논문 2004-41SC-1-5

유한체 $GF(2^n)$에서 낮은 공간 복잡도를 가지는 새로운 다중 분할 카라슈바 방법의 병렬 처리 곱셈기

(A New Low Complexity Multi-Segment Karatsuba Parallel Multiplier over $GF(2^n)$)

장 남 수*, 한 동 국*, 정 석 원**, 김 창 한***

(Nam-Su Chang, Dong-Guk Han, Seok-Won Jung, and Chang Han Kim)

요 약

유한체 $GF(2^n)$에서 두 원소의 곱셈을 수행하는 공간 복잡도가 낮은 병렬 처리 곱셈기의 구현에 있어서 divide-and-conquer 방법은 유용하게 사용된다. 이를 이용한 가장 널리 알려진 알고리듬으로는 카라슈바 (Karatsuba-Ofman) 알고리듬과 다중 분할 카라슈바(Multi-Segment Karatsuba) 알고리듬이 있다. Leo ne은 카라슈바 알고리듬의 최적화된 반복 횟수를 제안하였고, Ernst는 다중 분할 카라슈바 방법을 이용한 일반적이고 확장 가능한 유한체 곱셈기를 제안하였다. 본 논문에서는 Ernst가 제시한 다중 분할 카라슈바 병렬 처리 곱셈기의 복잡도를 제시한다. 또한 기존 방법의 병렬 처리 곱셈기와 시간 복잡도는 같지만 공간 복잡도는 낮은 새로운 다중 분할 카라슈바 방법의 병렬 처리 곱셈기를 제안하며 그에 따른 최적화된 반복 횟수를 제안한다. 나아가서 제안하는 곱셈기가 몇몇 유한체에서 카라슈바 방법의 병렬 처리 곱셈기 보다 공간 복잡도에서 효과적임을 제시한다.

Abstract

The divide-and-conquer method is efficiently used in parallel multiplier over finite field $GF(2^n)$. Leone proposed optimal stop condition for iteration of Karatsuba-Ofman algorithm(KOA). Ernst et al. suggested Multi-Segment Karatsuba(MSK) method. In this paper, we analyze the complexity of a parallel MSK multiplier based on the method. We propose a new parallel MSK multiplier whose space complexity is same to each other. Additionally, we propose optimal stop condition for iteration of the new MSK method. In some finite fields, our proposed multiplier is more efficient than the KOA.

Keywords: Elliptic Curve Cryptosystem, Karatsuba-Ofman Algorithm, Multi-Segment Karatsuba

I. 서 론

V. Miller[10]와 N. Kobliz[8]에 의하여 타원곡선 암호(ECC)는 처음으로 제안되었다. 타원곡선 암호는 일반적으로 RSA(Rivest-Shamir-Adleman)와 비교하여 작은 키 사이즈로 RSA와 같은 안전성을 가질 수 있는 것으로 알려져 있다. 또한 작은 키 사이즈(Key Size)는 시스템을 효과적으로 구성할 수 있게 한다. 타원곡선 암호시스템의 수행에 있어서 유한체 연산은 주요 관심의 대상이 된다. 타원곡선 암호에 대한 관심의 증가로 몇몇 새로운 유한체 연산의 구조가 제안되었다[5]. 유한체에서 기본적인 연산들 중 곱셈은 가장 중요한 고려 사항이다. 또한 하드웨어에서는 시간과 공간의 복잡도의 효율성을 요구한다. 따라서 곱셈 연산량의 감소는 공간복잡도 감소에 많은 효과를 가져 올 수 있다. 일반적으로 $GF(2^n)$에서 공간 복잡도는 XOR 게이트와 AND 게이트의 수

* 학생회원, **정회원, 고려대학교 정보보호대학원
(Center for information and security Technologies(CIST), Korea Univ.)
*** 정회원, 세명대학교 정보보호학과
(Dept. of Information and Security Semyung University.)
※ 이 논문은 세명대학교 IT학과 장비지원사업 연구과제비를 지원 받았음
접수일자: 2003년6월3일, 수정완료일: 2003년12월22일

를 의미하며, XOR 게이트와 AND 게이트의 지연시간을 각각 T_X와 T_A로 표현할 때 시간 복잡도는 조합논리로 구성된 병렬구조에서 최대 지연 시간을 갖는 경로에 대한 T_X와 T_A의 배수로 표현된다.

스마트 카드와 모바일 폰 같은 소형 장비의 적용에 있어서 가장 중요한 디자인 관점은 공간 복잡도의 감소이다. 이런 이유로 Leone은 [9]에서 낮은 공간 복잡도를 가지는 카라슈바 방법(KOA)의 병렬 처리 곱셈기를 제안하였으며, 이는 시간 복잡도와 공간 복잡도 사이의 교환을 이용한 것이다. 또한 [4]에서 Ernst는 일반적으로 확장 가능한 구조를 가지는 다중분할 카라슈바 방법(MSK)의 곱셈기를 제안하였다. 그러나 Ernst가 제안한 곱셈기는 병렬구조가 아니며 곱셈기에 대한 복잡도도 제시되지 않았다.

본 논문에서는 [4]방법을 이용한 병렬 처리 곱셈기와 이때의 시간-공간 복잡도를 제시한다. 또한 불필요한 패턴을 가지는 기존의 MSK 방법을 개선하여 더 낮은 공간 복잡도를 가지는 새로운 MSK방법의 병렬처리 곱셈기를 제안한다. 예를 들어, 5중분할(5-segment) 방법으로 곱셈기를 구성하면 기존의 방법은 $O(n^{1.68})$의 곱셈을 수행하지만 본 논문에서 제안하는 방법으로 하면 $O(n^{1.63})$의 곱셈으로 수행이 가능하다. 또한 7중분할(7-segment) 방법으로 곱셈기를 구성하면 기존의 방법은 $O(n^{1.73})$의 곱셈을 수행하지만 본 논문에서 제안하는 방법으로 하면 $O(n^{1.65})$의 곱셈으로 수행이 가능하다.

