논문 2006-43SD-1-7

타입 k 가우시안 정규기저를 갖는 유한체의 병렬곱셈 연산기

(A Multiplier for Type k Gaussian Normal Basis)

김 창 한*, 김 소 선**, 장 남 수***

(Chang Han Kim, Sosun Kim, and Nam Su Chang)

요 약

유한체의 H/W 구현에는 정규기저를 사용하는 것이 효과적이며, 특히 타입 I의 최적 정규기저를 갖는 유한체의 H/W 구현이 가장 효율적이다. 이를 이용하기 위하여 타입 (m,k) 인 가우스 주기를 갖는 유한체 중에서 $GF(mk+1)^*=\langle 2\rangle$를 만족하는 유한체 $GF(2^m)$을 타입 I 최적 정규기저를 갖는 유한체인 $GF(2^{mk})$의 부분체인 것을 이용한 새로운 병렬곱셈 연산기를 제안하였으며, 이러한 곱셈기는 암호학적으로 널리 응용되는 타입 k=2, 4, 6등의 경우에 기존에 알려진 가장 효율적인 Reyhani-Masoleh 과 Hasan[1][2] 의 연산기와 같은 복잡도를 갖는 효과적인 연산기이다

Abstract

In H/W implementation for the finite field, the use of normal basis has several advantages, especially, the optimal normal basis is the most efficient to H/W implementation in $GF(2^m)$. In this paper, we propose a new, simpler, parallel multiplier over $GF(2^m)$ having a Gaussian normal basis of type k, which performs multiplication over $GF(2^m)$ in the extension field $GF(2^{mk})$ containing a type-I optimal normal basis. For k=2,4,6 the time and area complexity of the proposed multiplier is the same as tha of the best known Reyhani-Masoleh and Hasan multiplier[1][2]

Keywords : 유한체 연산, 병렬곱셈 연산기, 가우시안 정규기저, 최적 정규기저

I. 서 론

유한체는 암호학과 코딩이론 등에 응용되고 있으며, 특히 최근 들어 공개키 암호인 타원곡선암호(ECC), XTR, ElGamal 타입 암호등의 관련 응용 분야에 활발하게 사용되고 있는 관계로 유한체의 효율적인 연산 방법이 많은 관심의 대상이 되고 있다[3][4]. 유한체의 연산은 표현방법에 따라 달라지는데, 대표적으로 다항식 기저[5]~[7], 정규기저[8]~[10] 등을 이용한 것이고 또 Nonconvention 기저[11]를 이용한 것도 사용된다. 특히, H/W 구현에는 정규기저를 이용한 경우 제곱이 Cyclic Shift 에 의하여 이루어지는 등 많은 장점을 가지고 있다. 그 중에서도 기약 AOP(All One Polynomial)에 의해 생성되는 타입I 최적 정규기저를 갖는 유한체가 가장 효과적으로 구현된다[1][2][12]. 정규기저를 이용한 병렬곱셈 연산기는 Massey-Omura[12]에 의하여 제안된 이후, 같은 해에 C.C. Wang 등[10]에 의하여 VLSI 구현이 보고된 이후 많은 발전이 있었으며, 최근에 Reyhani-Masoleh 과 Hasan[1]에 의해 개선된 연산기가 2002 제안되었고, 2003년에는 AND 게이트 수를 줄이는 연산기를 제안하였다[2]. 이러한 연산기도 타입I 최적 정규기저의 경우 가장 효율적으로 구성되는 것을 알 수 있는바, 우리는 타입 (m, k)인 가우스 주기를 갖고, $GF(mk+1)^*=\langle 2\rangle$인 유한체 $GF(2^m)$은 $GF(2^{mk})$

* 정회원, 세명대학교 정보보호학과
(Dept. of Information Security Semyung, University)
** 정회원, 소프트포럼 (Softforum Co., LTD)
*** 학생회원 고려대학교 정보보호대학원
(Center for Information Security Technologies(CIST), Korea University)
※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.
접수일자 : 2005년3월25일 수정완료일 : 2005년12월16일

의 부분체 이고 $GF(2^{mk})$는 타입 I의 최적 정규기저를 갖는다는 것을 잘 알려져 있다[3]. 그리고 표준문서 IEEE P1363[13]에 의하면 m 이 8의 배수가 아니면 타입 (m,k) 의 가우스 주기를 갖는 k 가 항상 존재 한다는 것이 알려져 있다. 또한 표준문서 P1363[14], ANSI X9.63[15]의 자료에 수록된 2000이하의 홀수인 m 중에서 $GF(2^m)$이 타입 IV 인 가우스안 정규기저를 갖는 모든 m은 위의 조건($GF(mk+1)^* = <2>$)을 만족하고, 또한 타입 II, VI, X 의 경우도 이러한 조건을 많이 만족한다. 예를 들면 II인 경우, m=113, 173, 183, 189, 209, 221, 233, 245, 161, 273, 281, 293, 309, 329, 545, 561, 585, 593 등이고, IV인 경우 m=103, 151, 291, 311, 331, 355, 395, 447, 451, 503, 511, 551, 591 등, X 인 경우는 m=109, 145, 157, 237, 253, 285, 301, 337, 349, 357, 369, 501, 613등이다. 또, m 이 소수인 경우 4m+1이 소수이면 $GF(4m+1)^* = \langle 2 \rangle$ 인 것은 잘 알려진 사실이다[10]. 따라서 $GF(2^m)$의 연산기를 타입 I 확대체인 $GF(2^{mk})$의 연산기를 활용하여 구성하고자 한다. 기본 타입 I 연산기는 Reyhani-Masoleh 과 Hasan[1,2]의 연산기를 이용하였고, 타원곡선 암호를 비롯한 암호 응용에서는 m이 소수인 경우를 주로 사용하고 있어서 본 논문에서는 m이 홀수인 경우만 고려 하였다.

결론적으로 본 논문에서는 n=mk 이고, 타입 I 최적 정규기저를 갖는 유한체 $GF(2^n)$의 부분체가 되는 타입 (m,k)인 가우스 주기를 갖는 유한체 $GF(2^m)$을 확대체 $GF(2^n)$에서 병렬곱셈 연산기를 작동하여 $GF(2^m)$의 곱셈을 수행하는 구조를 갖는 새로운 병렬 곱셈 연산기를 제안하였으며, 공간 및 시간 복잡도는 타입 k=2, 4, 6인 경우 알려진 것 중에서 가장 효율적인 Reyhani-Masoleh 과 Hasan[1][2]의 연산기와 같은 복잡도를 갖는다.

II. 수학적 배경

1. 유한체의 정규기저를 이용한 표현과 곱셈

양의 정수 l 에 대하여 유한체 $GF(2)$위에서 $GF(2^l)$의 정규기저가 존재한다는 것은 잘 알려진 결과이다[7][10]. 즉, $\beta \in GF(2^l)$가 존재하여 $N = \beta, \beta^2, \ldots, \beta^{2^{l-1}}$ 이 $GF(2)$위에서 $GF(2^l)$의 기저일 때 N를 정규기저라 하고 β를 정규기저 생성자라 한다. 이 경우, $A \in GF(2^l)$ 에 대하여

$$A = \sum_{i=0}^{l-1} a_i \beta^{2^i}, \quad a_i \in GF(2)$$

로 표현되며, 간단히 $A = (a_0, a_1, \ldots, a_{l-1})$와 같이 좌표로도 표현한다. 또한 벡터(행렬)표현으로

$$A = \bar{a} \times \bar{\beta}^T = \bar{\beta} \times \bar{a}^T, \ \bar{a} = [a_0 \ a_1 \ \ldots \ a_{l-1}],$$
$$\bar{\beta} = [\beta \ \beta^2 \ \ldots \ \beta^{l-1}]$$

그리고 T 는 행렬의 치환(Transpose)를 나타낸다. 그리고 정규기저의 특징이자 장점은 A^2 이 Right Cyclic Shift(RCS)에 의하여 주어진다는 것이다. 즉,

$$A^2 = (a_{l-1} \ a_0, \ \ldots, \ a_{l-2}).$$

$A, B \in GF(2^l)$, $C = AB$ 라 하자. 그러면

$$C = \bar{a} \times \bar{\beta}^T)(\bar{\beta} \times \bar{b}^T) = \bar{a} M \bar{b},$$
$$M = \bar{\beta}^T \bar{\beta} = (\beta^{2^i + 2^j}), \ 0 \leq i, j \leq l-1.$$

$\beta^{2^i+2^j}$ 를 기저 N를 사용하여 곱의 행렬 M을 다시 표현하면 다음과 같이 주어진다.

$$M = M_0 \beta + M_1 \beta^2 + \ldots + M_{l-1} \beta^{2^{l-1}},$$
$$M_i \in Mat_{l \times l}(GF(2)). \qquad (1)$$

A^2 이 cyclic shift 인 것을 이용하면 $C = AB = (c_0, c_1, \ldots, c_{l-1})$의 값은 다음과 같이 얻어진다.

$$c_i = \bar{a} M_i \bar{b}^T = \bar{a}^{(i)} M_0 \bar{b}^{(i)T},$$
$$\bar{a}^{(i)} = [a_i, \ a_{i+1}, \ldots, a_{i-1}],$$
$$\bar{b}^{(i)} = [b_i, \ b_{i+1}, \ldots, b_{i-1}]$$

이 같은 결과에 의하여 각 i 에 대하여 행렬 M_i 의 1의 개수는 모두 같음을 알 수 있고 이때 M_0 의 1의 개수를 정규기저 B 의 복잡도라 하고 C_N 으로 표시한다. 또한 Gao등은 다음과 같은 결과를 증명하였다[4][10].

