가우시안 정규기저를 이용한 $GF(2^m)$상의 새로운 곱셈 알고리즘 및 VLSI 구조

정회원 권 순 학*, 김 창 훈**°, 김 희 철**, 홍 춘 표**

A New Multiplication Algorithm and VLSI Architecture Over $GF(2^m)$ Using Gaussian Normal Basis

Soonhak Kwon*, Chang Hoon Kim**°, Hiecheol Kim, Chun Pyo Hong** ***Regular Members***

요 약

유한체상의 곱셈은 타원곡선 암호시스템의 구현에 있어 가장 중요한 연산 중 하나이다. 본 논문에서는 가우시안 정규기저를 이용하여, $GF(2^m)$상의 새로운 곱셈 알고리즘 및 VLSI 구조를 제안한다. 제안된 곱셈 알고리즘은 정규기저 원소의 대칭성이용과 계수의 인덱스 변형에 기반 하며, 타원곡선 암호 시스템을 위해 NIST(National Institute of Standards and Technology) 및 IEEE 1363에서 권고하는 다섯 가지 $GF(2^m)$, $m\in\{163, 233, 283, 409, 571\}$, 모두에 적용 할 수 있다. 제안된 곱셈알고리즘에 기반한 VLSI 구조는 기존의 $GF(2^m)$상의 정규기저 곱셈기에 비해 속도 혹은 하드웨어 면적에 있어 향상된 성능을 보인다. 또한 본 논문에서는 정규기저 원소의 기본 곱셈 행렬을 쉽게 찾을 수 있는 방법을 제시한다.

Key Words : GNB, Finite Field, Elliptic Curve Cryptosystem, Multiplication, VLSI

ABSTRACT

Multiplications in finite fields are one of the most important arithmetic operations for implementations of elliptic curve cryptographic systems. In this paper, we propose a new multiplication algorithm and VLSI architecture over $GF(2^m)$ using Gaussian normal basis. The proposed algorithm is designed by using a symmetric property of normal elements multiplication and transforming coefficients of normal elements. The proposed multiplication algorithm is applicable to all the five recommended fields $GF(2^m)$ for elliptic curve cryptosystems by NIST and IEEE 1363, where $m\in\{163, 233, 283, 409, 571\}$. A new VLSI architecture based on the proposed multiplication algorithm is faster or requires less hardware resources compared with previously proposed normal basis multipliers over $GF(2^m)$. In addition, we gives an easy method finding a basic multiplication matrix of normal elements.

Ⅰ. 서 론

1980년대 중반 Victor Miller와 Neal Kobliz에 의해 제안된 타원곡선 암호 시스템(Elliptic Curve Cryptosystem: ECC)는 최근 학계나 산업계로부터 많은 관심을 모으고 있다[17]. ECC의 가장 주된 장점은 RSA나 ElGamal과 같은 다른 암호 시스템에 비해 현저히 작은 키를 사용하면서(약 1/6 정도) 동

※ 본 연구는 과학기술부 과학재단 목적기초연구(R01-2005-000-11261-0)지원으로 수행되었음.
* 성균관대학교 수학과 (shkwon@skku.edu)
** 대구대학교 정보통신공학과 (chkim@dsp.daegu.ac.kr, hckim@daegu.ac.kr, cphong@daegu.ac.kr) (° : 교신저자)
논문번호 : KICS2006-03-137, 접수일자 : 2006년 3월 21일, 최종논문접수일자 : 2006년 11월 23일

일한 안전도를 가진다[17]. 작은 키를 사용한다는 것은 계산 시간, 전력 소모 그리고 저장 공간의 감소를 의미한다. 이러한 장점 때문에 최근 IEEE 1363[18] 및 NIST[16]은 공개키 암호 시스템을 위해 ECC에 기반한 타원곡선 전자서명 알고리즘(Elliptic Curve Digital Signature: ECDSA)를 표준으로 채택하였다. ECDSA를 위해 유한체는 $GF(p)$와 $GF(2^m)$을, $GF(2^m)$상의 원소 표기법으로는 가우시안 정규기저(Gaussian Normal Basis: GNB)와 다항식 기저(Polynomial Basis: PB) 표기법을 사용한다. 여기서 p는 소수이고 GNB는 정규 기저(Normal Basis: NB)의 특별한 경우로서 8로 나누어지지 않는 모든 양의 정수 m에 대해 존재한다.