일반적으로 MSK 방법은 KOA 방법 보다 비효율적이다. 하지만 본 논문에서는 [9]의 방법을 MSK에 확장하여 적용한 병렬처리 곱셈기가 [9]에서 제안된 방법보다 몇몇 유한체에서 효과적임을 제시한다.

Ⅱ. 유한체 $GF(2^n)$에서의 다항식의 곱셈

본 절에서 $GF(2)$에서의 n 차 기약다항식 $f(x)$에 의하여 생성된 유한체 $GF(2^n)$의 원소간의 곱셈에 관하여 살펴보자. 다항식 기저에 의해 $GF(2^n)$의 원소 $a(x)$와 $b(x)$는

$$a(x)=a_0+a_1x+\cdots+a_{n-2}x^{n-2}+a_{n-1}x^{n-1},$$

$$b(x)=b_0+b_1x+\cdots+b_{n-2}x^{n-2}+b_{n-1}x^{n-1}$$

와 같이 표현되며 이때 $a_i,\ b_i \in GF(2)$이다. 유한체 위의 곱셈은 두 과정으로 나뉜다. 첫 번째는 유한체의 두 원소 $a(x)$와 $b(x)$를 $n-1$차 다항식으로 보고 곱하는 다항식 곱셈 과정으로 $a(x)$와 $b(x)$의 곱으로 $2n-2$차 다항식을 얻는다. 두 번째 과정은 모듈로 축소(Modulo Reduction)로 $2n-2$차 다항식 $c(x)$를 n차 기약 다항식 $f(x)$로 나눈 나머지를 구하는 과정이다. 효율적인 곱셈을 위해서는 다항식 곱셈 과정과 모듈로 축소과정을 최적화해야 한다. 본 논문은 모듈로 축소 과정을 배제하고 다항식 곱셈 과정을 효율적으로 계산하는 방법을 제시한다. 먼저 일반적인 곱셈 방법인 SchoolBook(SB) 방법의 복잡도를 살펴보자.

SchoolBook(SB)방법을 이용하여 유한체 원소의 다항식 곱셈을 수행하면 n^2번의 곱셈과 $(n-1)^2$번의 덧셈이 부분체 $GF(2)$에서 수행된다. 전체 AND 게이트 수, XOR 게이트 수, 전체 게이트 수를 각각 #AND, #XOR, #TOT라 정의하고 전체의 시간 지연(Time Delay)을 T_{TOT}라 정의하면 SB방법의 병렬처리 곱셈기의 복잡도는 다음과 같다.

$$\#AND = n^2,$$
$$\#\mathrm{XOR} = (n-1)^2,$$
$$\#TOT = 2n^2-2n+1,$$
$$T_{TOT} = T_A+\lceil \log_2 n \rceil T_X.$$

2.1. 카라슈바(KOA) 방법의 곱셈

$GF(2^n)$에서 기본적인 KOA방법은 유한체의 원소를 2중분할하여 divide-and-conquer방법을 적용한다. $n/2$이 짝수라면 KOA는 반복 적용될 수 있다. $n-1$차의 다항식에 KOA를 직접 적용하면 $\log_2 n$번 반복 수행이 가능하다. KOA 방법은 다항식의 곱셈에 효과적으로 적용된다.

XOR 연산은 두 과정으로 나누어진다. 첫째는 $n/2$ 비트 다항식 곱셈기의 입력 값의 연산이고, 두 번째는 $n/2$ 비트 다항식 곱셈기의 출력 값의 연산이다. 만약 KOA를 1번 반복 수행한다면 첫 번째 과정 $n/2$ 개의 비트 덧셈 (A_0+A_1), (B_0+B_1)에서 $2(n/2)$개의 XOR 게이트가 필요하고 두 번째 과정 $n/2$차 곱셈기의 출력 값 A_0B_0, A_1B_1, $(A_0+A_1)(B_0+B_1)$과 계수간의 덧셈연산에서 $3n-4$개의 XOR 게이트가 필요하다. 따라서 KOA가 m번의 반복 연산을 수행하며 최하위 연산기를 SB 곱셈기를 사용한다고 가정할 때 전체 연산량은 다음과 같다.

$$\#AND = \left(\frac{3}{4}\right)^m \cdot n^2 ,$$

$$\#XOR = \left(\frac{n^2}{4^m} + \frac{6n}{2^m} - 1\right)3^m - (8n-2) ,$$

$$\#TOT = \left(\frac{2n^2}{4^m} + \frac{6n}{2^m} - 1\right)3^m - (8n-2) ,$$

$$T_{TOT} = \left(3\lceil \log_2 2^m \rceil + \left\lceil \log_2 \frac{n}{2^m} \right\rceil\right)T_X + T_A$$

$$= (2m + \lceil \log_2 n \rceil)T_X + T_A .$$

위의 결과에 의하여 KOA 방법을 이용한 병렬 처리 곱셈기는 SB 방법을 이용한 병렬 처리 곱셈기보다 공간 복잡도가 낮다. 부가적으로 [9]에서는 최적화된 KOA의 반복 횟수를 $n/2^k=4$일 때의 k로 제안하였다.

