논문 2007-44SD-9-1

$GF(2^n)$ 곱셈을 위한 효율적인 MSK_k 혼합 방법

(Efficiently Hybrid MSK_k Method for Multiplication in $GF(2^n)$)

지 성 연*, 장 남 수**, 김 창 한****, 임 종 인***

(Sung Yeon Ji, Nam Su Chang, Chang Han Kim, and Jongin Lim)

요 약

유한체 $GF(2^n)$ 연산을 바탕으로 구성되는 암호시스템의 효율적 구현을 위하여 유한체의 곱셈의 하드웨어 구현은 중요한 연구 대상이다. 공간 복잡도가 낮은 병렬 처리 유한체 곱셈기를 구성하기 위하여 Divide-and-Conquer와 같은 방식이 유용하게 사용된다. 대표적으로 Karatsuba와 Ofman이 제안한 카라슈바(Karatsuba-Ofman) 알고리즘과 다중 분할 카라슈바(Multi-Segment Karatsuba) 방법이 있다. Leone은 카라슈바 방법을 이용하여 공간 복잡도 효율적인 병렬 곱셈기를 제안하였고, Ernst는 다중 분할 카라슈바 방법의 곱셈기를 제안하였다. [2]에서 제안한 방법을 개선하여 [1]에서 낮은 공간 복잡도를 필요로 하는 MSK_5 방법과 MSK_7 방법을 제안하였으며, [3]에서 곱셈 방법을 혼합하여 곱셈을 수행하는 방법을 제안하였다. 본 논문에서는 [3]에서 제안한 혼합 방법에 [1]에서 제안한 MSK_5 방법을 추가로 혼합하는 혼합 방법을 제안한다. 제안하는 혼합 방법을 적용하여 곱셈을 구성하면 $l>0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하는 차수에서 [3]에서 제안한 혼합 방법보다 $116\cdot 3^l$만큼의 게이트와 $2T_X$ 만큼의 시간 지연이 감소한다.

Abstract

For an efficient implementation of cryptosystems based on arithmetic in a finite field $GF(2^n)$, their hardware implementation is an important research topic. To construct a multiplier with low area complexity, the divide-and-conquer technique such as the original Karatsuba-Ofman method and multi-segment Karatsuba methods is a useful method. Leone proposed an efficient parallel multiplier with low area complexity, and Ernst at al. proposed a multiplier of a multi-segment Karatsuba method. In [1], the authors proposed new MSK_5 and MSK_7 methods with low area complexity to improve Ernst's method. In [3], the authors proposed a method which combines MSK_2 and MSK_3. In this paper we propose an efficient multiplication method by combining MSK_2, MSK_3 and MSK_5 together. The proposed method reduces $116\cdot 3^l$ gates and $2T_X$ time delay compared with Gather's method at the degree $25\cdot 2^l-2^l<n\le 25\cdot 2^l$ with $l>0$.

Keywords: Karatsuba-Ofman, Multi-Segment Multiplier, Parallel Multiplier, Elliptic Curve Cryptosystem

I. 서 론

V.Miller와 N.Kobliz에 의하여 제안된 타원곡선 암호(Elliptic Curve Cryptosystem)는 이산대수 문제(Discrete Logarithm Problem)를 기반으로 구성되었다[5][9]. 타원곡선 암호는 기존의 인수분해 문제를 기반으로 구성된 RSA(Rivest-Shmire-Adleman) 암호 시스템보다 작은 키 사이즈(Key Size)로 같은 안정성을 가진다고 알려져 있다. 타원곡선 암호는 유한체 연산을 통하여 구성되며, 유한체 연산은 타원곡선 암호 뿐 아니라 XTR, ElGamal 타입 암호 등의 관련 응용 분야에 활발하게 사용되므로 유한체 연산의 효율적인 구성은 주요 관심의 대상이다[7~8]. 유한체 연산에서 곱셈은 주요한 연산 중 하나이고, 하드웨어 구현은 공간 복잡도와 시간 복잡도를 통하여 효율성을 비교한다. 따라서 하드웨어에서 곱셈 연산의 복잡도를 효율적으로 구성하는 것은 전체 암호 시스템의 복잡도 감소에 많은 영

* 학생회원-주저자, ** 학생회원, *** 정회원-교신저자,
고려대학교 정보경영공학전문대학원
(Graduate School of Information Management and Security, Korea University)
**** 정회원, 세명대학교 정보통신학부
(Dept. of Information and Security Semyumg Univ.)
※ "본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2006-(C1090-0603-0025))]
접수일자: 2007년4월5일, 수정완료일: 2007년9월5일

향을 준다. 본 논문에서는 유한체 $GF(2^n)$에서 효율적인 곱셈기를 구성하기 위한 방법에 대하여 논한다.

스마트 카드, 모바일 폰, PDA와 같은 소형 장비는 저장 공간 활용에 제약이 있으므로 타원곡선과 같은 암호 시스템을 소형 장비에 적용하기 위하여 가장 중요한 관점은 공간 복잡도를 줄이는 것이다. 일반적으로 유한체 $GF(2^n)$에서 공간 복잡도는 필요한 *XOR* 게이트와 *AND* 게이트 수를 계산하여 측정하고, *XOR* 게이트와 *AND* 게이트의 공간 활용 비율은 하드웨어 구현 공정에 따라 다르므로 본 논문에서는 *XOR* 게이트와 *AND* 게이트 수의 합을 전체 게이트 수로 정의하고, 전체 게이트 수를 줄이기 위한 방법에 관하여 논한다.

