius expansion is proposed. It speeds up 50% and is much simpler compared with previous works. We also show that the proposal algorithm is very suitable for high speed VLSI implementation.**

Keywords— **Elliptic curves, multiplication, Frobenius expansion, Frobenius endomorphism, encryption.**

I. Introduction

THE important issue of information security has driven research into new cryptographic algorithms and their efficient implementations. Public-key cryptography has several advantages compared with non public-key cryptography. In particular, it eliminates the problem of the key distribution and provides some important function such as digital signature. To ensure a high degree of security, most proposed public key cryptographic algorithms require large keys and intensive computations such as modular multiplications and modular exponentiations. In recent years, elliptic curves over finite fields have been focused in public key cryptography [1], [2]. Schroeppel and his colleagues implemented a Diffie-Hellman type cryptosystems by elliptic curves over $F_{2^{155}}$ in [3]. In the elliptic curve cryptosystems, the most time consuming operation is the multiplication of a point with an integer which is approximately equal to the order of the underlying group.

In this paper we explore the ideas of multiplication elliptic curve points by Frobenius expansion presented in [4] and derive a new non-adjacent form(NAF) Frobenius expansion algorithm for high speed implementation of multiplication of the point on the elliptic curve which speeds up 50% compared to usual algorithms. Moreover the proposed algorithm is much simpler than Solinas' NAF algorithm [5].

The outline of the remainder of the paper is as follows. In Section 2, we introduce the Frobenius endomorphism of the elliptic curve and explore multiplication points by Frobenius Expansion. Section 3 contains the derivations of a new algorithm for multiplication of points by Frobenius expansion. Section 4 concludes with the improved result and a description of areas of applications.

Chang N. Zhang and Hua Li are with Department of Computer Science, TRLabs, University of Regina,Canada S4S 0A2. Email: zhang@cs.uregina.ca, huali@cs.uregina.ca .

II. Frobenius Endomorphism of the Elliptic Curve and Multiplication Points by Frobenius Expansion

A nonsupersingular elliptic curve E over F_q can be defined by

$$y^2 + xy = x^3 + a_2x^2 + a_6. \tag{1}$$

where $a_2, a_6 \in F_q$ and $a_6 \neq 0$. There exist simple algebraic formulas for adding two arbitrary points in the extension field F_{q^n} and $E(F_{q^n})$ forms an additive abelian group with zero element denoted by $\mathcal{O}$.

The qth-power Frobenius endomorphism of E is given as

$$\Phi : E \to E(x, y) \to (x^q, y^q). \tag{2}$$

The map Φ satisfies the equation

$$\Phi^2 - c\Phi + q = 0. \tag{3}$$

where $|c| \leq 2\sqrt{q}$ and trace c of Φ must be odd.

Müller [4] gave a detailed explanation on how to compute the multiplication of a elliptic curve point(P) by m, i.e., mP, on $E(F_{q^n})$ in terms of a polynomial in Φ. Following is an important referenced theorem [4] which serves as the base for the Algorithm 1.

Theorem 1. Let $s \in Z[\Phi]$, If we set $k = \lceil 2log_q\|s\|\rceil$, then there exist rational integers $r_i \in \{-q/2, ..., q/2\}$, $0 \leq j \leq k$, such that

$$s = \sum_{i=0}^{k} r_i\Phi^i.$$

The multiplication of point by m map on the elliptic curve can by implemented by Frobenius expansions. From Theorem 1, we can expand the integer m as a power sum

ISBN 0-7803-9814-9/$10.00

of the Frobenius endomorphism Φ:

$$m = \sum_{i=0}^{k} m_i \Phi^i.$$

where $k \leq \lceil 2log_q m \rceil$. Then mP for $P \in E(F_{q^n})$ can be computed by

$$\begin{aligned} mP = & \sum_{i=0}^{k} m_i \Phi^i(P) \\ = & \Phi(...\Phi(m_k \Phi(P) + m_{k-1}P) + ... + m_1 P) + m_0 P \end{aligned} \quad (4)$$

Algorithm 1 (EC point multiplication by Frobenius expansion)

Precompute table of points

(1) computer and store iP for all $1 \leq i \leq q/2$;

Compute Frobenius expansion

(2) $s_1 = m, s_2 = 0$ and $i = 0$;

(3) while ($|s_1| > q$ *or* $|s_2| > q/2$) do

(4) compute and store $m_i = s_1 \bmod q$;

(5) $h = (m_i - s_1)/q, i = i+1, s_1 = s_2 - c\ h$ and $s_2 = h$;

(6) end of while

"Left-to-right" multiplication

(7) $H = s_2\ \Phi(P) + s_1\ P$;

(8) for ($j = i - 1$; $j \geq 0$; $j - -$)

(9) if ($m_j \geq 0$)

(10) $H = \Phi(H) + m_j\ P$;

(11) else

(12) $H = \Phi(H) - |m_j|\ P$;

(13) end of for loop;

(14) return H.