2.2. 다중분할 카라슈바 방법의 곱셈

MSK 곱셈 방법은 KOA방법을 확장하여 유한체의 원소를 다중으로 분할하여 곱한다. 유한체 $GF(2^n)$에서 $n \pmod k = 0$ 이라 가정하면 k-다중 분할 카라슈바 (MSK_k) 방법으로 두 원소의 곱셈을 수행할 수 있다. 만약 $n \pmod k \neq 0$ 이라면 필요한 만큼의 계수를 0으로 채운다. 두 원소 $a(x)$와 $b(x)$를 k-다중 분할 (k-segment)로 나누어 표현하면, $\hat{x} = x^{n/k}$라 할 때

$$a(x) = \oplus_{i=0}^{k-1} a_i \cdot \hat{x}^i ,$$

$$b(x) = \oplus_{i=0}^{k-1} b_i \cdot \hat{x}^i$$

$$c(x) = a(x) \cdot b(x) = MSK_k(A,B)$$

이며 다음과 같음을 알 수 있다.

$$MSK_k(A,B) = \left(\oplus_{i=1}^{k} S_{i,0}(A,B) \cdot \hat{x}^{i-1}\right)$$
$$\oplus \left(\oplus_{i=1}^{k-1} S_{k-i,1}(A,B) \cdot \hat{x}^{i-1+k}\right), \quad (1)$$

$$S_{m,l}(A,B) = \left(\oplus_{i=1}^{m-1} S_{i,l}(A,B)\right.$$
$$\left. \oplus \left(\oplus_{i=1}^{m-1} S_{i,l+m-i}(A,B)\right)\right) \oplus M_{m,l}(A,B), \quad (2)$$

$$S_{1,l}(A,B) = M_{1,l}(A,B)$$

$$\text{and } M_{m,l}(A,B) = \left(\oplus_{i=l}^{l+m-1} A_i\right) \cdot \left(\oplus_{i=l}^{l+m-1} B_i\right) .$$

(1)에 의해서 $c(x) = a(x) \cdot b(x) = MSK_k(A,B)$곱셈은 $S_{m,l}(A,B)$의 부분 합으로 구성된다. 각각의 부분 합은 (2)에 의하여 부분 곱 $M_{m,l}(A,B)$에 의하여 구성된다. Ernst는 (1)과 (2)를 이용하여 포괄적이고 확장 구성이 가능한 구조를 제안하였다[4]. 그러나 그들이 제시한 곱셈기는 병렬 구조의 곱셈기가 아니며 복잡도 또한 제시하지 않았다.

Ⅲ. 새로운 MSK방법의 곱셈기

본 절에서는 다중분할을 이용한 병렬 처리 곱셈기에 관하여 설명한다. 이는 간단하게 5중분할(5-Segment)을 예로 설명한다.

$GF(2^n)$에서 $n=5^m$ ($m\neq 0$)라 가정하면 $a(x)$, $b(x) \in GF(2^n)$는 5중분할로 세분되며 이를 표현하면 다음과 같다.

$$A_0 = a_0 + \cdots + a_{\frac{n}{5}-1} x^{\frac{n}{5}-1}, \quad B_0 = b_0 + \cdots + b_{\frac{n}{5}-1} x^{\frac{n}{5}-1},$$

$$A_1 = a_{\frac{n}{5}} + \cdots + a_{\frac{2n}{5}-1} x^{\frac{n}{5}-1}, \quad B_1 = b_{\frac{n}{5}} + \cdots + b_{\frac{2n}{5}-1} x^{\frac{n}{5}-1},$$

$$A_2 = a_{\frac{2n}{5}} + \cdots + a_{\frac{3n}{5}-1} x^{\frac{n}{5}-1}, \quad B_2 = b_{\frac{2n}{5}} + \cdots + b_{\frac{3n}{5}-1} x^{\frac{n}{5}-1},$$

$$A_3 = a_{\frac{3n}{5}} + \cdots + a_{\frac{4n}{5}-1} x^{\frac{n}{5}-1}, \quad B_3 = b_{\frac{3n}{5}} + \cdots + b_{\frac{4n}{5}-1} x^{\frac{n}{5}-1},$$

$$A_4 = a_{\frac{4n}{5}} + \cdots + a_{\frac{5n}{5}-1} x^{\frac{n}{5}-1}, \quad B_4 = b_{\frac{4n}{5}} + \cdots + b_{\frac{5n}{5}-1} x^{\frac{n}{5}-1},$$

따라서 n비트의 곱셈 $c(x)$는 다음과 같다.

$$c(x) = A_0 B_0$$
$$+ ((A_0 + A_1)(B_0 + B_1) + A_0 B_0 + A_1 B_1) x^{\frac{n}{5}}$$
$$+ ((A_0 + A_2)(B_0 + B_2) + A_0 B_0 + A_1 B_1 + A_2 B_2) x^{\frac{2n}{5}}$$
$$+ ((A_0 + A_3)(B_0 + B_3) + (A_1 + A_2)(B_1 + B_2)$$
$$+ A_0 B_0 + A_1 B_1 + A_2 B_2 + A_3 B_3) x^{\frac{3n}{5}}$$
$$+ ((A_0 + A_1 + A_3 + A_4)(B_0 + B_1 + B_3 + B_4)$$
$$+ (A_0 + A_1)(B_0 + B_1) + (A_0 + A_3)(B_0 + B_3) \cdot$$
$$+ (A_1 + A_4)(B_1 + B_4) + (A_3 + A_4)(B_3 + B_4)$$
$$+ A_0 B_0 + A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4) x^{\frac{4n}{5}}$$
$$+ ((A_1 + A_4)(B_1 + B_4) + (A_2 + A_3)(B_2 + B_3)$$
$$+ A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4) x^{\frac{5n}{5}}$$
$$+ ((A_2 + A_4)(B_2 + B_4) + A_2 B_2 + A_3 B_3 + A_4 B_4) x^{\frac{6n}{5}}$$
$$+ ((A_3 + A_4)(B_3 + B_4) + A_3 B_3 + A_4 B_4) x^{\frac{7n}{5}}$$
$$+ A_4 B_4 x^{\frac{8n}{5}} . \quad (3)$$

위의 식에 의하여 25번의 $(n/5)$비트 다항식 곱셈과, 20번의 $(2n/5-1)$비트 다항식의 덧셈, 그리고 8번의 $(n/5-1)$비트 다항식의 덧셈이 필요하게 된다.