Karatsuba 곱셈기의 공간 복잡도를 줄이기 위한 방법으로 Leone은 1962년 Karatsuba와 Ofman이 제안한 *KOA*(Karatsuba-Ofman Algorithm) 방법을 적용하여 시간 복잡도는 증가하지만 낮은 공간 복잡도를 가지는 병렬 처리 곱셈기를 제안하였다[6]. [2]에서는 [6]에서 제안한 *KOA* 방법에서 분할 수를 증가시켜 양의 정수 k에 대하여 k-다중 분할 방법으로 확장할 수 있는 다중분할 카라슈바 방법(Multi-Segment Karatsuba: MSK_k 방법)을 제안하였다. 또, [1]에서는 [2]에서 제안한 MSK_k 방법이 k가 5 또는 7일 경우, 불필요한 패턴을 가지는 것을 개선하여 [2]에서 제안한 MSK_5 방법과 MSK_7 방법보다 낮은 공간 복잡도를 가지는 새로운 MSK_5, MSK_7 방법을 제안하였다. 그러나 일반적으로 $k>2$에 대해서 MSK_k 방법은 MSK_2(=*KOA*: MSK_k 방법에서 k가 2일 경우는 *KOA* 방법과 동일하다. 따라서 본 논문에서는 $k>1$에 대한 MSK_k 방법에 대하여 서술하므로 *KOA* 방법을 MSK_2로 표기하도록 한다.) 방법보다 곱셈을 구성할 경우 필요한 전체 게이트 수가 작아지는 차수가 드물게 나타나므로 비효율적이다. 따라서 [1]에서 낮은 공간 복잡도를 가지는 곱셈기를 구성하기 위해서 MSK_k 방법을 혼합하여 사용할 것을 권고 하였고, [3]에서 *SB* 방법과 MSK_2, MSK_3 방법을 혼합하여 곱셈을 구성할 경우 차수에 대한 분할 방법과 필요한 게이트 수를 제시하였다.

본 논문에서는 [3]에서 제안한 혼합 방법에 [1]에서 제안한 MSK_5 방법을 추가하는 혼합 방법을 제안하고, MSK_5 방법을 추가할 경우 [3]의 방법보다 곱셈의 효율적인 구성이 가능한 차수와 감소되는 게이트 수의 비율과 시간 지연을 제시한다. 또 제안하는 혼합 방법에 [1]에서 제안한 MSK_7 방법까지 혼합하여 곱셈을 구성할 경우 제안하는 혼합 방법을 적용하여 곱셈을 구성하는 것보다 효율성이 없으므로 혼합 방법에 추가할 필요가 없음을 살펴본다.

본 논문은 Ⅱ절에서 기존의 유한체 곱셈 방법에 관하여 살펴보고, Ⅲ절에서 제안하는 혼합 방법에 대하여 서술하며 Ⅳ 장에서 결론을 내린다.

Ⅱ. 기존의 유한체 $GF(2^n)$ 곱셈 방법

본 절에서는 유한체 $GF(2^n)$에서의 곱셈 방법에 관하여 살펴본다. n차 기약다항식 $f(x)$에 의하여 생성된 $GF(2^n)$의 원소는 다항식 기저와 $a_i \in GF(2)$에 의하여 다음과 같이 표현 된다

$$a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2+} \cdots + a_1 x + a_0. \quad (1)$$

식 (1)과 같이 표현된 $GF(2^n)$의 두 원소 $a(x)$, $b(x)$의 곱셈은 두 가지 단계로 나누어진다.

1. 곱셈 단계 : 두개의 $n-1$차 다항식 $a(x)$와 $b(x)$의 곱셈을 수행하여 $2n-2$차 다항식 $c(x)=a(x)\cdot b(x)$을 출력하는 단계,
2. 감산 단계 : 첫째 곱셈 단계의 결과인 $2n-2$차 다항식 $c(x)$를 유한체 $GF(2^n)$의 원소 $n-1$차 다항식으로 모듈러 감산(Modular Reduction)하는 단계.

본 논문에서는 효율적인 곱셈기를 구성하기 위하여 곱셈 단계를 효율적으로 구성하기 위한 곱셈 방법에 대하여 기술하도록 한다. 본 논문에서 곱셈기의 효율성을 비교하기 위하여 다음과 같은 기호를 사용 한다

- $\#XOR_{method}(n)$: method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 XOR 게이트의 수,
- $\#AND_{method}(n)$: method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 AND 게이트의 수,
- $\#TOT_{method}(n)$: method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 XOR 게이트와 AND 게이트 수의 합,
- $T_{method}(n)$: method 에 표기된 방법을 적용하여 n비트 곱셈의 시간 지연,
- (단, Method 표기가 없으면 임의의 방법을 적용한다.)
- T_X : *XOR* 연산의 시간 지연,
- T_A : *AND* 연산의 시간 지연.

본 절에서는 일반적인 SchoolBook 곱셈(SB) 방법과 Karatsuba-Ofman 곱셈(KOA) 방법, 다중 분할 곱셈 방법(MSK_k), [3]에서 제안한 혼합 MSK_k 방법에 대하여 살펴보도록 한다.

1. SB 곱셈 방법

본 소절에서는 가장 일반적으로 알려져 있는 SB 방법과 SB 방법을 적용하여 곱셈을 수행할 경우 필요한 연산량과 시간 지연에 대하여 서술한다.