ECC를 $GF(2^m)$상에서 구현 할 경우, $GF(2^m)$상의 덧셈, 곱셈, 역원(혹은 나눗셈) 연산이 필요하다. 여기서 덧셈은 비트별 XOR 연산으로, 기저 표기법에 상관없이 동일하다. 그러나 곱셈 및 역원 연산은 선택된 기저에 따라 서로 다른 하드웨어 구조를 갖는다. PB를 사용한 곱셈기 설계의 경우 m값과 원소 생성에 사용되는 기약다항식(Irreducible Polynomial)에 상관없이 동일한 하드웨어 구조 설계가 가능한 장점이 있다. NB를 사용할 경우 임의의 원소 $A \in GF(2^m)$의 A^{2^i} $(0 \le i \le m-1)$ 연산은 i-비트 순환 쉬프트로 Fermat의 이론을 이용하면 연속된 $A \times A^{2^i}$연산으로 역원 연산을 수행 할 수 있다[18]. 따라서 NB를 사용하여 ECC를 구현 할 경우 곱셈기의 성능은 매우 중요하다.

지금까지 다양한 NB 곱셈기들이 제안되었다[1], [3-5],[7],[10],[11]. 이러한 곱셈기들 중 Messey와 Omura 곱셈기는[7] 패러럴 입력 시리얼 출력 구조를 가지지만 곱셈기의 최대 처리기 지연시간은 $\log_2 m$에 비례한다. 따라서 이 곱셈기는 ECC와 같이 매우 큰 m을 요구하는 응용에서는 높은 최대 처리기 지연시간을 보인다. Agnew등[1]은 Messey와 Omura 곱셈 알고리즘을 이용하여 패러럴 입력 패러럴 출력 구조의 선형 곱셈기를 제안하였다. 이 곱셈기는 Messey와 Omura 곱셈기에 비해 m-비트 레지스터를 추가함으로서 최대 처리기 지연시간을 $TA+2TX$로 줄였다. 여기서 TA는 2-입력 AND 게이트 딜레이 시간이고 TX는 2-입력 XOR 게이트 딜레이 시간이다. 또한 최근 Reyhani-Masoleh와 Hasan[3]은 정규기저 원소의 대칭성을 이용하여, $TA+2TX$의 최대 처리기 지연시간을 가지는 저면적 선형 곱셈기를 제안하였다.

본 논문에서는 GNB를 이용하여, $GF(2^m)$상의 새로운 곱셈 알고리즘을 개발하고 이를 바탕으로 효율적인 선형 곱셈기를 제안한다. 제안된 곱셈기는 정규기저 원소들의 대칭성을 이용할 뿐만 아니라 정규기저 원소 계수의 인덱스를 변형함으로서 Agnew 등이 제안한 곱셈기 보다 낮은 하드웨어 복잡도를 가지지만 동일한 최대 처리기 지연시간을 같고 Reyhani-Masoleh와 Hasan이 제안한 곱셈기와 동일한 하드웨어 복잡도를 가지만 낮은 최대 처리기 지연시간을 보인다. 또한, 본 논문에서는 정규기저 원소의 기본 곱셈 행렬을 찾는 쉬운 방법을 제시하며, 제안된 곱셈 알고리즘은 NIST[16]에서 권고하는 다섯 가지 $GF(2^m)$, $m \in \{163, 233, 283, 409, 571\}$, 모두에 적용 할 수 있다.

Ⅱ. 관련 연구

$GF(2^m)$을 표수 2인 유한체라 하면, $GF(2^m)$은 $GF(2)$상의 m차원 벡터 공간이다. 임의의 원소 α에 대해, $N=\{\alpha, \alpha^2, \alpha^{2^2}, \cdots, \alpha^{2^{m-1}}\}$ 형태의 기저를 $GF(2^m)$상의 정규 기저라 한다. 이 때 임의의 원소 α는 정규 기저의 생성원(generator)이다. $GF(2^m)$상의서 모든 $m \ge 1$에 대해 정규 기저가 존재함은 잘 알려져 있다[6]. $GF(2^m)$상의 정규 기저 $N=\{\alpha, \alpha^2, \alpha^{2^2}, \cdots, \alpha^{2^{m-1}}\}$에 대해, $\alpha_i = \alpha^{2^i}$라 놓으면, 우리는 $N=\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$와 같이 표현 할 수 있다.