3.1. 병렬 처리 MSK 곱셈기

Ernst의 방법으로 (3)을 변형하면 다음과 같은 식을 얻을 수 있다.

$$c(x) = A_0 B_0$$
$$+ ((A_0 + A_1)(B_0 + B_1) + A_0 B_0 + A_1 B_1) x^{\frac{n}{5}}$$
$$+ ((A_0 + A_1 + A_2)(B_0 + B_1 + B_2) + (A_0 + A_1)(B_0 + B_1)$$

$$+(A_1+A_2)(B_1+B_2))x^{\frac{2n}{5}}$$
$$+((A_0+A_1+A_2+A_3)(B_0+B_1+B_2+B_3)$$
$$+(A_0+A_1+A_2)(B_0+B_1+B_2)+A_1B_1+A_2B_2$$
$$+(A_1+A_2+A_3)(B_1+B_2+B_3))x^{\frac{3n}{5}}$$
$$+((A_0+A_1+A_2+A_3+A_4)(B_0+B_1+B_2+B_3+B_4)$$
$$+(A_0+A_1+A_2+A_3)(B_0+B_1+B_2+B_3)$$
$$+(A_1+A_2)(B_1+B_2)+(A_2+A_3)(B_2+B_3)$$
$$+(A_1+A_2+A_3+A_4)(B_1+B_2+B_3+B_4))x^{\frac{4n}{5}}$$
$$+((A_1+A_2+A_3+A_4)(B_1+B_2+B_3+B_4)$$
$$+(A_1+A_2+A_3)(B_1+B_2+B_3)+A_2B_2$$
$$+(A_2+A_3+A_4)(B_2+B_3+B_4)+A_3B_3)x^{\frac{5n}{5}}$$
$$+((A_2+A_3+A_4)(B_2+B_3+B_4)$$
$$+(A_2+A_3)(B_2+B_3)+(A_3+A_4)(B_3+B_4))x^{\frac{6n}{5}}$$
$$+((A_3+A_4)(B_3+B_4)+A_3B_3+A_4B_4)x^{\frac{7n}{5}}$$
$$+A_4B_4x^{\frac{8n}{5}}$$

MSK_5 방법의 병렬처리 곱셈기의 복잡도를 분석하기 위해 다음과 같이 세 과정으로 나누자.

■ 모든 MSK 방법의 곱셈은 다음과 같은 과정으로 분류된다.

STEP 1 : 다항식 곱셈 이전의 덧셈 연산

STEP 2 : 다항식 곱셈

STEP 3 : 다항식 곱셈 이후의 덧셈 연산

STEP 1에서 A_0+A_1, B_0+B_1, … , $A_0+A_1+A_2$, $+A_3+A_4$, $B_0+B_1+B_2+B_3+B_4$와 같은 조각(Segment)들의 다항식의 덧셈에서 10번의 $(n/5)$비트 덧셈(XOR)이 필요하고, STEP 2에서는 $(A_0+A_1)(B_0+B_1)$, … , $(A_0+A_1+A_2+A_3+A_4)(B_0+B_1+B_2+B_3+B_4)$와 같은 조각들의 다항식의 곱셈에서 $15\cdot TOT_{n/5}$번의 연산량이 필요하다. STEP 3에서는 15개의 곱셈기의 $2\cdot(n/5-1)$비트 출력값의 연산에서 $20\cdot(2n/5-1)$개와 $8\cdot(n/5-1)$개의 XOR 게이트가 필요하다. 추가적인 시간 지연은 STEP 1과 STEP 3에서 $5T_X$번 발생한다. 이를 종합한 MSK_5의 복잡도는 다음과 같다.

$$\#AND = 15\cdot\#AND_{n/5},$$
$$\#\mathrm{XOR} = 15\cdot\#\mathrm{XOR}_{n/5}+68\cdot(n/5)-28,$$
$$\#TOT = 15\cdot\#TOT_{n/5}+68\cdot(n/5)-28.$$

따라서 MSK_5가 m번 반복 수행할 경우 최하위를 SB 방법으로 곱셈기를 구성하면 다음과 같은 복잡도를 얻을 수 있다.