식 (1)과 같이 표현 된 $GF(2^n)$의 두 원소 $a(x)$와 $b(x)$의 곱 $c(x)$는 다음과 같이 표현 된다

$$c(x)=a(x)\cdot b(x)=\sum_{i=0}^{n-1}a_ix^i\cdot\sum_{i=0}^{n-1}b_ix^i$$
$$=\sum_{i=0}^{2n-2}c_ix^i,\ c_i=\sum_{s+t=i}a^sb^t.$$

따라서 SB 방법을 적용하여 곱셈을 수행 할 경우 필요한 연산량과 시간 지연은 다음과 같다

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

2. MSK_2(=KOA) 곱셈 방법

본 소절에서는 1962년 Karatsuba와 Ofman에 의해 제안된 KOA(Karatsuba-Ofman Algorithm) 방법에 대하여 서술한다. MSK_2 방법은 Divide-and-Conquer방법으로 $\log_2 n$번 반복 수행이 가능하고, SB 방법보다 효율적인 방법으로 알려져 있다. MSK_2 방법의 기본 아이디어는 두 이차 다항식 $a(x)$, $b(x)$의 곱셈을 수행하기 위한 다음 방법과 같다.

$$a(x)=a_1x+a_0,\quad b(x)=b_1x+b_0,$$
$$c(x)=a_1b_1x^2+(a_1b_0+a_0b_1)x+a_0b_0, \tag{2}$$

$$c(x)=a_1b_1x^2+((a_1+a_0)(b_1+b_0)-a_1b_1-a_0b_0)x+a_0b_0. \tag{3}$$

식 (2)는 SB 방법을 이용하여 곱셈을 수행한 결과이고, 식 (3)은 MSK_2 방법을 이용한 곱셈 결과이다. 식에서 +와 - 는 $GF(2)$에서 연산이므로 XOR 게이트로 수행된다. 따라서 식 (2)와 (3)에서와 같이 MSK_2 방법을 이용할 경우 추가적으로 XOR 연산이 세 번 필요 하지만, 곱셈을 수행하는 AND 연산은 SB 방법 4번에서 3번으로 한 번 감소하게 된다. 유한체의 차수 n이 짝수일 경우, 두 원소 $a(x),b(x)\in GF(2^n)$에 대해 MSK_2 방법을 한번 적용하면 다음과 같다

$$a(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$$
$$=a^Hx^{2/n}+a^L,$$
$$b(x)=b_{n-1}x^{n-1}+b_{n-2}x^{n-2}+\cdots+b_1x+b_0$$
$$=b^Hx^{2/n}+b^L.$$
$$c(x)=(a^Hx^{n/2}+a^L)\cdot(b^Hx^{n/2}+b^L)$$
$$=a^Hb^Hx+(a^Hb^L+a^Lb^H)x^{n/2}+a^Lb^L$$
$$=a^Hb^Hx$$
$$+((a^H+a^L)\cdot(b^H+b^L)+a^Hb^H+a^Lb^L)x^{n/2}$$
$$+a^Lb^L.$$

a^H, a^L, b^H, b^L은 $n/2$비트의 수이므로 a^H+a^L과 b^H+b^L을 구하기 위하여 $n/2$비트 덧셈 2번, a^Hb^H, a^Lb^L과 $(a^H+a^L)(b^H+b^L)$을 구하기 위하여 $n/2$비트 곱셈 3번, $(a^H+a^L)\cdot(b^H+b^L)+a^Hb^H+a^Lb^L$을 구하기 위하여 $n-1$비트 덧셈을 두 번, 그리고 $x^{n/2}$부터 $x^{3n/2-1}$까지 중복되는 부분의 덧셈 $2(n/2-1)$을 필요로 한다. 따라서 MSK_2 방법을 한 번 수행할 경우 연산량과 시간 지연은 다음과 같다

$$\#\mathrm{AND}_{\mathrm{MSK}_2}(\mathrm{n})=3\#\mathrm{AND}(\mathrm{n}/2),$$
$$\#\mathrm{XOR}_{\mathrm{MSK}_2}(\mathrm{n})=3\#\mathrm{XOR}(\mathrm{n}/2)+4\cdot\mathrm{n}-4,$$
$$\#\mathrm{TOT}_{\mathrm{MSK}_2}(\mathrm{n})=3\#\mathrm{TOT}(\mathrm{n}/2)+4\cdot\mathrm{n}-4,$$
$$T_{MSK_2}(n)=T(n)+3T_X.$$

MSK_2 방법을 반복적으로 사용할 경우에 대하여 살펴보도록 한다. 유한체 $GF(2^n)$의 차수를 $n=2^k$라 하자. [6]에서 Leone이 제안한 최적의 반복 횟수는 $n/2^t=4$를 만족하는 t 이므로 MSK_2 방법을 $k-2$번 반복 수행하고, 최하위 곱셈은 4비트 SB 방법 곱셈기를 사용한다. 이 때, 연산량은 다음과 같다.

$$\#\mathrm{AND}_{\mathrm{MSK}_2}(\mathrm{n})=16\cdot3^{\mathrm{k}-2},$$
$$\#\mathrm{XOR}_{\mathrm{MSK}_2}(\mathrm{n})$$
$$=9\cdot3^{k-2}+32(3^{k-2}-2^{k-2})-2(3^{k-2}-1),$$
$$\#\mathrm{TOT}_{\mathrm{MSK}_2}(\mathrm{n})=55\cdot3^{\mathrm{k}-2}-32\cdot2^{\mathrm{k}-2}+2.$$

3. [1]에서 제안된 MSK_5, MSK_7 방법

본 소절에서는 2004년 [1]에서 제안한 유한체 $GF(2^n)$에서 낮은 공간 복잡도를 가지는 새로운 다중 분할 카라슈바 방법에 대하여 서술한다. 다중 분할 방

