A Multiplier to Enhance the Speed of Encryption/Decryption

YANG Ai-Min,ZHANG Wen-Xiang
Doctor of Wuhan University of Technology;

Computer Science and Information Technology College
Zhejiang Wanli University
No.8, South Qian Hu Road Ningbo, Zhejiang Province
China

Abstract: - It has become increasingly common to implement discrete-algorithm based public-key protocols on elliptic curves over finite fields. The operations, especially multiplication, over finite fields affect greatly the speed of encryption/decryption. For this reason the contribution describes a fast multiplier for Elliptic Curve Cryptosystems over finite fields $GF((2^{m1})^{m2})$. This multiplier adopts mixed parallel-serial approaches., The number of clock cycles for one field multiplication can be reduced from former $m= m_2m_1$ to current $m2$ with less increase of hardware scales.

Key-Words: - multiplication, LFSR, discrete-algorithm , finite fields ,composite fields ,Massey-Omura

1 Introduction

The finite field $GF(2^m)$ is a digital system with 2^m elements. Each element can be expressed in the form of m binary numbers when practically applied in the field $GF(2^m)$. Besides its application in the encoding theory, the computation of the finite field $GF(2^m)$ has also been applied in the communication security of the digital information , such as the enciphering and deciphering computation. The most important computation in the finite field is multiplication. There are three presentations of computation data: Dual, Normal and Standard (or Polynomial). For the Dual multiplication, the multiplicand is expressed in Dual , while the multiplier is expressed in Standard; and yet the result is expressed in Dual[1]. In assey-Omura multiplication, both the multiplier and the multiplicand are presented with Normal[2，3]. In Scott-Tavares-Peppard multiplication, however, all the elements in the finite filed are presented with the Standard[4]. Among the multipliers presented by the three data, the Dual multiplication requires the fewest gate counts, and thus, it occupies the smallest areas in the VLSI design. Massey-Omura multiplication is very effective in the inversion, square and index computations where the operation process only involves a shift. It directly maps the binary input and output of the data. Nevertheless, both Dual and Massey-Omura multiplications involve the data transformation. The circuit structure is irregular. Moreover, when the dimension m increases in the finite field, the circuit size of Massey-Omura multiplier will augment rapidly, and thus it is not suitable to the actualization of VLSI. Comparatively, there is no requirement for the transformation of data presentation in the Standard multiplication, its circuit being regular, simple and fast. It is much easier for the standard multiplication to be actualized when the dimension m increases in the finite field. Besides, the irreducible unary primitive polynomial in the field can be changeable, while in Dual and Massey-Omura multiplications. it must be selected.

Of the public key systems, the elliptic curve cryptosystem has been regarded as an ideal type. What makes this system superior lies in its use of shorter key on precondition of security. It is generally believed that, for the elliptic curve cryptosystem of q meta-field, when the length of q is 160bit, its security equals that of RSA applying 1024bit modulo. If the selected finite field is $GF(2^m)$, only when $m \geqslant 160$, can the elliptic curve cryptosystem be regarded secure. Therefore, to make the elliptic curve cryptosystem in the finite field $GF(2^m)$ possible, the Standard base presentation should be used in design, with a consideration of the computation characteristics of the elliptic curve. And the finite computation should mainly be multiplication with parallel and serial ways combined

2 Multiplication based on the finite field

2.1 Traditional finite field multiplication

Here the computation of module multiplication by using matrix should be introduced. Suppose the two elements A, B in *GF（2^n）* are as follows:

$A(x)=a_{n-1}x^{n-1}+\ldots+a_1x+a_0 \quad a_i \in$ GF（2^n）

$B(x)=b_{n-1}x^{n-1}+\ldots+b_1x+b_0 \quad b_i \in$GF（$2^n$）

$Q(x)=x^n+q_{n-1}x^{n-1}\ldots+q_1x+q_0$, $q_{0=1}$, $q_i \in GF(2)$ is the irreducible unary primitive polynomial of the finite field *GF（2^n）*.