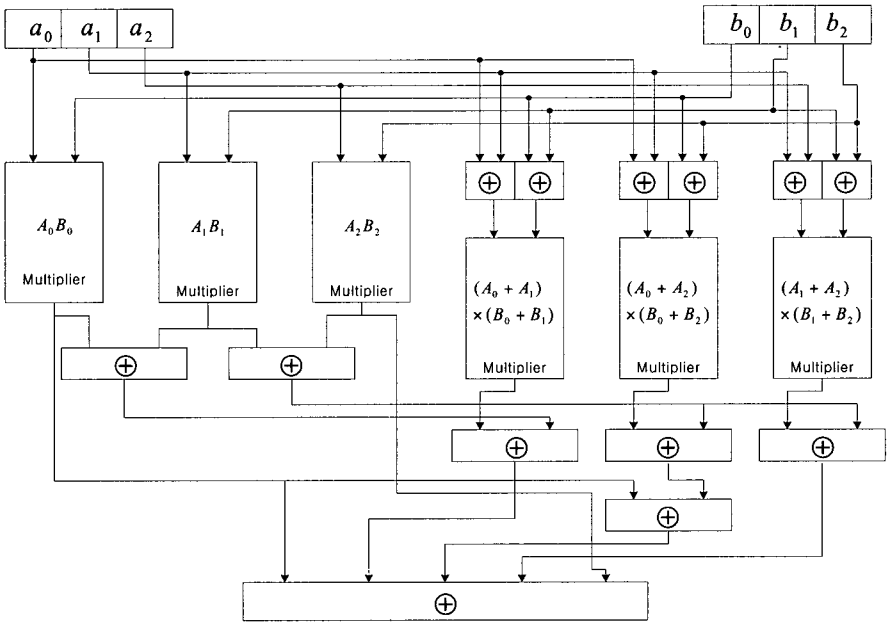


그림 1. 유한체 $GF(2^n)$의 새로운 병렬 처리 MSK_3 곱셈기

Fig. 1. Our proposed parallel MSK_3 multiplier over $GF(2^n)$.

$$\#AND = (3/5)^m \cdot n^2,$$
$$\#\mathrm{XOR} = (3/5)^m \cdot n^2 + (2/5)\cdot(12\cdot 3^m - 17)n - 15^m + 2,$$
$$\#TOT = 2\cdot(3/5)^m \cdot n^2 + (2/5)\cdot(12\cdot 3^m - 17)n - 15^m + 2,$$
$$T_{TOT} = 5m\cdot T_X + T_{TOT(n/5^m)} = (5m + \lceil \log_2 n/5^m \rceil)T_X + T_A .$$

3.2. 새로운 병렬 처리 MSK 곱셈기

본 소절에서는 효과적인 새로운 MSK 방법과 그를 이용한 병렬 처리 곱셈기와 복잡도를 제안한다. 또한 제안된 방법에 최적화된 반복횟수를 구하고 이를 적용하여 더욱 효율적으로 하드웨어를 구성할 수 있음을 보인다. Ernst방법은 불필요한 패턴을 가진다. 예를 들어 (3)에서 패턴 $(A_0+A_1+A_2+A_3)(B_0+B_1+B_2+B_3)$ 는 $A_1B_2+A_2B_1$을 포함한다. 또한 (3)의 패턴 $(A_0+A_1+A_2)(B_0+B_1+B_2)$에서도 $A_1B_2+A_2B_1$를 포함하므로 $(A_0+A_1+A_2)(B_0+B_1+B_2)$는 제거할 수 있다. 따라서 기존의 MSK방법은 최적화되지 않았음을 쉽게 확인할 수 있다. 본 논문에서 제안하는 방법은 이와 같은 불필요한 패턴을 가지지 않는다. 기존의 KOA는 본 논문에서 제안하는 방법의 특수한 형태이다. (3)을 새로운 방법을 이용하여 곱셈량을 줄이면 다음과 같이 표현된다.

$$\begin{aligned}
c(x) &= A_0B_0 \\
&+((A_0+A_1)(B_0+B_1)+A_0B_0+A_1B_1)x^{n/5} \\
&+((A_0+A_2)(B_0+B_2)+A_0B_0+A_1B_1+A_2B_2)x^{2n/5} \\
&+((A_0+A_3)(B_0+B_3)+(A_1+A_2)(B_1+B_2) \\
&\quad +A_0B_0+A_1B_1+A_2B_2+A_3B_3)x^{3n/5} \\
&+((A_0+A_1+A_3+A_4)(B_0+B_1+B_3+B_4) \\
&\quad +(A_0+A_1)(B_0+B_1)+(A_0+A_3)(B_0+B_3) \\
&\quad +(A_1+A_4)(B_1+B_4)+(A_3+A_4)(B_3+B_4) \\
&\quad +A_0B_0+A_1B_1+A_2B_2+A_3B_3+A_4B_4)x^{4n/5} \\
&+((A_1+A_4)(B_1+B_4)+(A_2+A_3)(B_2+B_3) \\
&\quad +A_1B_1+A_2B_2+A_3B_3+A_4B_4)x^{5n/5} \\
&+((A_2+A_4)(B_2+B_4)+A_2B_2+A_3B_3+A_4B_4)x^{6n/5} \\
&+((A_3+A_4)(B_3+B_4)+A_3B_3+A_4B_4)x^{7n/5} \\
&+A_4B_4x^{8n/5}.
\end{aligned}$$

이와 같이 본 논문에서 제안하는 형태로 연산을 하면 기존의 방법에서 $O(n^{1.68})$번 소요되었던 곱셈을 $O(n^{1.63})$으로 줄일 수 있다. $GF(2^5)$에서 병렬 처리 곱셈기를 구성할 때 기존의 MSK방법은 40개의 XOR 게이트와 15개의 AND 게이트가 필요하지만 본 논문에서 제안하는 방법으로 구성하면 38개의 XOR 게이트와 14개의 AND 게이트로 구성이 가능하다. 시간 지연은 기존의 방법과 같이 T_A+5T_X이다. 따라서 새로운 MSK 방법은 기존의 방법보다 공간 복잡도에서 효율적이다. 새로운 MSK_3방법의 곱셈기를 구성하면 그림 1과 같다.