법을 적용하여 유한체 곱셈을 구성하기 위하여 2중분할, 3중분할, 5중분할 등 2보다 큰 양의 정수 k에 대하여 일반화 할 수 있는 k-다중 분할(MSK_k) 방법 공식이 [2]에 제안되었다. 그러나 [2]에 제안된 방법을 사용하여 MSK_k 방법을 표현 하면 k가 5 또는 7일 경우 불필요한 패턴을 가지게 된다. 따라서 [1]에서는 [2]에서 제안된 분할 방법 중 MSK_5, MSK_7 방법에서 불필요한 패턴을 제거하여 새로운 MSK_5, MSK_7 방법을 제안 하였다. [2]에 제안된 방법으로 $GF(2^n)$의 두 원소를 5중 분할하여 곱셈을 수행하면 15개의 $\lceil n/k \rceil$ 곱셈연산과 40개의 XOR 연산을 필요로 한다. 그러나 [1]에서 제안된 방법은 14개의 $\lceil n/k \rceil$ 곱셈연산과 38개의 XOR 연산으로 $GF(2^n)$의 두 원소의 5중 분할하여 곱셈을 수행 할 수 있다. n이 5의 배수일 경우 [1]에서 제안된 MSK_5 방법을 한번 적용할 경우 곱셈을 수행하기 위해 필요한 연산량과 시간지연은 다음과 같다

$$\#\mathrm{AND}_{\mathrm{MSK}_5}(n) = 14\#\mathrm{AND}(n/5),$$
$$\#\mathrm{XOR}_{\mathrm{MSK}_5}(n) = 14\#\mathrm{XOR}(n/5) + 66 \cdot (n/5) - 28,$$
$$\#\mathrm{TOT}_{\mathrm{MSK}_5}(n) = 14\#\mathrm{TOT}(n/5) + 66 \cdot (n/5) - 28,$$
$$T_{MSK_5}(n) = T(n/5) + 5T_X.$$

같은 방법으로 [2]에서 제안된 방식으로 MSK_7 방법을 표현할 경우 발생하는 불필요한 패턴을 없애고, 효율적으로 재구성한 [1]에서 제안된 MSK_7 방법을 7의 배수 n에 대하여 한번 적용하면 다음과 같은 연산량과 시간 지연을 필요로 한다.

$$\#\mathrm{AND}_{\mathrm{MSK}_7}(n) = 25\#\mathrm{AND}(n/7),$$
$$\#\mathrm{XOR}_{\mathrm{MSK}_7}(n) = 25\#\mathrm{XOR}(n/7) + 130 \cdot (n/7) - 53,$$
$$\#\mathrm{TOT}_{\mathrm{MSK}_7}(n) = 25\#\mathrm{TOT}(n/7) + 130 \cdot (n/7) - 53,$$
$$T_{MSK_7}(n) = T(n/7) + 7T_X.$$

[1]에서 분할의 수가 증가할수록 다중 분할 방법은 MSK_2 방법에 비하여 공간 복잡도면에서 효율적으로 곱셈을 구성할 수 있는 차수의 빈도가 드물게 나타나므로 분할 방법을 사용하여 곱셈기를 구성할 때 낮은 공간 복잡도를 가지기 위해서는 혼합(hybrid)방법으로 구성할 것을 권고 하였다.

4. [3]에서 제안한 혼합 MSK_k 방법

본 소절에서는 [3]에서 제안한 혼합 방법에 관하여 기술한다. [3]에서 SB 방법과 MSK_2, MSK_3 방법을 사용하여 혼합 방법을 구성하는 곱셈 방법을 제안하였다. [3]에서 제안한 혼합 방법은 하위 차수부터 각 차수에 대하여 SB 방법과 MSK_2, MSK_3 방법을 각각 적용할 경우 게이트 수를 구하고, 가장 작은 게이트 수를 필요로 하는 분할 방법과 게이트 수를 테이블에 저장하는 방법으로 구성하였다. 즉, $n=1$일 경우는 단일 AND 연산이므로 테이블에 1을 저장한다. 2보다 큰 n에 대하여 n비트 곱셈을 수행하기 위하여 SB 방법과 MSK_2, MSK_3 방법을 각각 적용하여 계산한다. 이 때 재귀적으로 호출되는 n보다 작은 차수에 대한 분할 방법과 게이트 수는 이미 계산되어 테이블에 저장된 값을 사용한다. 그리고 각 방법을 적용하여 수행할 경우 필요한 게이트 수를 비교하여 가장 작은 게이트를 필요로 하는 분할 방법과 게이트 수를 테이블에 저장한다. [3]에서 이와 같은 방법으로 하위 차수부터 차수 128까지 각 차수에 대하여 곱셈을 구성할 경우 차수에 대하여 맨 처음 적용하는 분할 방법과 필요한 게이트 수를 테이블을 구성하여 제시하였다. 테이블에 차수에 대한 분할 방법과 필요한 게이트 수를 저장하는 방법으로 SB 방법과 MSK_2 방법, SB 방법과 MSK_3 방법을 적용할 경우 하위 차수부터 128까지 곱셈을 구성하기 위한 게이트 수를 비교하면 그림 1과 같다.

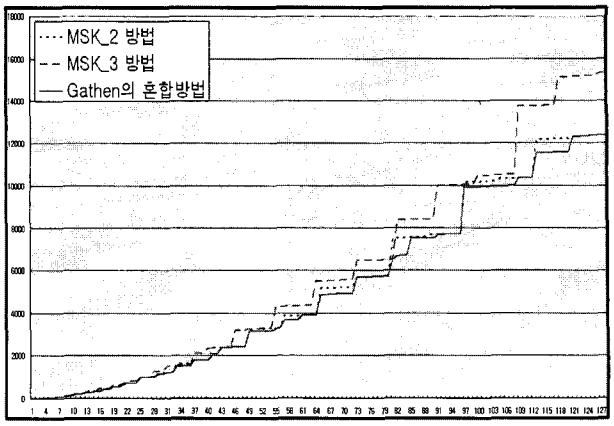

그림 1. 차수 1부터 128 까지 MSK_2 방법, MSK_3 방법, [3]에서 제안한 혼합 방법을 적용하여 곱셈을 구성할 경우 필요한 게이트 수 비교

Fig. 1. Comparison of the number of gate using MSK_2, MSK_3 methods and the hybrid method [3] following the degree.