정리 1. $C_N \geq 2l-1$[4].

이 논문에서는 모든 유한체는 정규기저에 의하여 표현되는 것으로만 고려한다.

2. 가우시안 정규기저

m, k 는 양의 정수, p = mk +1 ≠ 2 인 소수, 그리고 e 를 $GF(p)^*$ 에서 2의 위수(order), (mk/e, m) = 1 라 하자. 그리고 $GF(2^{mk})$에서 p의 원시근(a primitive nth root of unity)을 γ, kth root of unity를 τ 라 하고 $\beta = \gamma + \gamma^{\tau} + \gamma^{\tau^2} + \ldots + \gamma^{\tau^{k-1}}$ 로 놓으면 β는 $GF(2^m)$의 정규기저 생성자 이다[4][10]. 즉, $\{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{m-1}}\}$는 GF(2) 위에서 $GF(2^m)$의 정규기저이다. 이때 β를 GF(2) 위에서 타입 (m, k) 인 가우스 주기(Gauss period of type (m, k)) 라 한다[4][14][15]. 우리는 이 논문에서 $GF(2^m)$이 타입 (m,k)인 가우스 주기를 갖는 경우, $GF(2^m)$를 타입 k 라 한다.

정리에서 $C_N = 2m-1$일때 정규기저 N을 최적정규기저(Optimal Normal Basis, ONB) 라한다. 가우시안 정규기저에서 k= 1, 2인 경우 $C_N = 2m-1$을 만족하는 것은 잘 알려져 있다[4][8]. 이때 k=1인 경우를 타입 I 최적 정규기저, k=2인 경우를 타입 II의 최적 정규기저라 한다. 모든 계수가 1인 다항식을 All-One-Polynomial(AOP) $x^n + x^{n-1} + \ldots + x + 1$ 이라 한다.

정리 2. (타입 I 최적 정규기저)

$GF(2)$위에서 $GF(2^n)$ 이 타입 I 의 최적 정규기저를 갖기위한 필요충분 조건은 n+1 이 소수이고 $GF(n+1)^* = <2>$이다. 또는 n 차의 AOP $x^n + x^{n-1} + \ldots + x + 1$ 가 $GF(2)$위에서 기약다항식인 경우 AOP의 근이 최적 정규기저의 생성자 이다[4][8].

정리 3. (타입 II의 최적 정규기저)

2m+1은 소수이고, $GF(2m+1)^* = <2>$ 이거나 $2m+1 \equiv 3 \bmod 4$이고 $GF(2m+1)^* = <-1, 2>$ 이면 $\beta = \gamma + \gamma^{-1}$는 GF(2) 위에서 $GF(2^m)$ 의 최적 정규기저 생성자 이다. 여기서 γ는 2m+1의 원시근이다[4][8].

3. 부분체와 확대체의 관계

n=mk 인 경우 유한체 $GF(2^m)$은 $GF(2^n)$ 의 부분체이다. 이 경우 $A \in GF(2^m)$ 가 $GF(2^n)$에서 어떻게 표현되는지 살펴보자. 유한체의 기본 성질에 의하면 $B \in GF(2^n)$ 인 경우, $B \in GF(2^m)$ 이기위한 필요충분조건은 $B^{2^m} = B$ 이다. 따라서 정규기저를 사용하여 표현할 경우 제곱이 Right Cyclic Shift(RCS) 인 것을 이용하면 다음과 같은 결과를 얻 을 수 있다. 앞으로 $A \in GF(2^n)$를 좌표로 표현하면 $A = (a_0, a_1, \ldots, a_{n-1}),\ a_i \in GF(2)$ 와 같다.

정리 4. $B = (b_0, b_1, \ldots, b_{n-1}) \in GF(2^n)$라 하자. 그러면 $B \in GF(2^m)$ 이기위한 필요충분조건은 $0 \leq i, t < mk$ 에 대하여 $i \equiv t \bmod m$ 이면 $b_i = b_t$ 이다. 즉,

$$(b_0, b_1, \ldots, b_{m-1}, \cdots, b_0, b_1, \ldots, b_{m-1}).$$

앞으로 이 논문에서는 m 은 홀수, n = mk, n+1 은 소수 이고 $GF(n+1)^* = <2>$ 인 경우 만 고려하자. 이때 γ를 n+1 의 원시근이라 하면 정리2 에 의하여 γ는 $GF(2^n)$의 타입 I 의 최적정규 기저 생성자이다. 그리고 이 경우 e = mk, $\tau = 2^m$ 이므로 (m, k)는 가우스 주기이고 $\beta = \gamma + \gamma^{2^m} + \gamma^{2^{2m}} + \ldots + \gamma^{2^{m(k-1)}}$는 GF(2) 위에서 $GF(2^m)$ 의 정규기저 생성자 이다. 그러므로 원소 A ∈ $GF(2^m)$는

$$A = A_0\beta + A_1\beta^2 + A_2\beta^{2^2} + \ldots + A_{m-1}\beta^{2^{m-1}},$$
$$A_i, \in GF(2)$$

와 같이 표현된다. 그리고 $GF(2^m)$는 $GF(2^n)$의 부분체 이므로

$$\begin{aligned} A = {} & A_0\gamma + A_1\gamma^2 + A_2\gamma^{2^2} + \ldots + A_{m-1}\gamma^{2^{m-1}} \\ & + A_0\gamma^{2^m} + A_1\gamma^{2^{m+1}} + \ldots + A_{m-1}\gamma^{2^{2m-1}} \\ & + \ldots + \\ & + A_0\gamma^{2^{m(k-1)}} + A_1\gamma^{2^{m(k-1)+1}} + \ldots + A_{m-1}\gamma^{2^{mk-1}} \\ & \in GF(2^n) \end{aligned}$$

이다. 그리고

$$\begin{aligned} A = {} & a_0\gamma + a_1\gamma^2 + a_2\gamma^{2^2} + \ldots + a_{mk-1}\gamma^{2^{mk-1}} \\ & \in GF(2^n) \end{aligned}$$

라 하면

$a_{i+mj} = A_i,\ 0 \leq i \leq m-1,\ 0 \leq j \leq k-1$ (2)

이다.

Ⅲ. Reyhani-Masoleh and Hasan의 AOP를 이용한 곱셈기

Ⅲ장 에서는 Reyhani-Masoleh 와 Hasan[1][2]의 연산기에서 AOP의 경우에 제한하여 적용한 유한체의 병렬 곱셈기의 구조를 살펴보고자 한다.

1. Reyhani-Masoleh and Hasan의 AOP를 이용한 곱셈기(RR_MO_AOP)

Ⅱ장에서와 같이 $GF(2^m)$에서의 곱 $C = AB$를 계산하는 경우를 고려하자. 2002년에 Reyhani-Masoleh 와 Hasan은 유한체의 곱의행렬 $M = (\beta^{2^i+2^j})$을

$$M = U + U^T + D,$$

$$,D = \begin{pmatrix} \beta^2 & 0 & \cdots & 0 & 0 \\ 0 & \beta^{2^2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta^{2^{l-1}} & 0 \\ 0 & 0 & \cdots & 0 & \beta \end{pmatrix},$$

$$U = \begin{pmatrix} 0 & \beta^{1+2^1} & \cdots & \beta^{1+2^{l-2}} & \beta^{1+2^{l-1}} \\ 0 & 0 & \cdots & \beta^{2+2^{l-2}} & \beta^{2+2^{l-1}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \beta^{2^{l-2}+2^{l-1}} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

로 분리하고 $\delta_i = \beta^{1+2^i},\ i = 1, 2, \cdots,\ v = l/2$ 로 놓으면 U의 모든 원소는 $\delta_i^{2^j}, j = 0, 1, \cdots, l-1$ 로 표현되는 것을 이용하여 다음과 같은 정리를 제시하였고, 이 정리를 이용한 병렬 곱셈기[1][Fig.1]를 제안하였다.