새로운 MSK_5방법을 이용하여 병렬 처리 곱셈기를 구성하면 다음과 같다. STEP 1에서 A_0+A_1, B_0+B_1, … , $A_0+A_1+A_2+A_3$, $B_0+B_1+B_2+B_3$와 같은 조각들의 다항식의 덧셈에서 9번의 $(n/5)$ 비트 덧셈(XOR)가 필요하고, STEP 2에서는 $(A_0+A_1)(B_0+B_1)$, … , $(A_0+A_1+A_2+A_3)(B_0+B_1+B_2+B_3)$와 같은 조각들의 다항식의 곱셈 수행의 경우에 $14\cdot(TOT_{n/5})$번의 연산량이 필요하다. STEP 3에서는 14개의 곱셈기의 $2\cdot(n/5-1)$ 비트 출력값의 연산에서 $20\cdot(2n/5-1))$개와 $8\cdot(2n/5-1)$개의 XOR 게이트가 필요하다. 추가적인시간 지연은 STEP 1과 STEP 3에서 $5T_X$번 발생한다. 이를 종합한 복잡도는 다음과 같다.

$$\begin{aligned}
\#AND &= 14\cdot\#AND_{n/5}, \\
\#\mathrm{XOR} &= 14\cdot\#\mathrm{XOR}_{n/5}+66\cdot(n/5)-28, \\
\#TOT &= 14\cdot\#TOT_{n/5}+66\cdot(n/5)-28.
\end{aligned}$$

따라서 새로운 MSK_5가 m번 반복 수행할 경우 최하위를 SB방법으로 곱셈기를 구성하면 다음과 같은 복잡도를 얻을 수 있다.

$$\begin{aligned}
\#AND &= (14/25)^m \cdot n^2, \\
\#\mathrm{XOR} &= (14/25)^m \cdot n^2 + (2/3)\cdot(8\cdot 14/5^m - 11)n \\
&\quad -(1/13)\cdot(15\cdot 14^m - 28), \\
\#TOT &= 2\cdot(14/25)^m \cdot n^2 + (2/3)\cdot(8\cdot 14/5^m - 11)n \\
&\quad -(1/13)\cdot(15\cdot 14^m - 28),
\end{aligned}$$

그러므로 기존의 MSK방법의 곱셈기와 같은 시간 지연이 발생한다. STEP 2에서 발생하는 시간 지연을 $T_{TOT(n/5)}$라 하면 m번 반복 수행 후 다음을 얻을 수 있다.

$$T_{TOT} = 5m\cdot T_X + T_{TOT(n/5^m)} = (5m + \lceil \log_2 n/5^m \rceil)T_X + T_A .$$

표 1은 SB, MSK, 그리고 새로운 MSK방법의 곱셈기의 복잡도를 비교한 결과이다. 표 1의 결과에 의하여 새로운 MSK방법의 곱셈기가 가장 작은 공간 복잡도를 가짐을 알 수 있다. 또한 이 결과는 최적화된 반복횟수를 제한하여 공간 복잡도가 최적화됨을 보인다.

Divide-and-conquer방법은 시간과 공간 복잡도에서 교환이 일어난다. 그러므로 반복횟수가 늘어남에 따라 시간 복잡도는 증가하지만 효과적으로 공간복잡도가 감

소됨을 알 수 있다. $n=p^m t$ 라 가정하면 최대 m번의 반복 수행이 가능하다. 하지만 m은 최적화된 반복횟수가 아니다. 왜냐하면 $GF(2^p)$에서 SB 곱셈기가 MSK_p보다 낮은 공간 복잡도를 가지기 때문이다. 그러므로 [4]의 아이디어를 적용하면 $GF(2^{p^m t})$에서 $m \neq 0$이라 가정하면 MSK_p의 최적화된 반복횟수는 $n/p^k \neq p$를 만족하는 k이다.

표 1. $GF(2^n)$ 병렬 처리 곱셈기의 복잡도 비교
Table. 1. Comparing complexity of parallel multi- plier over $GF(2^n)$ between SB, MSK_5 and proposed MSK_5.

Multiplication Method		n=5	n=25	n=125	n=625
SchoolBook Method	#Gate	41	1,201	31,001	780,001
	Delay	4	6	8	11
MSK_5[4]	#Gate	55	1,137	18,727	289,377
	Delay	6	12	18	24
Optimal condition MSK_5[4]	#Gate	41	927	15,577	242,127
	Delay	4	10	16	22
Proposed MSK_5	#Gate	52	1,030	16,042	232,810
	Delay	6	12	18	24
Optimal condition Proposed MSK_5	#Gate	**41**	**876**	**13,866**	**202,626**
	Delay	**4**	**10**	**16**	**22**

표 2. 특수한 유한체 $GF(2^n)$에서의 병렬처리 곱셈기의 복잡도 비교
Table. 2. Comparing complexity of parallel multiplier in special finite field $GF(2^n)$.

Multiplication Method		n=113	n=131	n=161
SchoolBook Method	#Gate	25,313	34,061	51,521
	Delay	8	9	9
KOA[9]	#Gate	**12,343**	16,256	23,461
	Delay	**18**	17	17
Proposed MSK_3	#Gate	13,772	**15,356**	**21,710**
	Delay	13	**16**	**18**
Proposed MSK_5	#Gate	13,886	19,060	25,018
	Delay	19	16	16
Proposed MSK_7	#Gate	15,782	19,227	28,262
	Delay	13	17	13

Multiplication Method		n=163	n=193	n=200
SchoolBook Method	#Gate	52,813	74,113	79,601
	Delay	9	9	9
KOA[9]	#Gate	**23,461**	31,953	31,987
	Delay	**17**	17	19
Proposed MSK_3	#Gate	27,632	35,066	35,066
	Delay	16	16	16
Proposed MSK_5	#Gate	25,081	31,760	**31,760**
	Delay	16	16	**16**
Proposed MSK_7	#Gate	30,692	**30,887**	44,342
	Delay	13	**17**	13

Ⅳ. 비교와 결론

MSK_3과 MSK_7의 방법으로 병렬 처리 곱셈기를 구성할 경우의 복잡도는 부록 1과 2에 제안되어 있다. 표 2는 KOA와 MSK_k ($k=3,5,7$)방법의 연산량과 시간지연을 비교한 결과이다. 또한, 이 결과는 최적화된 반복회수를 적용하여 최하위 연산을 SB방법으로 구성한 결과이다.