그림 1과 같이 MSK_2 방법, MSK_3 방법을 적용하는 것 보다 SB 방법과 MSK_2, MSK_3 방법을 혼합하여 적용하여 곱셈을 구성하는 것이 공간 복잡도가 효율적이다. 따라서 [3]에서 제안한 혼합 방법이 단일 다중 분할 방법 보다 공간 복잡도가 효율적임을 알 수 있다. 그러

나 [3]의 방법은 기존의 곱셈 방법 중 *SB* 방법과 MSK_2, MSK_3 방법만을 혼합하는 혼합방법에 대하여 논하였다.

Ⅲ. 제안하는 혼합 MSK_k 방법

본 장에서는 $GF(2^n)$의 곱셈을 효율적으로 구성하기 위하여 MSK_k 방법들과 *SB* 방법을 혼합하여 구성하는 방법에서 혼합 다중 분할 방법의 확장에 따른 효율성에 관하여 서술한다. [3]에서 제안한 혼합(이하 2:3) 방법은 *SB* 방법 MSK_2, MSK_3 방법을 혼합하여 구성하였으나, [1]에서 제안한 MSK_5, MSK_7 방법과 같은 다중 분할 방법을 함께 혼합하여 구성할 수 있다. 본 논문에서는 2:3 방법에 [1]에서 제안한 MSK_5 방법을 혼합하여 분할 방법을 확장한 혼합(이하 2:3:5) 방법을 제안하고, 2:3:5 방법과 2:3 방법을 비교하여 효율적인 차수와 효율성에 관하여 서술하도록 한다. 그리고 MSK_7 방법까지 혼합할 경우에 대하여 논한다.

1. 테이블 구성

본 소절에서는 2:3 방법과 같은 방법으로 분할 방법에 MSK_5 방법을 추가하여, 하위 차수부터 차수에 따른 최소 게이트 수를 필요로 하는 차수에 따라 맨 처음 적용하는 분할 방법과 게이트 수를 저장하는 테이블 저장 방식을 사용한다. 차수 n에 따라서 *SB* 방법과 각 MSK_k 방법을 적용할 경우 곱셈을 구성하기 위한 게이트 수를 구한 후, 각 MSK_k 방법에서 필요한 게이트 수를 비교하여 최소값을 가지는 분할 방법과 게이트 수를 저장하는 방식으로 차수에 대한 맨 처음 적용하는 분할 방법과 필요한 게이트 수 테이블을 구성한다. 차수 n에 대하여 MSK_k 방법을 적용하기 위하여 n이 k의 배수가 아닐 경우 $k-(n \bmod k)$ 만큼 0으로 패딩하여 차수를 k의 배수로 만들고 MSK_k 방법을 적용한다. 차수 n에 대응하는 최소 게이트를 구할 경우, 하위 차수부터 $n-1$까지 각 차수에 대응하는 분할 방법과 그에 따른 게이트가 테이블에 저장 되어있다. 따라서 차수 n에 대하여 MSK_k 를 적용할 경우 분할 방법을 적용하기 위하여 필요한 $\lceil n/k \rceil$ 차 곱셈은 테이블에 저장되어 있는 곱셈 방법과 게이트를 사용하여 구성한다.

2. 효율적인 차수

본 소절에서는 2:3:5 방법을 적용하여 곱셈을 구성할 경우 2:3 방법 보다 효율적으로 구성할 수 있는 차수에 대하여 알아본다.

2:3 방법을 적용하여 곱셈을 구성할 경우 차수에 따른 분할 방법과 게이트 수 테이블을 살펴보면 차수가 49와 50일 경우 분할 방법은 MSK_2 방법으로 25차 곱셈을 필요로 한다. 그리고 차수가 97과 98, 99와 100일 경우 분할 방법은 MSK_2 방법으로 차수 49와 50의 곱셈을 필요로 한다[3]. 또, 같은 차수에 대하여 2:3:5 방법을 적용하여 곱셈을 구성하여도 2:3 방법과 같은 방법으로 구성된다. 위의 방법을 확장하면 다음 보조정리가 성립한다.

보조정리1. 2:3 방법을 적용하여 곱셈을 구성할 경우 $l>0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하면 차수 n에 대하여 맨 처음 적용하는 분할 방법은 MSK_2 방법이다([3] 표 1 참고). 그리고 제안하는 2:3:5 방법을 적용할 경우도 같은 범위에서 맨 처음 적용하는 분할 방법은 MSK_2 방법이 된다.

보조정리 1에서 $l>0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하는 차수 n에 대하여 2:3 방법과 2:3:5 방법을 통하여 곱셈을 구성할 때, 맨 처음 MSK_2 방법을 적용하므로 곱셈을 수행하기 위하여 필요한 분할된 곱셈의 차수는 $25\cdot 2^{l-1}-2^{l-1}<n\le 25\cdot 2^{l-1}$이 된다. 만약

표 1. 차수 24까지 제안하는 혼합 방법을 적용하여 곱셈을 구성 할 경우 분할 방법과 필요한 게이트 수
Table 1. The optimal method and the number of required gates applying propose method

degree	r	#gate	degree	r	#gate	degree	r	#gate	degree	r	#gate	degree	r	#gate
1	C	1	6	2	59	11	C	221	16	2	369	21	3	654
2	C	5	7	C	85	12	2	221	17	2	470	22	2	647
3	C	13	8	2	103	13	2	307	18	2	470	23	2	755
4	C	25	9	3	134	14	2	307	19	2	553	24	2	755
5	C	41	10	2	159	15	3	346	20	2	553	25	5	876

표 2. $l=4$일 경우 범위 $25\cdot 2^l-2^l<n\le 25\cdot 2^l$의 차수 n에 대한 게이트 수 비교
Table 2. Comparison of the number of required gates used in hybrid methods.