아래 식 (1)과 같이

$$\alpha_i \alpha_j = \sum_{s=0}^{m-1} \lambda_{ij}^{(s)} \alpha_s \tag{1}$$

라 하면, $\lambda_{ij}^{(s)}$는 $GF(2)$의 원소이고 임의의 정수 t에 대해 아래 식 (2)는 성립한다.

$$\begin{aligned} \alpha_i \alpha_j = (\alpha_{i-t}\alpha_j)^{2^t} &= \sum_{s=0}^{m-1} \lambda_{i-t,j-t}^{(s)} \alpha_{s+t} \\ &= \sum_{s=0}^{m-1} \lambda_{i-t,j-t}^{(s-t)} \alpha_s \end{aligned} \tag{2}$$

식 (2) 이 때, λ의 위, 아래 첨자는 mod m에 기약된다. 따라서 α_s의 계수를 비교하면,

$$\lambda_{ij}^{(s)} = \lambda_{i-t,j-t}^{(s-t)} \tag{3}$$

임을 알 수 있고 아래의 식 (4) 역시 만족한다.

$$\lambda_{ij}^{(s)} = \lambda_{i-s,j-s}^{(0)} \tag{4}$$

$A=\sum_{i=0}^{m-1}a_i\alpha_i$, $B=\sum_{j=0}^{m-1}b_j\alpha_j$를 $GF(2^m)$상의 두 원소라 하면, 두 원소의 곱 $C=AB=\sum_{s=0}^{m-1}c_s\alpha_s$는 식 (5)와 같다.

$$C=\sum_{i,j}a_ib_j\alpha_i\alpha_j=\sum_{i,j}a_ib_j\sum_{s=0}^{m-1}\lambda_{ij}^{(s)}=\sum_{s=0}^{m-1}(\sum_{i,j}a_ib_j\lambda_{ij}^{(s)})\alpha_s \quad (5)$$

따라서, 식 (4)를 사용하여 C의 계수 C_s는 아래의 식 (6)을 이용하여 계산 할 수 있다.

$$c_s=\sum_{i,j}a_ib_j\lambda_{ij}^{(s)}=\sum_{i,j}a_ib_j\lambda_{i-s,j-s}^{(0)}=\sum_{i,j}a_{i+s}b_{j+s}\lambda_{ij}^{(0)} \quad (6)$$

이 때, a, b, λ의 아래 첨자는 mod m에 기약된다. Agnew등[1]은 $m\times m$ 곱셈 행렬 $\lambda_{ij}^{(0)}$와 함께 위의 식 (6)을 직접적으로 구현하였다. [1]에 기술된 바와 같이, 최적 정규기저(Optimal Normal Basis: ONB) Type 2가 존재할 경우, 우리는 $\lambda_{ij}^{(0)}$을 쉽게 찾을 수 있다. 즉,

$$\lambda_{ij}^{(0)}=1 \text{ iff } 2^i\pm2^j\equiv\pm1 \pmod{2m+1} \quad (7)$$

이다. 아래 그림 1은 Agnew등이 제안한 곱셈기의 구조를 나타며, $m=5$인 경우로서 ONB Type 2이다. ONB와 같이 특별한 경우를 제외한, 임의의 유한체에 대해, $\lambda_{ij}^{(0)}$을 찾는 것은 쉽지 않다. 그러나 다음절에서 설명하겠지만, GNB를 사용한다면 간단한 산술적 방법에 의해 $\lambda_{ij}^{(0)}$을 쉽게 찾을 수 있다.

최근 Reyhani-Masoleh와 Hasan[3]은 Agnew등의 곱셈기에 비해 하드웨어 복잡도를 줄인 새로운 정규 기저 곱셈기를 제안하였다. 이들은 $\alpha_i\alpha_j$ 대신 $\alpha\alpha_i$을 사용했고, $\alpha\alpha_i$와 $\alpha\alpha_{m-i}$의 대칭성을 이용하였다. 이들은 XESMPO와 AESMPO라 명한 두 가지의 선형 곱셈기 제안하였다. AESMPO의 하드웨어 복잡도는 XESMPO의 하드웨어 복잡도보다 높고, 최대 처리기 지연시간은 같기 때문에 XESMPO에 경우에 대해서만 다룬다. 두 원소의 곱 $C=A\cdot B$는

$$\sum_{i,j}a_ib_j\alpha_i\alpha_j=\sum_{i=0}^{m-1}a_ib_i\alpha_{i+1}+\sum_{i=0}^{m-1}\sum_{j\neq1}a_ib_j(\alpha\alpha_{j-i})^{2^i}=\sum_{i=0}^{m-1}a_ib_i\alpha_{i+1}+\sum_{i=0}^{m-1}\sum_{j\neq0}a_ib_{j+i}(\alpha\alpha_j)^{2^i} \quad (8)$$

와 같이 나타낼 수 있다[3],[4]. 만약 m이 홀수이면, 위 수식의 오른쪽 두 번째 부분은

$$\sum_{i=0}^{m-1}\sum_{j=1}^{\nu}a_ib_{j+i}(\alpha\alpha_j)^{2^i}+\sum_{i=0}^{m-1}\sum_{j=m-\nu}^{m-1}a_ib_{j+i}(\alpha\alpha_j)^{2^i} \quad (9)$$