KOA 방법은 차수가 2의 거듭제곱의 형태를 가질 때 적용되므로 MSK 방법 보다 차수의 증가량이 작다. 따라서 일반적으로 확장체의 차수 n이 증가함에 따라 KOA 방법이 MSK 방법의 곱셈기 보다 더 낮은 복잡도를 가진다.

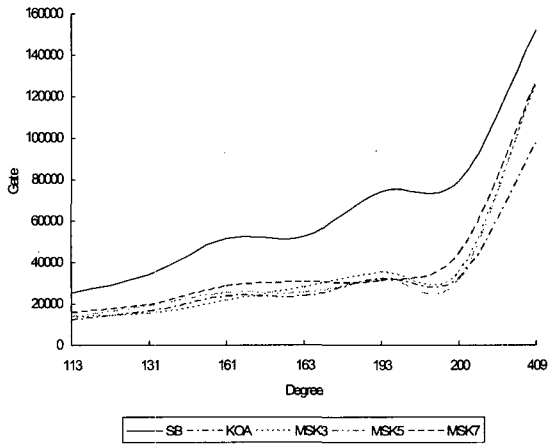


그림 2. 유한체 차수에 따른 게이트 수 비교
Fig. 2. Comparing the number of gates over $GF(2^n)$.

그러나 표 2의 결과에서처럼 본 논문에서 제안하는 MSK 방법으로 구성하면 유한체의 차수에 따라 KOA 방법 보다 시간-공간 복잡도면에서 효율적임을 알 수 있다. 또한 MSK 방법은 한번의 알고리듬 반복 시 차수가 KOA 보다 많이 증가하므로 KOA보다 시간 복잡도가 작은 곱셈기의 구성이 요구될 때 효과적으로 사용될 수 있다. Segment의 수는 소수이며 차수는 거듭제곱으로 증가하므로 큰 소수일수록 효율성이 떨어진다. 따라서 확장체의 차수가 대략 512비트 보다 작은 경우에는 MSK 방법이 효과적으로 적용될 수 있다.

본 논문에서는 기존에 제시되지 않았던 MSK 방법의 병렬처리 곱셈기의 시간-공간 복잡도를 제시하였고, 또한 기존의 것보다 낮은 공간 복잡도를 가지는 병렬처리 곱셈기를 구성하는 방법을 제안하였다. 이러한 특

성은 특히 스마트 카드와 토큰 하드웨어, 모바일 폰 등의 낮은 공간 복잡도를 요구하는 장비에 효과적으로 적용될 수 있다.

참 고 문 헌

[1] ANSI X9.62, "Public key cryptography for the financial services industry : The Elliptic Curve Digital Signature Algorithm (ECDSA)", (available from the ANSI X9 catalog), 1999.

[2] H. Cohen, "A Course in Computational Alge bric Number Theory", Springer-Verlag, Berlin, Heidelberg, 1993.

[3] G. Drolet, "A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Compl-exity Arithmetic circuit}, IEEE Trans. on Computers, vol 47, 1998, 353-356.

[4] M. Ernst, M. Jung, F. Madlener, S. Huss, and R. Blümel, "A Reconfigurable System on Chip Implementation for Elliptic Curve Ctryptogra-phy over $GF(2^n)$", In Work shop on Crypto-graphic Hardware and Embedded Systems (CHES'02), LNCS2523, (2002), 381-399.

[5] IEEE 1363, "Standard Specifications For Public Key Cryptography", http://grouper.ieee.org/gro ups /136 3/,2000. 381-399.

[6] K.O. Geddes, S.R. Czapor, and G.Labahn,"Al-gorithms for Computer Algebra, Kluwer Aca-demic Publishers", 1992.

[7] C. K. Koc, and B. Sunar, "Low- Complexity Bit-Parallel Canonical and Normal Basis Multi pliers for a Class of Finite Fields", Proceeding of 1998 IEEE International Symposium on Information Theory, MIT, Cambridge, Massachu setts, August 16-21, 1998.

[8] N. Koblitz, "Elliptic Curve Ctypto- systems", Mathmetics of Computation, vol. 48, 1987, 203-209

[9] M. Leone, "A New Low Complexity Parallel Multiplier for a Class of Finite Fields", In Workshop on Cryptographic Hardware and Embedded Systems (CHES'01),LNCS2162, (2001), 160-170.

[10] V. Miller, "Use of Elliptic Curve Cryptosyst ems", Advances in Cryptology, CRYPTO'85, LNCS 218, H. C. Williams,Ed.,Springer-Verlag, 1986, 417-426.

[11] C. Paar, "Efficient VLSI Architecture for Bit -Parallel Computation in Galois Fields", PhD thesis, (Engl. transl.), Institude for Experimen tal Mathmatics, University of Essen, Essen, Germany, June 1994.

[12] C. Paar, "Low complexity parallel Multipliers for Galois fields $GF((2^n)^4)$ based on special types of primitive polynomials, In 1994 IEEE International Symposium on Information The ory, Trondheim, Norway, June 27-July 1 1994.

[13] Paar C., "A new architecture for a parallel finite fields multiplier with Low Complexity Based on Composite Fields", IEEE Trans. on Computers, vol45, no. 7, ,July 1996, 846-861.

[14] C. Paar, P. Fleischmann, P. Roelse, "Efficient Multiplier Architectures for Galois Fields $GF((2^n)^4)$, IEEE Transactions on Compute rs, February 1998, vol. 47, no. 2, 162-170.