method \ degree(n)		385	386	387	388	389	390	391	392
2:3 방법	$\#TOT_{2:3}(n)$	92992	92992	93000	93000	93032	93032	93040	93040
	$T_{2:3}(n)$	T_A+22T_X							
2:3:5 방법	$\#TOT_{2:3:5}(n)$	83596	83596	83604	83604	83636	83636	83644	83644
	$T_{2:3:5}(n)$	T_A+20T_X							
method \ degree(n)		393	394	395	396	397	398	399	400
2:3 방법	$\#TOT_{2:3}(n)$	93144	93144	93152	93152	93184	93184	93192	93192
	$T_{2:3}(n)$	T_A+22T_X							
2:3:5 방법	$\#TOT_{2:3:5}(n)$	83748	83748	83756	83756	83788	83788	83796	83796
	$T_{2:3:5}(n)$	T_A+20T_X							

$l-1>0$이면 MKS_2 방법을 반복하여 수행하고 $l-1=0$ 이면 25차 곱셈을 수행한다. 따라서 차수 n에 대하여 MSK_2 방법을 l 번 반복 수행하고 하위 25차 곱셈을 통하여 구성된다.

보조정리2. 차수 25 에 대하여 MSK_5 방법을 적용하여 곱셈을 구성하면 SB, MSK_2, MSK_3 방법보다 적은 게이트 수를 필요로 한다.

증명) 차수가 $n=25$이므로 차수 $n-1=24$까지 각 차수에 대하여 맨 처음 적용하는 분할 방법과 필요한 게이트 수가 표 1과 같이 테이블에 저장 되어있다. 표 1에서 r 열은 각 차수에 대하여 효율적인 곱셈을 구성하기 위하여 n에 대하여 맨 처음 적용하는 방법으로 C는 SB 방법, 2는 MSK_2 방법, 3은 MSK_3 방법, 5는 MSK_5 방법을 의미한다. 차수 $n=25$에 대하여 각 방법을 적용할 경우 필요한 게이트 수와 시간 지연을 구하면 다음과 같다.

- $\#TOT_{SB}(25)=25^2+24^2=1201$,
- $T_{SB}(n)=1T_A+5T_X$,
- $\#TOT_{MSK_2}(26)=3\#TOT(13)+8\cdot 13-4$
  $=3\cdot 307+8\cdot 13-4=1021$,
- $T_{MSK_2}(26)=1T_A+9T_X$,
- $\#TOT_{MSK_3}(27)=6\#TOT(9)+22\cdot 9-10$
  $=6\cdot 134+22\cdot 9-10=992$,
- $T_{MSK_3}(27)=1T_A+10T_X$,
- $\#TOT_{MSK_5}(25)=14\#TOT(5)+66\cdot 5-28$
  $=14\cdot 41+66\cdot 5-28=876$,
- $T_{MSK_5}(25)=1T_A+8T_X$.

따라서 차수 n이 25일 경우에 MSK_5 방법을 적용하면 SB 방법, MSK_2, MSK_3 방법 보다 적은 게이트를 필요로 한다. □

보조정리 2에서 차수 $n=25$에 대하여 2:3 방법을 적용할 경우 최소의 게이트를 필요로 하는 분할 방법은 2만큼 0을 패딩한 27차에 대하여 MSK_3 방법을 맨 처음 적용하는 것으로 992개 의 게이트 수와 $1T_A+10T_X$의 시간 지연을 필요로 한다. 하지만 2:3:5 방법을 적용하면 $n=25$일 경우 테이블에 저장되는 분할 방법 및 게이트 수는 추가적인 패딩없이 차수 $n=25$에 대하여 MSK_5 방법을 적용하여 곱셈을 구성하는 것으로 876개의 게이트와 $1T_A+8T_X$의 시간 지연을 필요로 한다. 따라서 차수가 25일 경우에 곱셈을 구성하기 위하여 2:3 방법보다 2:3:5 방법을 적용하는 것이 효율적이다. 보조정리 1과 보조정리 2에 의하여 다음과 같은 정리를 만족한다.

정리1. $l\ge 0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하는 차수에 대하여 2:3:5 방법을 적용하여 곱셈을 구성하면 2:3 방법 보다 더 효과적이다.

증명) $l=0$이면 차수 $n=25$이므로 보조정리 2에 의하여 정리가 성립한다. $l>0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하는 차수일 경우 곱셈을 구성하기 위하여 2:3:5 방법과 2:3 방법을 적용 할 경우 보조정리 1과 같이 MKS_2 방법을 l 번 반복 적용하고 하위 25차 곱셈을 통하여 구성하는 것은 동일하다. 하지만 25차에 대한 곱셈은 보조정리 2와 같이 2:3:5 방법의 MSK_5 방법을 적용할 경우 2:3 방법 보다 적은 게이트를 필요로 한다. 따라서 $l\ge 0$, $25\cdot 2^l-2^l<n\le 25\cdot 2^l$을 만족하는 차수 n에 대하여 MSK_2 방법을 l번 반복하여 적용하고 차수 25에 대해서 MSK_5 방법을 수행하는 2:3:5 방법이 2:3 방법보다 곱셈을 구성하기 위하여 적은 게이트 수를 필요로 한다. □

2:3:5 방법과 2:3 방법을 적용하여 곱셈을 수행할

표 3. $25 \cdot 2^l$ 형태의 차수 n에 대한 2:3 방법을 적용할 경우 필요한 게이트 수와 2:3:5 방법을 적용할 경우 필요한 게이트 수와 그 차이

Table 3. Comparison of the number of gates used Gather's hybrid method [3] and propose hybrid method.