정리 5. $A,\ B \in GF(2^l)$이고 $C = AB$라 하자. 그러면

$$C = \begin{cases} \sum_{j=0}^{l-1} a_j \odot b_j \beta^{2^{j+1}} + \sum_{i=1}^{v} \sum_{j=0}^{l-1} x_{j,i} \delta_i^{2^j}, for\ l\ odd \\ \sum_{j=0}^{l-1} a_j \odot b_j \beta^{2^{j+1}} + \sum_{i=1}^{v-1} \sum_{j=0}^{l-1} x_{j,i} \delta_i^{2^j} \\ \qquad + \sum_{j=0}^{v-1} x_{j,i} \delta_v^{2^j},\ for\ l\ even. \end{cases}$$

여기서 a_j, b_j는 A, B 의 좌표,

$x_{j,i} = (a_j \odot b_{i+j}) \oplus (b_j \odot a_{i+j})$, $\oplus$는 XOR, $\odot$ 는 AND 연산자이고 모든 색인은 모듈러 l 값이다.

특별히 이 정리를 $GF(2^n)$이 타입 I 의 최적 정규기저인 경우에 적용한 병렬곱셈기를 살펴보자. 즉, $GF(2^n)$은 기약다항식 AOP $x^n + x^{n-1} + \cdots + x + 1$에 의하여 생성된 유한체이고 AOP 의 근을 γ라 하면 γ 는 $GF(2^n)$의 타입 I 의 최적 정규기저를 생성한다. 이 경우 n 은 짝수이고 정리 5의 $\beta = \gamma$ 이므로 $\delta_i = \gamma^{1+2^i},\ i = 1, 2, \cdots, v = n/2$ 이다. 그리고 γ 가 AOP 의 근인 성질을 이용하여 다음과 같은 Lemma 를 얻을 수 있으며, 이 Lemma를 적용하여 그림1과 같은 타입 I 의 최적 정규기저를 이용한 Reyhani-Masoleh 와 Hasan의 연산기를 볼 수 있다.

Lemma 1.

$$\delta_i = \begin{cases} \gamma^{2^{k_i}}, & i = 1, 2, \cdots, n/2 - 1 \\ 1 = \sum_{j=0}^{v-1} \gamma^{2^j}, & i = v = n/2 \end{cases},$$

여기서 k_i는 $2^i + 1 \equiv 2^{k_i} \ mod\ n+1$ 을 만족하는 값이다.

구체적인 것을 보기 위하여 A, B를 $GF(2^n)$의 원소라 하자. 그러면

$$A = a_0\gamma + a_1\gamma^2 + a_2\gamma^{2^2} + \ldots + a_{n-1}\gamma^{2^{n-1}},$$

$$B = b_0\gamma + b_1\gamma^2 + b_2\gamma^{2^2} + \ldots + b_{n-1}\gamma^{2^{n-1}}$$

와 같이 표현되고

$$C = AB = c_0\gamma + c_1\gamma^2 + c_2\gamma^{2^2} + \ldots + c_{n-1}\gamma^{2^{n-1}}$$

라 하면 다음과 같은 Lemma 2 와 RR_MO[1][Fig.3]의 곱셈기가 얻어진다는 것을 알 수 있다. 이 논문에서 $\ll i+j \gg$ 는 $i+j\ mod\ n$ 값이다.

Lemma 2. $GF(2^n)$은 타입 I의 최적 정규기저를 갖는 유한체이고 γ 를 이 정규기저의 생성자라 하자. 그리고 $A, B \in GF(2^n),\ C = AB$ 라 하면

$$C = \sum_{j=0}^{n-1} a_j \odot b_j \gamma^{2^{\ll j+1 \gg}} + \sum_{i=1}^{v-1} (\sum_{j=0}^{n-1} x_{j,i} \gamma^{2^j})^{2^k} + \sum_{j=0}^{n-1} (\sum_{i=1}^{v-1} x_{i,v}) \gamma^{2^j},\ v = n/2,$$

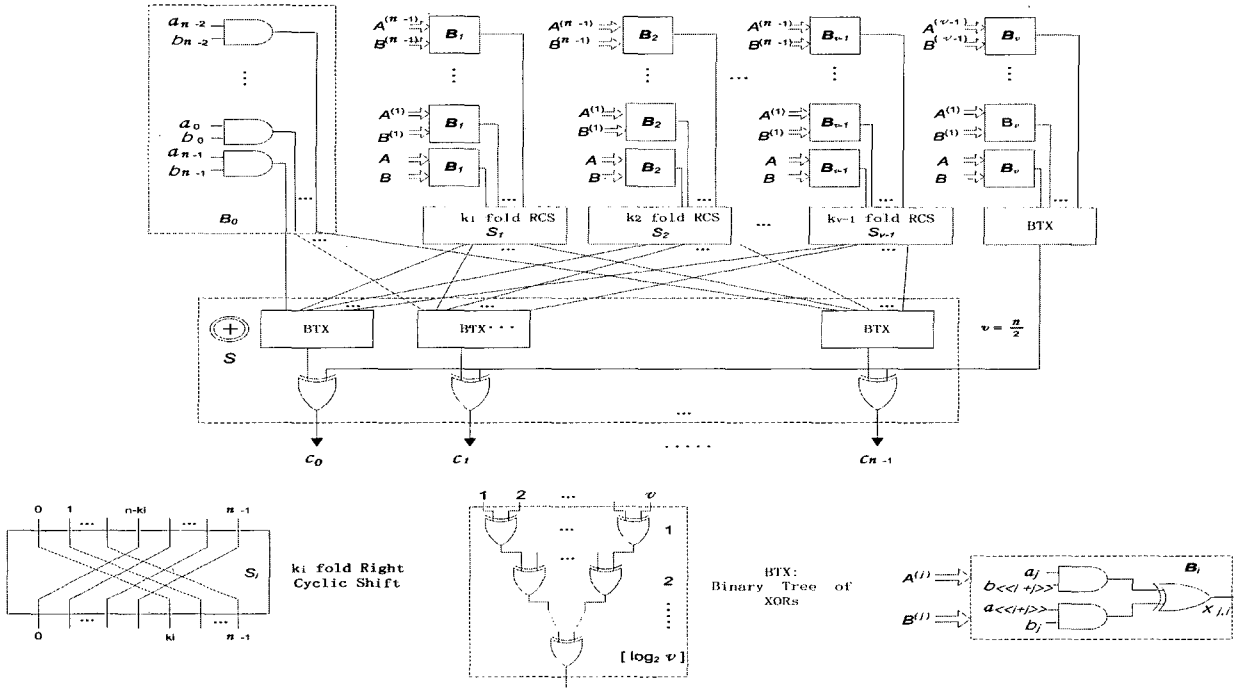

그림 1. RR-MO-AOP 연산기
Fig. 1. RR-MO-AOP Multipliers.

$$x_{j,i} = (a_j \odot b_{\ll i+j \gg}) \oplus (a_{\ll i+j \gg} \odot b_j).$$

2. Reyhani-Masoleh and Hasan의 AND 연산을 줄인 곱셈기(LCONB-I)

한편 2003년에는 Reyhani-Masoleh 와 Hasan[2]은 곱셈행렬 M을 $0 \leq i \leq v = l/2$ 에 대하여 $\delta_0^{2^j}, \delta_1^{2^j}, \cdots, \delta_v^{2^j}$, $j = 0, 1, \cdots, m-1$, $\delta_i = \beta^{1+2^i}$ 에 관한 행렬로 구분하여 표현되는 것을 이용하여 다음과 같은 정리를 바탕으로 하여 곱(AND)의 연산수를 줄이는 병렬곱셈 알고리즘를 제안하였다[2][알고리즘1].

정리 6. $A, B \in GF(2^m)$ 라 하고 $C = AB$ 라 하자. 그러면

$$C = \begin{cases} \sum_{j=0}^{l-1} a_j \odot b_j \delta_0^{2^{j-1}} + \sum_{j=0}^{l-1} \sum_{i=1}^{v} y_{j,i} \delta_i^{2^j}, & for\ l\ odd \\ \sum_{j=0}^{l-1} a_j \odot b_j \delta_0^{2^{j-1}} + \sum_{j=0}^{l-1} \sum_{i=1}^{v-1} y_{j,i} \delta_i^{2^j} + \sum_{j=0}^{v-1} y_{j,i} \delta_u^{2^j}, & for\ m\ even, \end{cases}$$

$$y_{j,i} = (a_j \oplus a_{i+j}) \odot (b_j \oplus b_{i+j}), \quad 1 \leq i \leq v, \quad 0 \leq j \leq m-1$$

이고 모든 색인은 모듈러 l 값이다[2].

Reyhani-Masoleh 와 Hasan은 이 정리를 이용한 알고리즘[2][알고리즘1]을 타입 I 의 최적 정규기저에 적용하여 알고리즘1을 제시하였다. 즉, 3.1에서와 같이 $GF(2^n)$ 은 AOP에 의하여 생성된 유한체이다.

알고리즘 1 (Low Complexity ONB-I Multiplication over $GF(2^n)$)[2]

Input: $A, B \in GF(2^n)$, $k_i, 1 \leq i < v$, $v = n/2$

Output: C = AB

1. Generate $y_{j,i} = (a_j \oplus a_{\ll i+j \gg}) \odot (b_j \oplus b_{\ll i+j \gg})$, $1 \leq i < v$, $0 \leq j \leq n-1$.
2. generate $y_{j,v} = (a_j \oplus a_{\ll v+j \gg}) \odot (b_j \oplus b_{\ll v+j \gg})$, $0 \leq j \leq v-1$.
3. Initialize $C_j = a_j b_j$, $0 \leq j \leq n-1$, $f := y_{0,v}$ $f \in GF(2)$.
4. For i =1 to v-1 {
5. $r_j = y_{j,i}$, $0 \leq j \leq m-1$, $R = (r_0, r_1, \cdots, r_{m-1})$.
6. R : = R^{2^k} .
7. C : = C + R .
8. f : = f + $y_{i,v}$.