The advantage of this algorithm is that the number of quadratic field operations in F_{q^n} is much smaller that in the binary algorithm with the Frobenius expansion. The time consuming elliptic curve doublings and additions can be replaced by fewer additions and some power evaluations in a finite field.

III. An Improved Multiplication Algorithm by Using NAF

Solinas proposed an algorithm that can reduce the length of the Frobenius expansion by nearly 50% [5]. The essential idea in Solinas's algorithm is to use the non-adjacent form (NAF) of the binary expansion for the coefficient which has the property that no two consecutive coefficients are non-zero. But Solinas's algorithm is much complicated compared with Müller's algorithm , as it requires more multiplication and division operations than M]"uller's algorithm. In the following, We show that the computations required by Algorithm 1 can be significantly reduced by applying the non-adjacent form (NAF) to the binary expansion. The NAF coefficient can be obtained when an endomorphism Φ of norm 2 is used. Assume that $m_0, ..., m_k$ in $\{0, 1, -1\}$ are the coefficients after Frobenius expansion. We can change these coefficients by the characteristic polynomial of Φ to "kill" one of successive non-zero coefficients. For example, assume $\Phi^2 - \Phi + 2 = 0$ ($c = 1, q = 2$) and $m_0 = 1$, there are three cases for m_1:

1. $m_1 = 0$, then everything is fine and we check m_2.

2. $m_1 = 1$, that is $m = 1 + \Phi + m_2 * \Phi^2 + ...$

Note that $2 = \Phi - \Phi^2$, we can get $1 + 1 = \Phi - \Phi^2$, thus $1 = \Phi - \Phi^2 - 1$, then we get

$$\begin{aligned} m = & (\Phi - \Phi^2 - 1) + \Phi + m_2 * \Phi^2 + ... \\ = & -1 + 2 * \Phi + (m_2 - 1) * \Phi^2 + ... \end{aligned} \quad (5)$$

As $2 * \Phi = \Phi^2 - \Phi^3$, we get

$$\begin{aligned} m = & -1 + (\Phi^2 - \Phi^3) + (m_2 - 1) * \Phi^2 + m_3 * \Phi^3 + ... \\ = & -1 + 0 * \Phi + m_2 * \Phi^2 + (m_3 - 1) * \Phi^3 + ... \end{aligned} \quad (6)$$

3. $m_1 = -1$, that is $m = 1 - 1 * \Phi + m_2 * \Phi^2 + ...$

In this case, we substitute 1 in the above equation with $1 = \Phi - \Phi^2 - 1$ and get

$$\begin{aligned} m = & (\Phi - \Phi^2 - 1) - 1 * \Phi + m_2 * \Phi^2 + .. \\ = & -1 + 0 * \Phi + (m_2 - 1)\Phi^2 + ... \end{aligned} \quad (7)$$

After these transformations, the two first coefficients m_0 and m_1 satisfy the definition of the NAF. We can repeat this process until all coefficients becoming NAF. Note that the length of the expansion grows by 2 bits more than original. The new coefficients can be reduced to the correct range by the characteristic polynomial. Note that $2 = \Phi - \Phi^2$ and so $2 * \Phi = \Phi^2 - \Phi^3$, we can "shift" the characteristic polynomial, i.e., if all the coefficients $m_0, ..., m_{i-1}$ are already NAF, but m_i is not, then we can correct m_i with $\Phi^{i-1} * 2 = \Phi^{i-1} * (\Phi - \Phi^2)$.

For coefficient $m_0 = -1$, everything is symmetric, just multiply all the equations with -1. Since $2 = \Phi - \Phi^2$, for every constant c in $Z[\Phi]$ we have $c * 2 = c * (\Phi - \Phi^2)$.

Similar reducing results can be obtained for $c = -1$. In implementation, we may use a look-up table to get the NAF Frobenius expansion. Table I lists the corresponding value or correction value for the coefficients of NAF Frobenius expansion and the Algorithm 2 describes how to change the coefficients of Frobenius expansion to NAF.