degree(n)		25	50	100	200	400	800
2:3 방법	$\#TOT_{2:3}(n)$	992	3172	9912	30532	93192	282772
	$T_{2:3}(n)$	T_A+10T_X	T_A+13T_X	T_A+16T_X	T_A+19T_X	T_A+22T_X	T_A+25T_X
.2:3:5 방법	$\#TOT_{2:3:5}(n)$	876	2824	8868	27400	83796	254584
	$T_{2:3:5}(n)$	T_A+8T_X	T_A+11T_X	T_A+14T_X	T_A+17T_X	T_A+20T_X	T_A+23T_X
$\#TOT_{2:3}(n)-\#TOT_{2:3:5}(n)$		116	348	1044	3132	9396	28188

경우 2:3:5 방법을 적용할 때 효율적인 구성이 가능한 차수에 대하여 알아보았다. $l=4$일 경우 2:3 방법과 2:3:5 방법을 적용하여 곱셈을 구성할 경우 필요한 게이트 수는 표 2와 같다.

표 2에서와 같이 $l \geq 0$, $25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$을 만족하는 차수에 대하여 2:3:5 방법이 2:3 방법보다 적은 게이트 수를 필요로 하고 시간 복잡도 또한 $2T_X$ 만큼 감소한다.

3. 효율성 분석

본 소절에서는 효율적인 차수에서 감소되는 게이트 비율에 대하여 알아보도록 한다.

차수가 범위 $l \geq 0$, $25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$ 을 만족하는 n일 경우 2:3 방법과 2:3:5 방법을 적용하여 곱셈을 수행하기 위하여 필요한 게이트 수와 시간 지연은 다음 정리를 만족한다.

정리2. 차수가 $l \geq 0$, $25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$을 만족하는 n에 대하여 2:3:5 방법을 적용하여 곱셈을 구성하면 2:3 방법 적용한 것 보다 게이트 수는 $116 \cdot 3^l$, 시간 지연은 $2T_X$ 감소한다.

증명) 2:3:5 방법과 2:3 방법을 적용하여 곱셈을 구성할 경우 주어진 차수에 대하여 보조정리 1과 같이 MSK_2 방법을 맨 처음 적용한다. 따라서 MSK_2 방법을 한 번 적용하여 곱셈을 구성할 경우 필요한 게이트 수와 시간 지연은

$$\#TOT_{2:3}(n) = 3\#TOT_{2:3}(\lceil n/2 \rceil) + 8\lceil n/2 \rceil - 4,$$

$$T_{2:3}(n) = T_{2:3}(\lceil n/2 \rceil) + 3T_X,$$

$$\#TOT_{2:3:5}(n) = \#TOT_{2:3:5}(\lceil n/2 \rceil) + 8\lceil n/2 \rceil - 4,$$

$$T_{2:3:5}(n) = T_{2:3:5}(\lceil n/2 \rceil) + 3T_X,$$

가 된다. 두 혼합 방법은 MSK_2 방법을 l 번 반복하고 하위 25차 곱셈을 필요로 한다. 따라서 차수 n에 대하여 MSK_2 방법을 l 번 반복 적용할 경우 곱셈을 수행하기 위하여 필요한 게이트 수와 시간 지연은 다음과 같다

$$\#TOT_{2:3}(n) = 3^l\#TOT_{2:3}(25) + 8 \cdot 25(3^l - 2^l) - 2(3^l - 1),$$

$$T_{2:3}(n) = T_{2:3}(25) + 3 \cdot lT_X,$$

$$\#TOT_{2:3:5}(n) = 3^l\#TOT_{2:3:5}(25) + 8 \cdot 25(3^l - 2^l) - 2(3^l - 1),$$

$$T_{2:3:5}(n) = T_{2:3:5}(25) + 3 \cdot lT_X.$$

따라서 2:3 방법을 적용할 경우 필요한 게이트 수에서 2:3:5 방법을 적용할 경우 게이트 수의 차이는

$$\begin{aligned} &\#TOT_{2:3}(n) - \#TOT_{2:3:5}(n) \\ &= 3^l(\#TOT_{2:3}(25) - \#TOT_{2:3:5}(25)) \\ &= 2^l(992 - 876) = 116 \cdot 3^l \end{aligned}$$

가 되고, 시간 지연 차이는

$$\begin{aligned} &T_{2:3}(n) - T_{2:3:5}(n) \\ &= T_{2:3}(25) - T_{2:3:5}(25) \\ &= T_A + 10T_X - (T_A + 8T_X) = 2T_X \end{aligned}$$

가 된다. □

정리 2에 의하여 $l \geq 0$, $25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$을 만족하는 각 차수 n에 대하여 2:3 방법과 2:3:5 방법을 적용하여 곱셈을 구성 할 경우 필요한 게이트 수는 $116 \cdot 3^l$ 만큼 감소한다. 혼합 방법에 따른 게이트 수의 감소량을 비교하기 위하여 2:3 방법을 적용할 경우 가장 많은 게이트를 필요로 하는 차수에 대한 게이트 수를 구하고 그에 따른 감소율을 구하도록 한다. 2:3 방법을 적용하여 곱셈을 구성할 경우 차수가 $n=25 \cdot 2^l$일 때 가장 많은 게이트 수를 필요로 하고, 이 때 필요한 게이트 수는

$$\#\mathrm{TOT}_{2:3}(25\cdot 2^l) = 3^l\#\mathrm{TOT}_{2:3}(25)+8\cdot 25(3^l-2^l)-2(3^l-1)$$

이 된다. 차수가 $l \geq 0$이고, $25\cdot 2^l$ 형태 일 때 차수에 따라 2:3 방법과 2:3:5 방법을 적용할 경우 필요한 게이트 수와 시간 지연은 표 3과 같다.