9. }
10. If f is 1, C : = C + (1, 1, ... , 1).
11. }

이 알고리즘에서 k_i 는 Lemma 1 에 있는 k_i 이다.

IV. 타입 (m,k)의 가우시안 정규기저를 갖는 유한체의 연산기

Ⅲ장에서 언급한 바와 같이 $GF(2^m)$은 타입 k 인 가우시안 정규 기저를 갖고 $GF(n)^* = < 2 >$, n=mk 인 경우를 고려한다. 여기서 m이 홀수인 경우만 고려하므로 k는 짝수이다. 우리의 생각은 $GF(2^m)$의 원소 A, B 를 확대체인 타입 I 의 최적 정규기저를 갖는 $GF(2^n)$의 부분체의 원소로 생각하여 Ⅲ장의 연산기에 적용하여 A, B 의 곱셈 연산기를 구현 하고자 한다. 먼저 ς_i 를 정의 하자.

정의 1. n=mk, n+1 은 소수, $GF(n+1)^* = \langle 2 \rangle$라 하고 k_i 는 Lemma 1 의 값이라 하자.

이때, $1 \leqq i_0 \leqq u = (m-1)/2$에 대해 $i \in \{i_0,\ m-i_0,\ m+i_0,\ \ldots, km/2-i_0\}$ 인 경우 다음과 같이 ς_i를 정의 한다.

$$\varsigma_i = \begin{cases} k_i \mod m, & i \equiv i_0 \mod m \\ k_i + i_0 \mod m, & i \equiv -i_0 \mod m. \end{cases}$$

1. 2002년 Reyhani-Masoleh and Hasan의 곱셈기에 적용

A, B $\in GF(2^m) \subset GF(2^n)$ 인 경우를 살펴하자. 그러면 Ⅲ.1의 연산기에서,

먼저, $a_i \odot b_i = A_{((i))} \odot B_{((i))}$, $0 \leqq i \leqq n-1$ 이므로 n 개를 계산하여야 하나 (2) 식에 의하여 m 개인 $A_i \odot B_i, 0 \leqq i \leqq m-1$만 계산하면 된다.

두 번째로 $A^{(j)}, B^{(j)}$ 가 B_i에 입력되어 $x_{j,\,i} = (a_j \odot b_{\ll i+j \gg}) \oplus (a_{\ll i+j \gg} \odot b_j)$로 출력되는 값을 계산하여 보자. 2.3 의 식(2)에 의하여 $a_{\ll i \gg} = a_{((i))} = A_{((i))}$인 것을 이용하자.

1) i = 1 일 때 즉,

$$x_{j,\,1} = (a_j \odot b_{\ll 1+j \gg}) \oplus (a_{\ll 1+j \gg} \oplus b_j),$$

$0 \leqq j \leqq n-1$를 보자. Ⅱ.3의 (2)의 관계식을 이용하면

$$x_{0,\,1} = (a_0 \odot b_1) \oplus (a_1 \odot b_0) = (A_0 \odot B_1) \oplus (A_1 \odot B_0)$$
$$x_{1,\,1} = (a_1 \odot b_2) \oplus (a_2 \odot b_1) = (A_1 \odot B_2) \oplus (A_2 \odot B_1)$$
$$\vdots$$
$$x_{m-1,\,1} = (a_{m-1} \odot b_{((m))}) \oplus (a_{((m))} \odot b_1)$$
$$= (A_{m-1} \odot B_0) \oplus (A_0 \odot B_{m-1})$$
$$x_{m,\,1} = (a_m \odot b_{((m+1))}) \oplus (a_{((m+1))} \odot b_m)$$
$$= (A_0 \odot B_1) \oplus (A_1 \odot B_0) = x_{0,\,1}$$

일반적으로

$$x_{tm+j,\,1} = (a_{tm+j} \odot b_{((tm+j+1))}) \oplus (a_{((tm+j+1))} \odot b_{tm+j})$$
$$= (A_j \odot B_{j+1}) \oplus (A_{j+1} \odot B_j) = x_{j,\,1}$$
$$0 \leqq t \leqq k-1, 0 \leqq j \leqq m-1 \ .$$

2) i = 2, ... , u=(m-1)/2 일 때 까지 비슷한 방법으로 $0 \leqq j \leqq -1$ 인 경우는

$$x_{j\,i} = (a_j \odot b_{\ll j+i \gg}) \oplus (a_{\ll j+i \gg} \odot b_j)$$
$$= (A_j \odot B_{((j+i))} \oplus (A_{((j+i))} \odot B_j),$$
$$tm+j,\ o \leqq t \leqq k-1, 0 \leqq j \leqq m-1$$

인 경우는

$$x_{tm+j,\,i} = (a_{tm+j} \odot b_{\ll tm+j+i \gg}) \oplus (a_{\ll tm+j+i \gg} \odot b_{tm+j})$$
$$= (A_j \odot B_{((j+1))}) \oplus (A_{((j+1))} \odot B_j)$$
$$= x_{j,\,i}$$

이다.

3) i = u+1, ... m-1 일 때 i=u+s ,
$1 \leqq s \leqq u = (m-1)/2$ 로 나타내면
$o \leqq t \leqq k-1, 0 \leqq j \leqq m-1$ 인 경우에

$$x_{tm+j,\,i} = x_{tm+j,\,u+s}$$
$$= (a_{tm+j} \odot b_{((tm+j+u+s))}) \oplus (a_{((tm+j+u+s))} \odot b_{tm+j})$$
$$= (A_j \odot B_{((j+u+s))}) \oplus (A_{((j+u+s))} \odot B_j)$$
$$= (A_{((j+u+s))} \odot B_j) \oplus (A_j \odot B_{((j+u+s))})$$

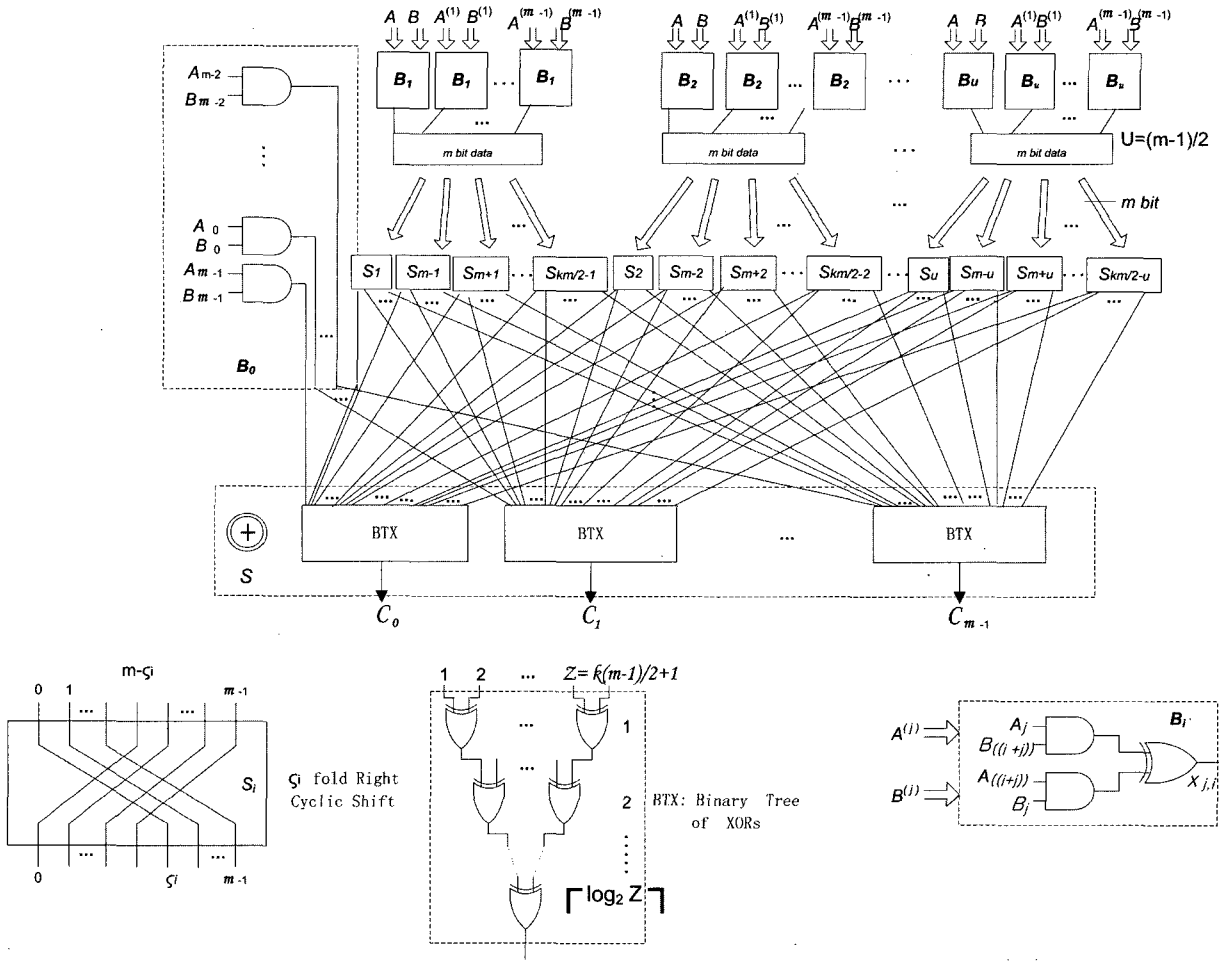

그림 2. AOP를 이용한 연산기
Fig. 2. Multipliers using AOP.