If $C(x)=A(x)\times B(x)$ mod Q(x), then

$C_{n-1}x^{n-1}+\ldots+c_0=(a_{n-1}x^{n-1}+\ldots+a_0)\,(b_{n-1}x^{n-1}+\ldots+b_0)$ mod Q(x) (1)

To compute the value of $x^n, x^{n-1}, \ldots, x^{2n-2}$ mod Q(x), then

$$\begin{pmatrix} x^n \\ x^{n+1} \\ \vdots \\ \\ x^{2n-2} \end{pmatrix} = \begin{pmatrix} q_{0,0} & \cdots & q_{0,n-1} \\ & & \\ \vdots & & \\ \vdots & & \\ q_{n-2,0} & \cdots & q_{n-2,n-1} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ \vdots \\ \\ x^{n-1} \end{pmatrix} \bmod Q(x) \tag{2}$$

Thus the value of $q_{i,j}$ satisfies

$q_{0,j}=q_j \quad j=0,1,\ldots,n-1.$

$$q_{i,j} = \begin{cases} q_{i-1,n-1} & i=1,\ldots,n-2;\ j=0. \\ q_{i-1,j-1}+q_{i-1,n-1}q_{0,j} & i=1,\ldots,n-2;\ j=1,\ldots,n-1 \end{cases} \tag{3}$$

Therefore, the computation of Formula (1) can be demonstrated as:

$$C = \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ \\ C_{n-1} \end{pmatrix} = FB = \begin{pmatrix} f_{0,0} & \cdots & f_{0,n-1} \\ & & \\ \vdots & & \\ \vdots & & \\ f_{n-1,0} & \cdots & f_{n-1,n-1} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \\ b_{n-1} \end{pmatrix} \tag{4}$$

Of which,

$$f_{i,j} = \begin{cases} a_i & i=0,\ldots,n-1;\ j=0 \\ u(i-j)a_{i-j}+\sum_{t=0}^{i-1} q_{i-1-t,j}\,a_{n-1-t} & i=0,1,\ldots,n-1;\ j=1,\ldots,n-1 \end{cases}$$

$$u(i-j) = \begin{cases} 0 & i<j \\ 1 & i \geq j \end{cases} \tag{5}$$

Computation of all the formulae above are done in the field *GF (2)*, which requires n^2 times of module multiplications.

2.2 Improved finite field multiplication

Given the large number of module multiplication in the finite field, the Karatsuba-Ofman computation（KOA）proposed by Karatsuba and Ofman can reduce the workload by $1/4n^2$ times. The core of KOA is the decomposition of polynomial and adoption of the recursive algorithm in the module computation of the finite field. As *A(x)*、*B(x)* is the polynomial of the high-order *n*-1of the field *GF（2^n）*, so the high-order of $D(x)= A(x)B(x)$, *D(x)* is 2*n*-2. In this case, *n* should be a prime number bigger than 2(obviously *n* is an odd number), let *r*=*n*+1, according to KOA, its decomposition should be as follows:

$$A(x)=x^{r/2}(0x^{r/2-1}+a_{r-2}x^{r/2-2}+\ldots+a_{r/2})+(a_{r/2-1}x^{r/2-1}+\ldots+a_0)=x^{r/2}A_h+A_l$$

$$B(x)=x^{r/2}(0x^{r/2-1}+b_{r-2}x^{r/2-2}+\ldots+b_{r/2})+(b_{r/2-1}x^{r/2-1}+\ldots+b_0)=x^{r/2}b_h+b_l \tag{6}$$

According to Formula (6), the computation

$$\left.\begin{aligned} D_0(x)&= A_l(x)\,B_l\,(x) \\ D_1(x)&=[\,A_l(x)+ A_h(x)]\,[\,B_l(x)+ B_h(x)] \\ D_2(x)&= A_h(x)\,B_h\,(x) \end{aligned}\right\} \tag{7}$$

As Fig.1 demonstrates,

$D(x)= A(x)B(x)$

$$D(x) = A_l(x)B_l(x)$$

$$D(x) = D_0(x) + x^{r/2}[D_1(x) - D_0(x) - D_2(x)] + x^r D_2(x) \tag{8}$$

Through Formula (7) and (8), the number of times of the coefficient multiplication can be reduced from the original n^2 to $3/4n^2$ [5].