표 3과 같이 정수 $l \geq 0$과 $25\cdot 2^l - 2^l < n \leq 25\cdot 2^l$을 만족하는 차수에 대하여 2:3:5 방법을 적용하여 곱셈을 수행하면 2:3 방법을 적용하여 곱셈을 적용하는 것보다 약 10%의 게이트가 감소하고 시간 지연은 $2T_X$ 만큼 감소한다.

4. [1]에서 제안한 MSK_7 방법에 대한 고려

2:3:5 방법에 [1]에서 제안한 MSK_7 방법을 추가한 혼합 방법에 대하여 논한다. [1]에서 제안한 MSK_7 방법을 적용할 경우, MSK_7 방법에 가장 최적화 된 차수는 7의 거듭 제곱 형태이다. 하지만 차수가 7^1이면 SB 방법을 적용하는 것이 최적의 곱셈 방법이 되고, 차수가 7^2일 경우 1만큼 0을 패딩하여 MSK_5 방법을 적용하는 것이 MSK_7 방법을 적용하는 것 보다 적은 게이트를 필요로 하게 된다. 따라서 MSK_7 방법에 최적인 7의 거듭 제곱 형태의 차수에 대하여 MSK_7 방법보다 다른 방법을 적용하여 곱셈을 구성하는 것이 효율 적이므로 MSK_7 방법은 혼합 방법에 추가하여 구성할 필요가 없을 것으로 추정 된다.

IV. 결 론

본 논문에서는 유한체 $GF(2^n)$에서 효율적인 곱셈을 구성하기 위한 방법을 서술하기 위하여 일반적인 SB 방법과 Karatsuba-Ofman이 제안한 MSK_2 방법, 분할의 수를 확장하여 [1]에서 제안한 MSK_5, MSK_7 방법, [3]에서 제안한 SB, MSK_2, MSK_3 방법을 혼합한 방법을 적용하여 곱셈을 수행하기 위한 필요한 게이트 수를 제시하였다. 그리고 [3]에서 제안한 혼합 방법에 포함하지 않은 [1]에서 제안한 MSK_5 방법을 추가하여 곱셈을 수행하는 방법을 제안하였다. 제안하는 혼합 방법을 적용하여 곱셈을 구성하면 차수가 $l \geq 0$에 대하여 $25\cdot 2^l - 2^l < n \leq 25\cdot 2^l$을 만족하는 n일 경우 [3]에서 제안한 혼합 방법보다 $116\cdot 3^l$ 만큼의 게이트와 $2T_X$ 만큼의 지연시간이 감소되었다.

참 고 문 헌

[1] 장남수, 김창한, "유한체 $GF(2^n)$에서 낮은 공간 복잡도를 가지는 새로운 다중 분할 카라슈바 방법의 병렬 처리 곱셈기," 전자공학회 논문지, 제41권 SC편, 제1호, 33-40쪽, 2004년 1월

[2] M. Ernst, M. Jung, F. Madlener, S. Huss, R. Blümel, "A Reconfigurable System on Chip Implementation for Elliptic Curve Cryptography over $GF(2^n)$," In Work shop on Cryptographic Hardware and Embedded Systems (CHES'02), LNCS 2523, pp. 381-399, 2002.

[3] J. von zur Gathen, J. Shokrollahi, "Efficient FPGA-Based Karatsuba Multipliers for Polynomials over F_2," Selected Areas in Cryptography (SAC 2005), LNCS 3897, pp. 359-369, 2006.

[4] A. Karatsuba, Y. Ofman. "Mltiplication of multdigit numbers on automata," Soviet Physics-Doklady 7 (1963) 595-596 transkated from Doklady Akademii Nauk SSSR, Vol. 145, No. 2, pp. 293-294, July, 1962.

[5] N. Koblitz, "Elliptic Curve Ctyptosystems," Mathmetics of Computation, vol. 48, pp. 203-209, 1987.

[6] M. Leone, "A New Low Complexity Parallel Multiplier for a Class of Finite Fields," in Workshop on Cryptographic Hardware and Embedded Systems (CHES'01), LNCS 2162, pp. 160-170, 2001.

[7] R. Lidl and H. Niederreiter, "Introduction to finite fields and its applications," Cambridge Univ. Press, 1994.

[8] A. J. Menezes, I.F. Blake, X. Gao, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian, "Applications of finitr fields," Kluwer Academic, 1993.

[9] V. Miller, "Use of Elliptic Curve Cryptosystems," Advances in Cryptology, CRYPTO'85, LNCS 218, Springer-Verlag, pp. 417-426, 1986.

저 자 소 개

지 성 연(학생회원)
2005년 2월 한신대학교 수학과 학사
2005년~현재 고려대학교 정보경영공학전문대학원 석사과정
<주관심분야 : 공개키 암호이론, 공개키 암호시스템 안전성 분석 및 고속구현, 스마트카드>

장 남 수(학생회원)
2002년 2월 서울시립대학교 수학과 학사
2005년 2월 고려대학교 정보경영공학전문대학원 석사 과정
2005년~현재 고려대학교 정보경영공학전문대학원 박사과정
<주관심분야 : 공개키암호 암호칩설계기술 부채널공격방법론>

김 창 한(정회원)
1985년 2월 고려대학교 수학과 학사
1987년 2월 고려대학교 수학과 석사
1992년 2월 고려대학교 수학과 박사
2002년 2월~현재 세명대학교 정보통신학부 부교수
<주관심분야 : 정수론, 공개키암호, 암호프로토콜>

임 종 인(정회원)
1980년 2월 고려대학교 수학과 학사
1982년 2월 고려대학교 수학과 석사
1986년 2월 고려대학교 수학과 박사
1999년 2월~현재 고려대학교 정보경영공학전문대학원 원장, CIST 센터장
<주관심분야 : 암호이론, 정보보호정책>