$= x_{j+u+s,\, u+1-s} = x_{((j+i)),\, m-i}$,

여기서 $1 \leqq u+1-s \leqq u$ 이므로 $x_{tm+j,\, i}$ 는 1), 2)의 경우에서 $B_{u+1-s} = B_{m-i}$, $1 \leqq m-i \leqq u$ 에서 계산된다.

4) $i = wm, 1 \leqq w \leqq k/2$ 인 경우

$$x_{j,\, wm} = (a_j \odot b_{\ll j+wm \gg}) \oplus (a_{\ll j+wm \gg} \odot b_j)$$
$$= (A_{((j))} \odot B_{((j))}) \oplus (A_{((j))} \odot B_{((j))}) = 0$$

이다.

5) i = m+1, ..., (k/2)m -1 이고 $i \neq wm$ 일 때,
$i = wm + r,\ 1 \leqq w \leqq k/2-1,$
$1 \leqq r \leqq m-1$ 라 하면

$$x_{j,\, i} = (a_j \odot b_{\ll wm+r+j \gg})$$
$$\oplus (a_{\ll wm+r+j \gg} \odot b_j)$$
$$= (A_{((j))} \odot B_{((r+j))}) \oplus (A_{((r+j))} \odot B_{((j))})$$
$$= x_{((j)),\, r} \ .$$

위 계산에서 $1 \leqq i_0 \leqq u = (m-1)/2$ 에 대하여 B_{i_0} 로부터 B_{m-i_0}, B_{m+i_0}, ..., $B_{(k/2-1)m-i_0}$, $B_{(k/2-1)m+i_0}$, $B_{(k/2)m-i_0}$은 ewiring에 의하여 얻어진다. 즉, $1 \leqq i_0 \leqq u$ 에 대하여 S_{i_0}, S_{m+i_0}, ..., $S_{(k/2-1)+i_0}$ 의 입력은 B_{i_0} 에 의하여 구해지는 대로 비트에 차례로 입력된다. 그러나 S_{m-i_0}, S_{2m-i_0}, ..., $S_{(k/2)m-i_0}$ 의 경우는, $1 \leqq w \leqq k/2$라 하면

$$x_{j,\, wm-i_0} = x_{j,\, m-i_0}$$

$$= (a_j \odot b_{\ll j+m-i_0 \gg}) \oplus (a_{\ll j+m-i_0 \gg} \odot b_j)$$
$$= (a_{\ll j+m-i_0 \gg} \odot b_j) \oplus (a_j \odot b_{\ll j+m-i_0 \gg})$$
$$= x_{j+m-i_0, i_0}$$
$$= x_{((j-i_0)), i_o},$$
$$i_0 = m-u-s = 2u+1-u-s$$

이므로 i_0 쉬프팅에 의하여 입력된다.

그리고 $B_m, B_{2m}, \ldots, B_{(k/2)m}$ 의 모든 값은 0 이므로 계산할 필요가 없다. 그리고 III.1의 Reyhani-Masoleh and Hasan의 곱셈기에서 $1 \leqq i < v = n/2$ 에 대하여 B_i 통과한 후 S_i 에서의 연산은 Lemma 1을 만족하는 k_i에 의한 RCS 에 의하여 수행된다. 한편 $A, B \in GF(2^m) \subset GF(2^n)$ 이므로 II.3 에서와 같이 m 개의 좌표가 k 번 반복하여 표현되며 $AB \in GF(2^m)$이므로 0부터 m-1 까지의 좌표만 고려하면 된다. 그러므로 $0 \leqq i_0 \leqq u$ 에 대해 B_i, $i \in \{i_0, m-i_0, m+i_0, \ldots, km/2-i_0\}$ 인 경우 B_{i_0} 로부터 S_i 의 출력값은 앞에 정의한 ς_i에 의한 RCS 에 의하여 주어진다. 이것을 요약하면 정리 7을 얻을 수 있다.

정리 7. $GF(2^m)$은 타입(m, k)인 가우스 주기를 갖고, n = mk, $GF(n+1)^* = \langle 2 \rangle$라 하자.
$A, B \in GF(2^m) \subset GF(2^n)$, $C = AB$ 이면

$$C = \sum_{j=0}^{m-1} A_j B_j \beta^{2^j} + \sum_{i_0=1}^{u} \{ \sum_{w=0}^{k/2-1} (\sum_{j=0}^{m-1} x_{j,i_0} \beta^{2^j})^{2^{wm+i_0}} + \sum_{w=1}^{k/2} (\sum_{j=0}^{m-1} x_{j,i_0} \beta^{2^j})^{2^{wm-i_0}} \},$$

$$x_{j,i_0} = (A_j \odot B_{((i_0+j))}) \oplus (A_{((i_0+j))} \odot B_j),$$
$$1 \leqq i_o \leqq u = (m-1)/2,\ 0 \leqq j < + m-1.$$

그리고 곱셈기의 구조는 그림 2와 같다.

2. 2003년 Reyhani-Masoleh and Hasan의 곱셈기에 적용

III. 2의 정리 6을 타입 I 의 최적 정규기저를 갖는 유한체에 적용한 알고리즘1을 바탕으로 Lemma 3을 표시할수 있다.

Lemma 3. $GF(2^n)$은 타입 I의 최적 정규기저를 갖는 유한체이고 γ 를 이 정규기저의 생성자라 하자. 그리고 $A, B \in GF(2^n)$, $C = AB$ 라 하면

$$C = \sum_{j=0}^{n-1} a_j \odot b_j \gamma^{2^j} + \sum_{i=1}^{v-1} (\sum_{j=0}^{n-1} y_{j,i} \gamma^{2^j})^{2^k} + \sum_{j=0}^{m-1} (\sum_{i=1}^{v-1} y_{v,i}) \gamma^{2^j},\ v = n/2,$$
$$y_{j,i} = (a_j \oplus a_{\ll i+j \gg}) \odot (b_j \oplus b_{\ll i+j \gg}).$$

여기서 k_i는 Lemma 1의 값이다.

4.1에서와 같이 A, B $\in GF(2^m) \subset GF(2^n)$ 의 원소로 생각하자. 그러면 3.2의 연산기 또는 Lemma 3 에서, 먼저, $a_i \odot b_i = A_{((i))} \odot B_{((i))}$, $0 \leqq i \leqq n-1$ 를 계산하여야 하나 (2) 식에 의하여 $A_i \odot B_i, 0 \leqq i \leqq m-1$만 계산하면 된다. 즉, m 개의 AND 연산으로 주어진다.

다음으로 $y_{j,i} = (a_j \oplus a_{\ll i+j \gg}) \odot (b_j \oplus b_{\ll i+j \gg})$, $1 \leqq i \leqq v = n/2$, $0 \leqq j \leqq n-1$를 계산하자. 4.1 에서와 같이 (2)식이 성립하므로

$$A_i = a_{i+mj},\ 0 \leqq i \leqq m-1,\ 0 \leqq j \leqq k-1$$
$$B_i = b_{i+mj},\ 0 \leqq i \leqq m-1,\ 0 \leqq j \leqq k-1$$

이 성립한다. 먼저

$$i = i_0 + tm,\ 1 \leqq j_0 \leqq m-1,\ 0 \leqq w \leqq k/2-1$$

과

$$j = j_0 + tm,\ 0 \leqq j_0 \leqq m-1,\ 0 \leqq t \leqq k-1$$

일 때

$$y_{j,i} = (a_j \oplus a_{\ll i+j \gg}) \odot (b_j \oplus b_{\ll i+j \gg})$$
$$= (a_{j_0+tm} \oplus a_{\ll i_0+wm+j_0+tm \gg}) \odot (b_{j_0+tm} \oplus b_{\ll i_0+wm+j_0+tm \gg})$$
$$= (a_{j_0} \oplus a_{((i_0+j_0))}) \odot (b_{j_0} \oplus b_{((i_0+j_0))})$$
$$= (A_{j_0} \oplus A_{((i_0+j_0))}) \odot (B_{j_0} \oplus B_{((i_0+j_0))})$$
$$= y_{j_0,i_0}$$

특별히 i = u + s, u = $m/2$, $1 \leqq s \leqq u$인 경우 $i_0 = m-i$ 라 하면 $1 \leqq i_0 \leqq u$ 이다.