부 록

1. 새로운 3중분할 방법

유한체 $GF(2^n)$의 임의의 두 원소 $a(x)$, $b(x)$의 곱 $c(x)=a(x)b(x)$를 $MSK_3(a,b)$를 이용하여 계산하면 다음과 같이 표현된다.

$$
\begin{aligned}
c(x) &= a(x)\cdot b(x) \\
&= A_0B_0 \\
&+((A_0+A_1)(B_0+B_1)+A_0B_0+A_1B_1)x^{\frac{n}{3}} \\
&+((A_0+A_2)(B_0+B_2)+A_0B_0+A_1B_1+A_2B_2)x^{\frac{2n}{3}} \\
&+((A_1+A_2)(B_1+B_2)+A_1B_1+A_2B_2)x^{\frac{3n}{3}} \\
&+A_2B_2x^{\frac{4n}{3}}.
\end{aligned}
$$

위의 식에서와 같이 새로운 3중 분할 방법의 복잡도는 기존의 3중 분할 방법과 같다. 위와 같은 방법을 m

번 반복하여 최하위를 SB방법의 곱셈기로 구성하면 다음과 같은 시간과 공간 복잡도를 가진다.

$$\#AND = \left(\frac{2}{3}\right)^m n^2,$$

$$\#\mathrm{XOR} = \left(\frac{2}{3}\right)^m n^2 + \frac{2}{3}(2^{m+3}-11)n-(6^m-2),$$

$$\#TOT = 2\left(\frac{2}{3}\right)^m n^2 + \frac{2}{3}(2^{m+3}-11)n-(6^m-2)$$

$$T_{TOT} = 4mT_X + T_{TOT(n/3^m)}$$
$$= (4m+\lceil \log_2 n/3^m \rceil)T_X + T_A.$$

2. 새로운 7중분할 방법

유한체 $GF(2^n)$의 임의의 두 원소 $a(x)$, $b(x)$의 곱 $c(x)=a(x)b(x)$를 기존의 $MSK_7(a,b)$를 이용하여 계산하면 복잡도는 다음과 같음을 알 수 있다.

$$\#AND = \left(\frac{4}{7}\right)^m n^2,$$

$$\#\mathrm{XOR} = \left(\frac{4}{7}\right)^m n^2 + \frac{2}{21}(38\cdot 4^m-59)n$$
$$-\frac{1}{27}(17\cdot 28^m-44),$$

$$\#TOT = 2\left(\frac{4}{7}\right)^m n^2 + \frac{2}{21}(38\cdot 4^m-59)n$$
$$-\frac{1}{27}(17\cdot 28^m-44),$$

$$T_{TOT} = 7mT_X + T_{TOT(n/7^m)}$$
$$= (7m+\lceil \log_2 n/7^m \rceil)T_X + T_A.$$

분할의 수가 증가할수록 다중 분할 방법은 KOA방법에 비하여 공간 복잡도면에서 장점이 없어진다. 따라서 분할 방법으로 곱셈기를 구성할 때 가장 낮은 공간 복잡도를 가지기 위해서는 혼합(Hybrid)방법으로 나아가야 할 것이다.

본 논문에서 제안하는 새로운 7중분할 방법으로 기존의 방법을 변형하여 병렬 처리 곱셈기를 구성하면 다음과 같은 복잡도를 얻을 수 있다.

$$\#AND = \left(\frac{25}{49}\right)^m n^2,$$

$$\#\mathrm{XOR} = \left(\frac{25}{49}\right)^m n^2 + \frac{1}{9}\left(47\left(\frac{25}{7}\right)^m-65\right)n$$
$$-\frac{1}{24}(29\cdot 25^m-53),$$

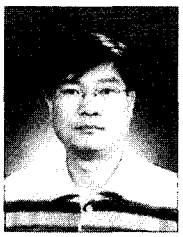
$$\#TOT = 2\left(\frac{25}{49}\right)^m n^2 + \frac{1}{9}\left(47\left(\frac{25}{7}\right)^m-65\right)n$$
$$-\frac{1}{24}(29\cdot 25^m-53),$$

$$T_{TOT} = 7mT_X + T_{TOT(n/7^m)}$$
$$= \left(7m+\left\lceil \log_2 \frac{n}{7^m} \right\rceil\right)T_X + T_A.$$

저 자 소 개

장 남 수(학생회원)

2002년 2월 서울시립대학교 수학과 학사. 2002년 3월 ~ 현재 고려대학교 정보보호대학원 석사과정. <주관심분야 : 공개키 암호, 무선LAN 기술, 암호칩 설계 기술>



정 석 원(정회원)

1991년 2월 고려대학교 수학과 졸업. 1993년 2월 : 고려대학교 수학과 석사. 1997년 2월 : 고려대학교 수학과 박사. 1997년 5월~1997년 11월: 한국전자통신연구원 박사후 연구원. 1999년 2월 ~ 2001년 2월: (주)텔리맨 책임연구원. 2002년 3월 ~ 현재 : 고려대학교 정보보호 대학원 조교수. <주관심분야 : 암호칩 설계, 부채널 공격 방법론, 공개키 암호알고리즘, 디지털 방송 보안>

한 동 국(학생회원)

1999년 2월 고려대학교 수학과 학사. 2002년 2월 고려대학교 수학과 석사. 2002년 3월 ~ 현재 고려대학교 정보보호대학원 박사과정. <주관심분야 : 수론, 공개키 암호, CMVP, 부채널 공격>

김 창 한(정회원)

1985년 2월 고려대학교 수학과 학사. 1987년 2월 고려대학교 수학과 석사. 1992년 2월 고려대학교 수학과 박사. 2002년 2월 ~ 현재 세명대학교 컴퓨터수리정보학과 부교수. <주관심분야 : 정수론, 공개키 암호, 암호 프로토콜>