In order to efficiently use the look-up table, we can change the sequence of the items in the table and using the following equation to get the address entry of the look-up table.

$$\text{Address} = c * 2^0 + m_{i-1} * 2^1 + m_i * 2^2 + 7 \quad (8)$$

Algorithm 2: Change the Frobenius expansion coefficients to NAF

c	m_{i-1}	m_i	a	b	d
1	1	1	-1	0	-1
1	1	-1	-1	-1	0
1	-1	1	1	1	0
1	-1	-1	1	0	1
-1	1	1	-1	-1	0
-1	1	-1	-1	0	1
-1	-1	1	1	0	-1
-1	-1	-1	1	1	0

TABLE I
LOOK-UP TABLE FOR NAF FROBENIUS EXPANSION

Input: $c, m_0, m_1, ..., m_k$
Output: NAF of $m_0, m_1, ..., m_k, m_{k+1}, m_{k+2}$
Begin
for $(i = 1; i \leq k; i++)$
begin
  if $(|m_{i-1}| == 2)$
    begin
      $m_{i-1} = 0$;
      $m_i = m_i + c$;
      $m_{i+1} = m_{i+1} - 1$;
    end
  else
    begin
      using the look-up table to get the values of a, b, d
        by Address=$c * 2^0 + m_{i-1} * 2^1 + m_i * 2^2 + 7$;
      $m_{i-1} = a$;
      $m_i = 0$;
      $m_{i+1} = m_{i+1} + b$;
      $m_{i+2} = m_{i+2} + d$;
    end
end of for-loop;
End.

We can combine the Algorithm 1 and Algorithm 2 to compute mP as shown in Algorithm 3. We choose q=2 for fast implementation and suppose $k = \lceil 2log_2 m \rceil$.

Algorithm 3: A fast EC point multiplication algorithm

Input: c, m, P
Output: mP
Begin
(Compute NAF Frobenius expansion)
$s_1 = m, s_2 = 0$;
$m_0 = s_1 \bmod 2$;
$h = (m_0 - s_1)/2,\ s_1 = s_2 - c\ h,\ s_2 = h$;
for $(i = 1; i \leq k; i++)$
begin
$m_i = (s_1 \bmod 2)$;
$h = (m_i - s_1)/2,\ s_1 = s_2 - c\ h,\ s_2 = h$;
if $(|m_{i-1}| == 2)$
  begin
    $m_{i-1} = 0$;
    $m_i = m_i + c$;
    $m_{i+1} = m_{i+1} - 1$;
  end
else
  begin
    Using the look-up table to get the values of a, b, d;
    $m_{i-1} = a$;
    $m_i = 0$;
    $m_{i+1} = m_{i+1} + b$;
    $m_{i+2} = m_{i+2} + d$;
  end
end of for-loop;
("Left-to-right" multiplication)
$H = s_2\ \Phi(P) + s_1\ P$;
for $(j = k + 2;\ j \geq 0;\ j--)$
begin
  if $(m_j \geq 0)$
    $H = \Phi(H) + m_j P$;
  else
    $H = \Phi(H) - |m_j| P$;
end
return H;
End.

From Algorithm 3, we can see that the binary expansion has been replaced by NAF number, i.e., there are no two consecutive coefficients which are non-zero, thus the number of point multiplication can be reduced dramatically. Compared to Algorithm 1,the number of point multiplication is reduced nearly 50%. Thus the new algorithm can speed up two times compared with Algorithm 1.

Moreover, as all the computations in algorithm 3 are very simple, the algorithm 3 can be efficiently implemented by the hardware. Figure 1 and Figure 2 depict the logic structure for computing NAF Frobenius expansion and the point multiplication on the elliptic curve.

IV. CONCLUSION

Asymmetric cryptographic algorithms, such as Elliptic Curve Cryptography, require large number of modular multiplications and/or exponentiations. In this paper, we propose a new algorithm for point multiplication on ECC based on the Müller's Frobenius expansion and the NAF representation. This new algorithm can speed up the Müller's algorithm by approximately 50%. In addition, a hardware design for multiplication on ECC is derived. It make the Elliptic Curve cryptosystem very suitable for VLSI implementation and the low-cost implementations are also feasible in the restricted computing environments.