그리고

$$y_{j,i} = (a_j \oplus a_{\ll j+u+s \gg}) \odot (b_j \oplus b_{\ll j+u+s \gg})$$
$$= (a_{\ll j+u+s \gg} \oplus a_j) \odot (b_{\ll j+u+s \gg} \oplus b_j)$$
$$= (A_{((j+u+s))} \oplus A_{((j))}) \odot (B_{((j+u+s))} \oplus B_{((j))})$$
$$= y_{((j+u+s)),((m-(u+s)))} = y_{((j+i)),i_0} \quad (3)$$

다음으로 i = m, 2m , ... , v = mk/2 인 경우

$$y_{j,i} = (a_j \oplus a_{\ll tm+j \gg}) \odot (b_j \oplus b_{\ll tm+j \gg})$$
$$= (A_{((j))} \oplus A_{((j))}) \odot (B_{((j))} \oplus B_{((j))}) = 0.$$

그러므로 알고리즘 1의 1, 2 단계에서 전체적으로 $y_{j,i}$ 는 $0 \leqq j \leqq m-1,$, $1 \leqq i \leqq u = (m-1)/2$ 인 경우만 계산하면 되고 알고리즘1의 8번 스텝에서 f 의 계산과 4번 루프에서 i = m, 2m, ..., m(k/2-1)의 계산은 할 필요가 없음을 알 수 있다. 그리고 $GF(2^m)$ 은 $GF(2^n)$ 의 부분체이므로 부분체의 성질에 의하여 $GF(2^n)$에서 m 개의 좌표를 이용하여 표현할 수 있으므로 알고리즘 1의 수행을 n 개의 좌표가 아닌 m 개의 좌표로 수행하면 된다. 그리고 알고리즘1의 스텝4의 For 문중 6의 계산 R^{2^k} 에 관하여 살펴보자. $0 \leqq i_0 \leqq u$ 에 대해 $i \in \{i_0,\ m-i_0,\ m+i_0,\ \ldots, km/2-i_0\}$ 인 경우 R 은 y_{j,i_0} 에 의해 구성된다. 즉, $R_{i_0} = (y_{0,i_0}\ y_{1,i_0} \cdots y_{m-1,i_0})$라 하면 (3)번 식에 의하여 $R_{wm+i_0} = R_{i_0}$, $0 \leqq w \leqq k/2-1$ 이고 $R_{wm-i_0} = R_{i_0}^{2^{i_0}}$, $1 \leqq w \leqq k/2$. 그리고 각 i 에 대하여 Lemma 1에서 얻어진 k_i 만큼의 RCS를 수행한다. 그러므로 i 에 대한 6번 과정은 R_{i_0} 를 앞에서 정의한 ς_i 만큼의 RCS 에 의하여 구해진다. 이 같은 결과를 종합하여 다음의 정리 8를 얻을 수 있다.

정리 8. $GF(2^m)$은 타입 k 이고, n = mk, $GF(n+1)^* = \langle 2 \rangle$라 하자.

$A, B \in GF(2^m) \subset GF(2^n)$, $C = AB$ 이면

$$C = \sum_{j=0}^{m-1} A_j B_j \beta^{2^j} + \sum_{i_0=1}^{u} \{ \sum_{w=0}^{k/2-1} (\sum_{j=0}^{m-1} y_{j,i_0} \beta^{2^j})^{2^{wm+i_0}}$$
$$+ \sum_{w=1}^{k/2} (\sum_{j=0}^{m-1} y_{j,i_0} \beta^{2^j})^{2^{wm-i_0}} \},$$
$$y_{j,i_0} = (A_j \oplus A_{((i_0+j))}) \odot (B_{((i_0+j))} \oplus B_j),$$

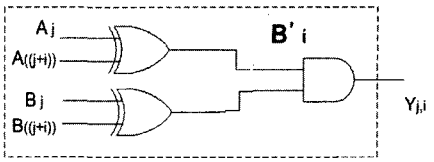

그림 3. $y_{j,i}$ 연산 블록
Fig. 3. The Operation Block of $y_{j,i}$.

$$1 \leqq i_0 \leqq u,\ 0 \leqq j \leqq m-1.$$

이 정리 8을 이용한 병렬곱셈 연산기는 2번 연산기에서 $\mathbf{B}_i$ 의 출력이 $y_{j,i}$가 되도록 입력을 A_j, $A_{((j+i))}$, B_j, $B_{((j+i))}$ 로 바꾸고 회로를 그림 3 과 같이 구성하고, 그리고 $A_j \odot B_j$값이 BTX의 j 번째 좌표의 입력값이 되도록 조정하면 연산기가 구성된다.

3. 실제구현

실제적 구현시 $1 \leqq i_0 \leqq u = (m-1)/2$ 에 대하여

$$i = i_0, m-i_0,\ m+i_0, \ldots, km/2-i_0 \quad (4)$$

인 경우 먼저, IV.1에서 위의 i 에 대하여 ς_i 중 2개의 같은 경우 S_i 의 출력값이 같으므로 그 2개를 삭제하여도 결과가 같다. 또한 IV.2의 경우 위의 i 에 대하여 ς_i 중 2개의 같은 경우 6번 스텝의 결과가 같으므로 그 경우 삭제하여도 결과가 같다는 것이다. 이 같은 사항을 고려하여 구현할 경우 XOR gate 수와 Time delay를 줄일 수 있다. 즉, 위의 각 i_0 에 대하여 모든 i 에 대한 ς_i 가 같은 것이 있는지 확인 한다. 다음의 $GF(2^m)$ 은 k 타입의 가우시안 정규기저를 갖고 $GF(km+1)^* = \langle 2 \rangle$ 를 만족한다.

1) k =4 인 경우를 살펴보자.

m 이 2000이하의 경우에 대하여 다음의 Conjeture 1 이 성립하는 것을 확인 하였다. 그러므로 이같은 결과에 의하여 XOR gate의 수를 2m 개를 줄일 수 있다. 타입 4인 권장 타원곡선인 m=163인 경우, i 가 163+81, 2*163-81일 때 ς_i = 81로 같다. 그리고 m=409인 경우,

i 가 409+204, 2*409-204일 때 ς_i = 204로 같다.

Conjecture 1. (k=4인 경우) m이 홀수, p=4m+1 은 소수, $GF(p)^* = \langle 2 \rangle$ 라 하자.
그리고 u = (m-1)/2 에 대하여, $GF(p)^* = \langle 2 \rangle$에서

$$\begin{cases} 2^u + 1 = 2^{k_1}, & 2^{m-u} + 1 = 2^{k_2}, \\ 2^{m+u} + 1 = 2^{k_3}, & 2^{2m-u} + 1 = 2^{k_4} \end{cases}$$

라 하면 다음 두식 중 하나가 성립한다.

$$k_1 = k_2 + u \ mod \ m \ , \quad k_3 = k_4 + u \ mod \ m \ .$$

그러므로 타입 4의 경우는 2m 개의 XOR gate 수를 줄일수 있다.

2) k=6인 경우를 살펴보자.

타입 k=6을 만족하는 1000 이하에 m 대하여 (4)식의 i 에 대한 ς_i 가 같은 쌍이 4개 나타난다. 즉, m=103인 경우, i 가 2*103 -1, 2*103+1 일 때 ς_i = 89이고, 14, 103-14 일 때 14, 103+14, 2*103-14 일 때 15, 15, 103-15 일 때 14이다. 이것을 이용하여 다음의 Conjecture 2를 얻을 수 있다.

Conjecture 2. (k=6인 경우) m이 홀수, p=6m+1 은 소수, $GF(p)^* = \langle 2 \rangle$ 라 하자.

$1 \leq i_0 \leq u = (m-1)/2$에 대하여,

$i = i_0, m - i_0, \ m + i_0, \ldots, km/2 - i_0$ 일 때

ς_i 가 같은 쌍이 4개 존재 한다.

그러므로 타입 6의 경우는 XOR gate를 8m 개를 제하여도 되는 것을 알 수 있다.

3) k=10 인 경우 m를 보자.

타입 k=10을 만족하는 1000 이하에 m 대하여 (4)식의 i 에 대한 ς_i가 같은 쌍이 16개 나타난다. 먼저 i_0 에 대하여 (4)식의 i_1, i_2에 대한 ς_i 가 같을 때를 $\{i_0, \ i_1, i_2, \varsigma_i\}$로 표현하자. m=337인 경우,

{6, m-6, 2*m+6, 278}, {7, 7, m+7, 116}, {7, m-7, 2*m +7, 0 }, {51, m+51, 2*m+51, 51}, {51, 2*m-51, 5*m-51, 272}, {52, m+52, 2*m +51, 109}, {57, m-57, 5*m-57, 285}, {58, m-58, 5*m-58, 0}, {58, 2*m-58, 4*m-58, 58}, {59, 2*m-59, 4*m-59, 65}, {65, 65, 5*m-65, 59}, {65, 2*m+65, 3*m+65, 116}, {109, m+109, 4*m-109, 52}, {109, 2*m+109, 5*m-109, 330}, {116, 116, 2*m-116, 65}, {116, 2*m+116, 4*m-116, 7} 이다.