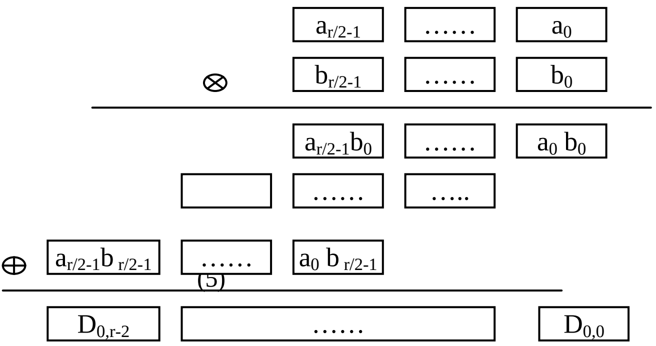

Fig.1 $D_0(x)= A_l(x)\,B_l\,(x)$

The multiplication and addition computations in diagram1(the modulo 2 multiplication and modulo 2 addition) are all operated in the field *GF（2）*. Though Fig.1 shows the computation of $D_0(x)= A_l(x)\,B_l\,(x)$, it can also be applied in the computation of $D_1(x)$ and $D_2(x)$. The only difference is when computing $D_1(x)$, an extra computation of $A_l(x) + A_h(x)$ and $B_l(x)+ B_h(x)$ is needed. This requires at most r/2-1 XOR gates.

When it comes to the computation like the form of $xsW(x)$ mod $Q(x)$, with the application of $x^n= q_{n-1}x^{n-1}\ldots+ q_1x+ q_0$ mod $Q(x)$, it can be realized through the looping of the LFSR s times as demonstrated in Fig.2

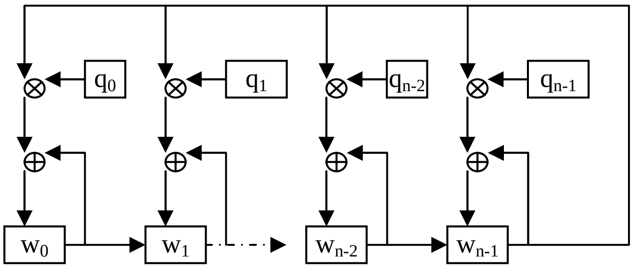

Fig.2 LFSR

If $W(x)= D_1(x) - D_0(x) - D_2(x)$, $s=r/2$ or $W(x)= D_2(x)$ $s=r$ is taken out of $x^sW(x)$ mod $Q(x)$, the computation of $C(x)= A(x)\times B(x)$ mod Q can be accomplished through the LFSR and the computation module in Fig.1.

3 Multiplication structure in the composite field

3.1 Finite compound field

Suppose $\overline{F}$ is a finite field, and F is a subfield of $\overline{F}$, then m is the extended dimension of field $\overline{F}$ based on the subfield F, $E_m=\{e_0,e_1,\ldots,e_{m-1}\}$ is a group base of field $\overline{F}$ on the basis of F. Of it, $e_0,e_1,\ldots,e_{m-1} \in \overline{F}$ is linearly independent. Once given the base Em of field $\overline{F}$ on the basis of the subfield F, to any $A \in \overline{F}$, it can be indicated by only one m dimension row vector, that is, $A=(a_0,a_1,\ldots,a_{m-1})$, of which $a_0,a_1,\ldots,a_{m-1} \in F$. Suppose p and p^l are the eigenvalue and base of the field F respectively（P is a prime number）.

Theorem 1[6]: suppose $\overline{F}$ contains P^n elements, then field $\overline{F}$ contains a subfield F with P^t elements and only when n can be divided by t with no remainder.

Suppose F_0, F_1, …,F_s are finite fields, and F_{i-1} is the subfield of F_i, i=1,2,…,s, m_i is the extended dimension of F_{i-1}based on F_i, thus F_0, F_1, …,Fs forms a group of chain fields in descending sort. Their relationship can be expressed as the following:

$$F_0 \underset{m1}{>} F_1 \underset{m2}{>} \ldots \underset{ms-1}{>} F_{S-1} \underset{ms}{>} F_S \qquad (9)$$

Of them, $F_{i-1} \underset{mi}{>} F_i$ denotes F_{i-1} is the compound field of F_i with a dimension of m_i.

Inference1: Suppose $\overline{F}$ is a finite field, F is a subfield of $\overline{F}$, m is the extended dimension of $\overline{F}$ based on the subfield of F, if $m=m_2m_1$, m_2,m_1 are prime numbers, and m_i>1isa positive integer, then there is a group of chain fields in descending sort between $\overline{F}$ and F.