ISBN 0-7803-9814-9/$10.00

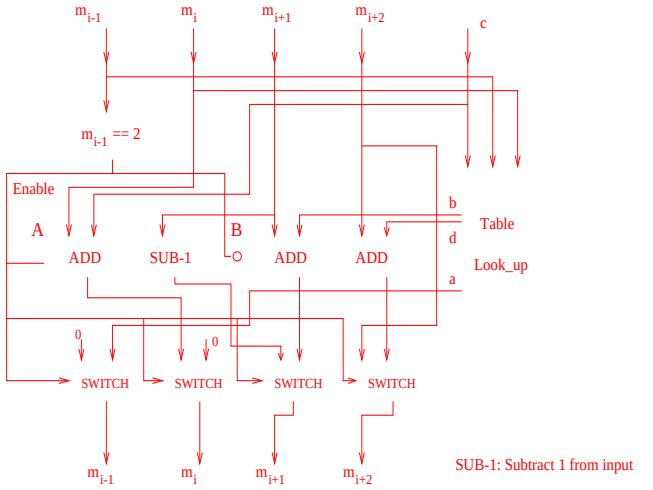

Fig. 1. The logic structure for correction of m_i

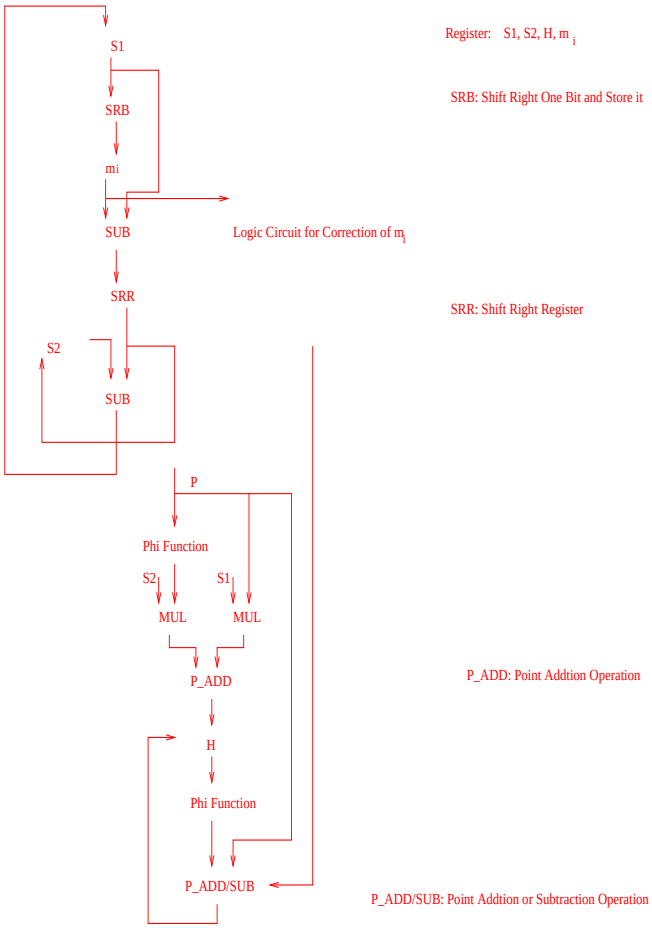

Fig. 2. The logic structure for computing NAF Frobenius expansion and the point multiplication on elliptic curve

Acknowledgments

Thanks Dr. Volker Müller to give us suggestions for this paper.

References

[1] N. Kobliz, "Elliptic Curve Cryptosystems," *Math. Comp*, Vol. 48, pp. 203-209, 1987.

[2] V.S. Miller, "Use of Elliptic Curves in Cryptography," *Advances in Cryptology-CRYPTO 85*, Lecture Notes in Computer Science, No. 218, Springer-Verlag, Berlin, pp. 417-426, 1986.

[3] R. Schroeppel, H. Orman, S. O'Malley, O. Spatschek, "Fast Key Exchange with Elliptic Curve Systems," *Advances in Cryptology-CRYPTO 95*, Lecture Notes in Computer Science, NO. 963, Springer-Verlag, Berlin, pp. 43-56, 1995.

[4] Vloker Müller, "Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two," *J. Cryptology*, Vol. 11, pp. 219-234, 1998.

[5] J. A. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves," *Advances in Cryptology*, CRYPTO 97, pp. 357 - 371, Springer-Verlag, Berlin, 1997.

[6] N. P. Smart, "Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic," *J. Cryptology*, Vol. 12, pp. 141-151, 1999.

[7] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[8] G.B. Agnew, R.C. Mullin and S.A. Vanstone, "An implementation of elliptic curve cryptosystems over $F_{2^{155}}$," *IEEE Journal on Selected Areas in Communications*, Vol.11, No.5, pp. 804-813, June, 1993.

[9] G. Harper, A. Menezes and S. Vanstone, "Public-key cryptosystems with very small key lengths," *Advances in Cryptology*, Proc. Eurocrypt'92, LNCS 658, pp. 163-173, Springer-Verlag, 1993.