Remark 1. (k=10, 12인 경우) m이 홀수, p=10m+1 또는 12m+1이 소수, $GF(p)^* = \langle 2 \rangle$ 인 경우를 살펴보며.

$1 \leq i_0 \leq u = (m-1)/2$에 대하여,

$i = i_0, m - i_0, \ m + i_0, \ldots, km/2 - i_0$ 일 때

ς_i 가 같은 쌍이 k=10이면 16 개, k=12 이면 25개 존재 한다. 따라서 타입 10의 경우 32m 개, 12의 경우 50 개의 XOR gate 수를 줄일 수 있다.

V. 타입 (m,k)의 가우시안 정규기저를 갖는 유한체의 연산기의 복잡도

IV.1과 IV.2의 연산기의 복잡도를 살펴보면 다음과 같은 정리를 얻을 수 있다.

정리 9. IV.1의 연산기의 복잡도의 최대값은 다음과 같다.

a) m^2 AND gate.

b) $(k+1)m(m-1)/2$ XOR gate.

c) $T_A + log_2\{k(m-1)+2\} \ T_X$, T_A 는 AND delay, T_X 는 XOR Delay 이다.

증명. a) 먼저, $0 \leq i \leq m-1$에 대하여 $A_i \odot B_i$ 의 계산에 m번의 AND 연산이 필요하다. 그리고 $1 \leq i \leq u = (m-1)/2$ 에 대하여 B_i를 계산하는데 2* m 개의 AND 연산이 필요하므로 전체는 m + 2m* u = m^2개가 필요하다.

b) 먼저 $1 \leq i \leq u$ 에 대하여 B_i의 계산에서 각 m 개가 필요하므로 전체 B_i 계산에는 $m(m-1)/2$ 개의 XOR 연산이 필요하다. 그리고 각 BTX 에는 $k(m-1)/2$ 개의 S로부터 1 비트씩 그리고 $A_i \odot B_i$ 로부터 1비트 가 주어져서 연산을 하여야 하므로 각 $k(m-1)/2$개의 XOR 이 필요하고 전체 BTX 에서는 $km(m-1)/2$개의 연산이 필요하다. 그러므로 전체 XOR 게이트는 $(k+1)m(m-1)/2$ 개 이다.

c) Time delay는 AND 의 경우는 1번이고 XOR 의 경우는 B_i 계산에 1번 BTX 계산에 $\lceil \log_2(k(m-1)/2+1) \rceil$ 이므로

전체는 $\lceil \log_2(k(m-1)+2) \rceil$ 이다.

Corollary 1. 타입 II 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) m^2 AND gate, $\frac{3}{2}m(m-1)$ XOR gate

b) $T_A + (1+ \log_2 m)T_X$.

Corollary 2. 타입 4 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) m^2 AND gate, $m(5m-9)/2$ XOR gate.

b) $T_A + log_2(4m-6) T_X$

$= T_A + \lceil \log_2 C_N \rceil T_X$

증명. 실제 구현할 경우 IV.3에서 보면 2m 개의 XOR 가 줄게 됨으로 XOR 개수를 구할 수 있고 4m-6이 짝수 이므로 $log_2(4m-6) = \log_2 C_N$.

Corollary 3. 타입 6 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) m^2 AND gate, $m(7m-23)/2$ XOR gate.

b) $T_A + log_2(6m-20) T_X$

$= T_A + \lceil \log_2 C_N \rceil T_X$

증명. 실제 구현할 경우 IV.3에서 보면 8m 개의 XOR 가 줄게됨으로 얻어짐.

한편으로 IV.2의 곱셈기의 복잡도를 살펴보면 다음과 같다.

정리 10. IV.2의 연산기의 복잡도의 최대값은 다음과 같다.

a) $m(m+1)/2$ AND gate.

b) $(k+2)m(m-1)/2$ XOR gate.

c) $T_A + log_2\{k(m-1)+2\} T_X$, T_A 는 AND delay, T_X 는 XOR Delay 이다.

증명. a) 먼저, $A_i \odot B_i, 0 \leq i \leq m-1$의 계산에 m 번, 그리고 $y_{j,i}$, $0 \leq j \leq m-1$, $1 \leq i \leq u = (m-1)/2$ 계산에 m(m-1)/2 번이므로 전체적으로 $m(m+1)/2$ 번의 AND 연산이 필요하다.

b) $y_{j,i}$, $0 \leq j \leq m-1$,

$1 \leq i \leq u = (m-1)/2$ 계산에 m(m-1) 번, $R_i^{2^{i_0}}$, $0 \leq i_0 \leq u$

$i = i_0,\ m-i_0,\ m+i_0,\ \ldots, km/2-i_0$ 에 대한

표 1. 정규기저에 의한 유한체의 병렬 곱셈 연산기의 복잡도 비교
Table 1. Comparison of Normal Basis Multipliers.

Multipliers		#AND	#XOR	Time Delay
MO [14]		mC_N	$m(C_N-1)$	$T_A + \lceil \log_2 C_N \rceil T_X$
RR_MO [11]	General	m^2	$m(C_N+m-2)/2$	$T_A + \lceil \log_2 C_N \rceil T_X$
	Type IV	m^2	$m(5m-9)/2$	$T_A + \lceil \log_2(4m-7) \rceil T_X$
	Type VI	m^2	$m(7m-23)/2$	$T_A + \lceil \log_2(6m-21) \rceil T_X$
LCNB [12]	General	$m(m+1)/2$	$m(C_N+2m-3)/2$	$T_A + \lceil \log_2 C_N \rceil T_X$
	Type IV	$m(m+1)/2$	$m(3m-5)$	$T_A + \lceil \log_2(4m-7) \rceil T_X$
	Type VI	$m(m+1)/2$	$4m(m-3)$	$T_A + \lceil \log_2(6m-21) \rceil T_X$
4.1	General	m^2	$(k+1)m(m-1)/2$	$\leq T_A + \lceil \log_2(k(m-1)+2) \rceil T_X$
	Type IV	m^2	$m(5m-9)/2$	$T_A + \lceil \log_2(4m-7) \rceil T_X$
	Type VI	m^2	$m(7m-23)/2$	$T_A + \lceil \log_2(6m-21) \rceil T_X$
4.2	General	$m(m+1)/2$	$\leq (k+2)m(m-1)/2$	$\leq T_A + \lceil \log_2(k(m-1)+2) \rceil T_X$
	Type IV	$m(m+1)/2$	$m(3m-5)$	$T_A + \lceil \log_2(4m-7) \rceil T_X$
	Type VI	$m(m+1)/2$	$4m(m-3)$	$T_A + \lceil \log_2(6m-21) \rceil T_X$

XOR 이 $m\{\frac{k}{2}(m-1)-1\}$, 이 결과에 $A_i \odot B_i, 0 \leqq i \leqq m-1$의 XOR 이 m 개 추가 됨으로 전체는 $\frac{k+2}{2}m(m-1)$ 개 이다.

c) T_A는 1 이고, T_X 는 $y_{j,i}$ 계산에 1번, $R_i^{2^s}$ 과 $A_i \odot B_i$ 의 계산에 $log_2\{k(m-1)/2+1\}$ 이 필요함으로 결과적으로 $log_2\{k(m-1)+2\}$ 이다.

Corollary 4. IV.2 연산기에서 타입 II 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) $m(m+1)/2$ AND gate, $2m(m-1)$ XOR gate.

b) $T_A + \log_2 m \; T_X$.

Corollary 5. IV.2 연산기에서 타입 IV 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) $m(m+1)/2$ AND gate, $m(3m-5)$ XOR gate.

b) $T_A + \log_2 C_N \; T_X$.

증명. 실제 구현할 경우 IV.3에서 보면 2m 개의 XOR 가 줄게됨으로 얻어짐.

Corollary 6. 타입 6 의 최적 정규 기저를 갖는 유한체 $GF(2^m)$의 복잡도는 다음과 같다.

a) m^2 AND gate, $4m(m-3)$ XOR gate.

b) $T_A + \log_2 C_N \; T_X$.

증명. 실제 구현할 경우 4.3에서 보면 8m 개의 XOR 가 줄게됨으로 얻어짐.

Remark 2. 타입 (m, 10) 또는 (m, 12)인 가우스 주기를 갖는 유한체에서의 연산기는 Remark 1 에서와 같이 16, 25쌍의 ς_i가 각각 일치함으로 정리 10으로부터 32m, 50m 개의 XOR gate를 빼면 되고 그에 따른 T_X의 값도 줄 것이다.

전체적인 유한체의 병렬 곱셈 연산기와 기존 연산기와의 복잡도 비교는 표 1과 같다.