Theorem 2[7]: suppose

$p^{i-1}(x) = x^{m_i} + \sum_{j=0}^{m_i-1} p_j^i x^j$ is an irreducible unary polynomial of m_i order in the field F_i,

$$P_j^i \in F_i \qquad j = 0,1,\ldots,m_i, i = 1,2,\ldots,s$$

Then $F_{i-1}$and F_i/ (p^{i-1}(x)) are in isomorphism.

3.2 Multiplication structure of the finite composite field

If the eigenvalue P of $\overline{F}$ 、F is 2，that is, $\overline{F}$ is $GF(2^n)$, F is $GF(2)$, in consideration of the computation, let $m=m_2m_1$, m_2, m_1 are big prime numbers，then the chain fields between $GF(2^n)$ and $GF(2)$ in descending order should be $GF(2^m) \underset{m2}{>} GF(2^{m1}) \underset{m1}{>} GF(2)$, that is the composite field $GF((2^{m1})^{m2})$ in short.

Suppose $U(x), V(x) \in GF(2^n)$, $P(x)$ is the irreducible polynomial of $GF(2^n)$, compute $W(x)= U(x)V(x)\bmod P(x)$. As is known, in the elliptic curve cryptosystem, m should be bigger than 160 from the perspective of security. If there is the same computation as that in section 2 like the module multiplication $W(x)= U(x)\ V(x)$ mod $P(x)$ in finite field $\in GF(2^n)$,the speed will be very slow because of the serialization in the computation. Thus, it is necessary to adopt the design of parallel operation and the actualization of this design is operated in the chain fields in descending sort with an eigenvalue of 2. The following is the description of the parallel multiplication structure of the standard data in chain fields in descending sort, the computation base is on the following three aspects:

A. The finite field $GF(2^n)$, the serial computation based on $GF(2)$

B. The chain fields in descending sort $GF(2^m) \underset{m2}{>} GF(2^{m1}) \underset{m1}{>} GF(2)$;

C. Adopting the partitioning method in KOA

Suppose $p(x) = x^{m_2} + \sum_{i=0}^{m_2-1} p_i x^i, p_i \in GF(2^m)$ is an irreducible unary primitive polynomial of finite field

GF（2^m）. Suppose the two elements of *GF*（2^m） *U(x),V(x)* are as follows respectively:

$U(x)=u_{m-1}x^{m_2-1}+\ldots+u_1x+u_0$

$V(x)=v_{m-1}x^{m_2-1}+\ldots+v_1x+v_0$

As m_2 is a big prime number（bigger than 2）, expressed as $r=m_2+1$, the partitioning methods of KOA adopted can be:

$$u(x)=x^{r/2}(0x^{r/2-1}+u_{r-2}x^{r/2-2}+\ldots+u_{r/2})+(u_{r/2-1}x^{r/2-1}+\ldots+u_0)=x^{r/2}u_h+u_l$$

$$v(x)=x^{r/2}(0x^{r/2-1}+v_{r-2}x^{r/2-2}+\ldots+v_{r/2})+(v_{r/2-1}x^{r/2-1}+\ldots+v_0)=x^{r/2}v_h+v_l \quad (10$$

(10)

According to Formula (7), the equivalence decomposition of formula (10) can be

$$W(x)=W_0(x)+x^{r/2}[w_1(x)-w_0(x)-w_2(x)]+x^rW_2(x) \bmod P(x) \quad (11)$$

The relationship between $w_1(x)$、$w_0(x)$、$w_2(x)$ and $U_h(x)$、$U_l(x)$、$V_h(x)$、$V_l(x)$ can be obtained from Formula (7). Similar to diagram1, the actualized main logic hardware of Formula (11) can be demonstrated by Fig.3:

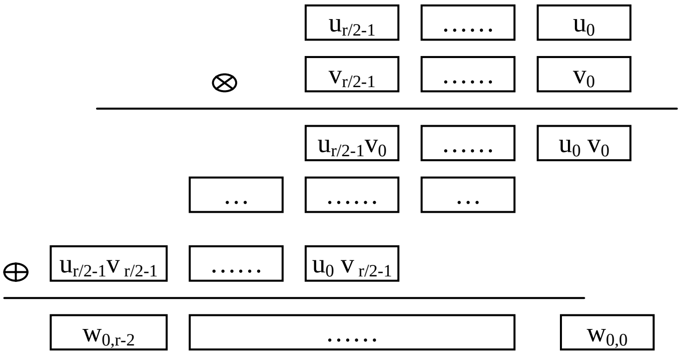

Fig.3 $u_l(x)\ \times v_l\ (x)$mod Px）

In Fig.3, u_i、v_j、$w_{0,k}\in GF$（2^{m1}）, bit width m_1, $w_{0,k}$, $i,j=0,1,\ldots,r/2-1$, $k=0,1,\ldots,r-1$. The multiplication and addition of u_i、v_j as the polynomial coefficient are all operated in *GF*（2^{m1}）, yet u_iv_j will be accomplished by the multiplication computation in section 2. Based on the LFSR in section 2 comes the feedback shift register computing $x^{r/2}[w_1(x)-w_0(x)-w_2(x)]$ mod $P(x)$ and $x^rw_2(x)$ mod

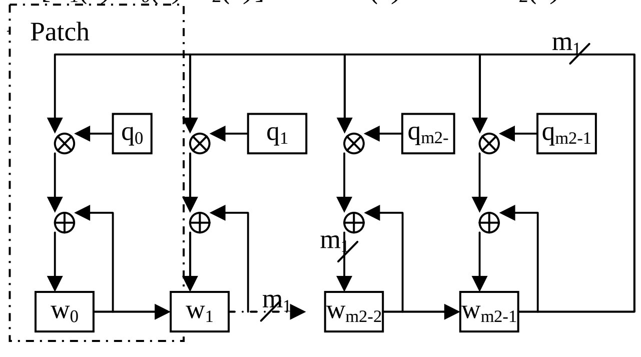

Fig.4 LFSR in composite field

Different from Fig.2, the computation bit width in Fig.4 is $m_1$bits. The multiplication and addition in Fig. 4 and 3 are accomplished by the computations in Fig. 2 and 1 together. If the multiplication and addition are divided into patches, Fig. 4 and 3 can be divided into m_2 patches. The multiplication and addition in each patch are operated in serial, while the computations between the patches are operated in parallel. The shift in Fig.4 is in serial from w_0 to w_{m2-1}, but the bit number of each transfer is $m_1$bits, thus it is a parallel operation. If counted by the bit serial shift, it needs *m* clock cycles. If the shift is to be done as shown in Fig.4, it needs only m_2 clock cycles.

4 Conclusion

This paper proposes a new parallel multiplier structure based on the composite field *GF*（（2^{m1}）^{m2}）. Aiming at the elliptic curve cryptosystem, the multiplier adopts the computations in parallel and serial. It has a fast computation speed and a regular structure. Besides, it is easy to be adopted in module design and suitable to the actualization of VLSI.

References:

[1] Sebastian TJF, Mohammed B, David T. Multiplication and Division over the Dual Basis. *On IEEE Tran Computers*1996 , 45(3): 319-327 [J].

[2] LU Jun-Sheng ZhANG Wen-Xiang WANG Xin-Hui. A Fast Multiplier Design and Implication over Finite Fields. JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT. 2004 Vol.41 No.4 P.755-760

[3] Massey Jl Omura Jk Computational Method and Apparatus for Finite Field Arithmetic U.S. Patent application, submitted, 1981.

[4] Charles C W , Truong HM Shao LJ, et al. VLSI Architectures for Computing Multiplications and Inverses in GF(2^m), *IEEE Trans on Computers* 1985, (C-34)8:709-717.

[5] Scott PA. Tavares SS Peppard LE. A Fast Multiplier for GF(2^m)[J]. *IEEE J Select Areas common.*,1986.

[6] Paar C. A New Architectures for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields [J] *IEEE Trans on Computers* 1996, 45(7): 856-861.

[7] Antonio. A New Algorithm for Multiplication in Finite Fields[J] *IEEE Trans on Computers*, 989 ,38(7):1045-1049.

[8] Hsu IS, Truong TK., Deutsch LJ, Reed I.S. A Comparison of VLSI Architecture of Finite

Fields Multipliers Using Dual, Normal, or Standard Bases[j]. *IEEE Trans on Computers*, 1988,37(6):735-739.

[9] Lin J C, Chang C T, Chung W T. Design, Implementation and Performance Evaluation of IP-VPN. 17th International Conference on Advanced Information Networking and Applications, 2003. 206-209.

[10] Pena C J , Evans J. Performance Evaluation of Software Virtual Private Networks (VPN). Proceedings of 25th Annual IEEE Conference on Local Computer Networks. New York: IEEE Computer Society, 2000. 522-523.