VI. 예 제

m =7, k = 4 이면 n = 28, p = 29 이므로 $GF(29)^* = \langle 2 \rangle$ 이다. Lemma 1을 만족하는 각 i

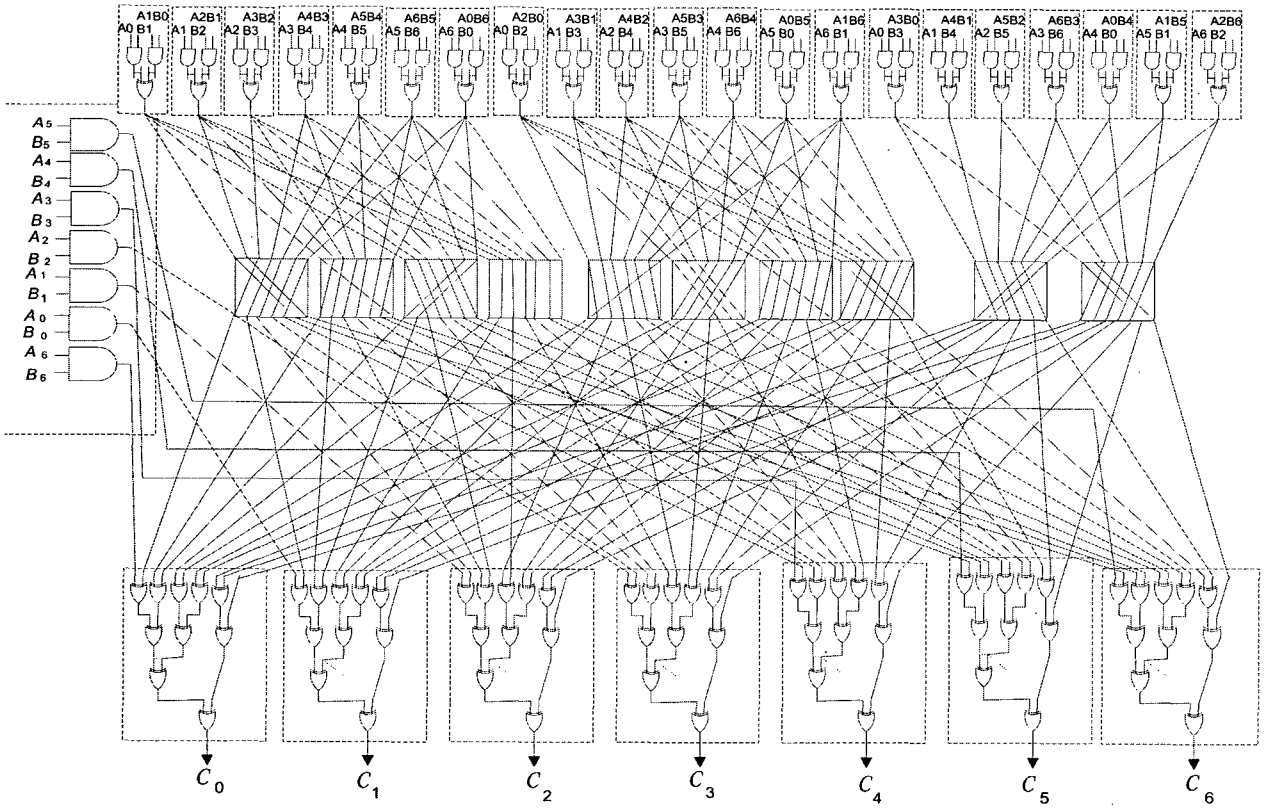

그림 4. 유한체 $GF(2^7)$의 병렬곱셈 연산기

Fig. 4. Parallel Multiplier for $GF(2^7)$.

에 대한 k_i 와 정의1 에서 구해지는 ς_i 의 값을 $\{i,\ k_i,\ \varsigma_i\}$와 같이 표현 하면 다음과 같다.

{{1,5,5}, {7−1,12,6}, {7+1,16,2}, {2*7−1,27,0}, {2,22,1}, {7−2,2,4}, {7,+2,24,3}, {2*7−2,3,5}, {3,10,3}, {7−3,21,3}, {7+3,23,2}, {2*7−3,9,5} }.

그러므로 $i = i_0 = 3$, $i = 7 - i_0 = 4$에서 $\varsigma_i = 3$이 같은 값 이므로 S_3, S_4 는 삭제하여도 된다.

Ⅶ. 결 론

유한체가 암호학적 분야에 응용되면서 유한체의 연산에 많은 관심을 가지고 있으며, 하드웨어 구현은 정규기저를 이용하여 표현할 경우 효율적으로 구현할 수 있다. 특히 타입 I의 최적정규기저를 갖는 유한체가 가장 효과적으로 구현된다. 그리고 양의 정수 m 이 8의 배수가 아니면 타입 (m, k)인 가우스 주기가 되는 k 가 존재한다는 것도 알려진 사실이다. 따라서 본 논문에서는 $GF(2^m)$이 타입 (m, k)인 가우스 주기를 갖고 $GF(mk+1)^* = \langle 2 \rangle$인 경우 $GF(2^{mk})$는 타입 I 의 최적 정규 기저를 갖는 $GF(2^{mk})$의 부분체인 것을 이용하여 $GF(2^m)$의 병렬곱셈 연산기를 타입 I 최적 정규기저를 갖는 $GF(2^{mk})$에 적용하였다. 그러한 결과 암호학적으로 많이 응용되는 유한체인 k= 2, 4, 6의 경우 지금까지 알려진 것 중 가장 효율적인 Reyhani-Masoleh 과 Hasan[1][2]의 연산기와 같은 공간 및 시간 복잡도를 갖는 새로운 연산기를 구성하였으며 따라서 암호관련 응용 분야의 H/W 구현에 유용하게 활용할 수 있을 것으로 기대된다.

참 고 문 헌

[1] A. Reyhani-Masolleh and M.H. Hasan, "A new construction of Massey-Omura parallel multiplier over $GF(2^m)$", *IEEE Trans.* vol.51, no.5, pp. 512-520, May, 2002.

[2] A. Reyhani-Masolleh and M.H. Hasan, "Efficient multiplication beyond optimal normal bases", *IEEE Trans.* vol.52, no.4, pp. 428-439, April, 2003.

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

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

[5] T. Itoh and S. Tsujii, "Structure of parallel multipliers for a class of fields $GF(2^m)$", *Information and Computation*, vol.83, pp. 21-40, 1989.

[6] C.K. Koc and B. Sunar, "Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields", *IEEE Trans.* vol.47, no.3, pp. 353-356, Mar, 1998.

[7] H. Wu and M.A. Hasan, "Low Complexity bit-parallel multipliers for a class of finite fields", *IEEE Trans.* vol.47, no.8, pp. 883-887, Aug., 1998.

[8] S. Gao Jr. and H.W. Lenstra, "Optimal normal bases", *Designs, Codes and Cryptography*, vol. 2, pp.315-323, 1992.

[9] B. Sunar and C.K. Koc, "An efficient optimal normal basis type II multiplier", *IEEE Trans.* vol.50, no.1, pp. 83-88, Jan., 2001.

[10] C.C Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, "VLSI architectures for computing multiplications and inverses in $GF(2^m)$", *IEEE Trans.* vol.34, no.8, pp. 709-716, Aug., 1985.

[11] C.H. Kim, S. Oh, and J. Lim, "A new hardware architecture for operations in $GF(2^n)$", *IEEE Trans.* vol.51, no.1, pp. 90-92, Jan, 2002.

[12] M.A. Hasan, M.Z. Wang, and V.K. Bhargava, "A modified Massey-Omura parallel multiplier for a class of finite fields", *IEEE Trans.* vol.42, no.10, pp. 1278-1280, Oct, 1993.

[13] J.L Massey and J.K. Omura, *Computational method and apparatus for finite field arithmetic*, US Patent No. 4,587,627, to OMNET Assoc., Sunnyvale CA, Washington, D.C.: Patent Trademark Office, 1986.

[14] IEEE P1363, *Standard specifications for public key cryptography*, Draft 13, 1999.

[15] ANSI X 9.63, *Public key cryptography for the financial sevices industry* : Elliptic curve key agreement and transport protocols, draft, 1998.

저 자 소 개

김 창 한 (정회원)
1985년 고려대학교 수학과 학사
1987년 고려대학교 수학과 석사
1992년 고려대학교 수학과 박사
<주관심분야 : 정보보호, 공개키 암호, 병렬연산>

김 소 선 (정회원)
2003년 이화여자대학교 수학과 학사
2005년 고려대학교 정보보호대학원 석사
2005년~현재 소프트포럼
<주관심분야 : 공개키 암호, 암호칩 설계 기술>

장 남 수 (학생회원)
2002년 서울시립대학교 수학과 학사
2005년 고려대학교 정보보호대학원 석사
2005년~현재 고려대학교 정보보호대학원 박사 과정
<주관심분야 : 공개키 암호, 암호칩 설계 기술, 부채널 공격>