로 표현되고, m이 짝수이면

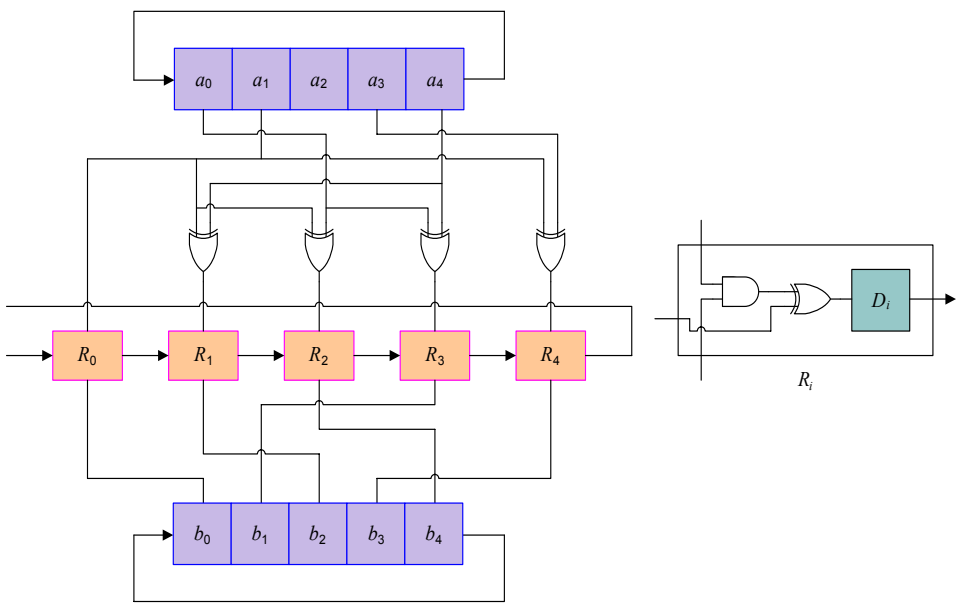

그림 1. $GF(2^5)$상의 Agnew등이 제안한 곱셈 회로도

$$\sum_{i=0}^{m-1}\sum_{j=1}^{\nu} a_i b_{j+i}(\alpha\alpha_j)^{2^i} + \sum_{i=0}^{m-1}\sum_{j=m-\nu}^{m-1} a_i b_{j+i}(\alpha\alpha_j)^{2^i} + \sum_{i=0}^{m-1} a_i b_{\nu+1+i}(\alpha\alpha_{\nu+1})^{2^i} \quad (10)$$

로 표현된다[4]. 이 때, $\nu = \lfloor \frac{m-1}{2} \rfloor$이다. 즉, $m = 2\nu+1$ 또는 $m = 2\nu+2$이다. 또한, 식 (9)와 (10)의 두 번째 부분은 아래의 식 (11)과 같이 나타 낼 수 있다.

$$\begin{aligned}\sum_{i=0}^{m-1}\sum_{j=m-\nu}^{m-1} a_i b_{j+i}(\alpha\alpha_j)^{2^i} &= \sum_{i=0}^{m-1}\sum_{j=1}^{\nu} a_i b_{n-j+i}(\alpha\alpha_{n-j})^{2^i} \\ &= \sum_{i=0}^{m-1}\sum_{j=1}^{\nu} a_{i+j} b_i(\alpha\alpha_{n-j})^{2^{i+j}} \\ &= \sum_{i=0}^{m-1}\sum_{j=1}^{\nu} a_{i+j} b_i(\alpha\alpha_j)^{2^i}\end{aligned} \quad (11)$$

첫 번째(두 번째) 등식은 $j(i)$에 대응되는 부분합의 재결합으로 얻어지고, 모든 아래 첨자는 mod m에 기약된다. 그러므로 Reyhani-Masoleh와 Hasan의 곱셈 방식을 m이 홀수인 경우와 짝수인 경우로 나누어 나타내면, 우리는 아래의 식 (12)와 (13)을 얻을 수 있다.

$$AB = \sum_{i=0}^{m-1} a_i b_i \alpha_{i+1} + \sum_{i=0}^{m-1}\sum_{j=1}^{\nu} (a_i b_{j+i} + a_{j+i} b_i)(\alpha\alpha_j)^{2^i} \quad (12)$$

또는

$$AB = \sum_{i=0}^{m-1} a_i b_i \alpha_{i+1} + \sum_{i=0}^{m-1}\sum_{j=1}^{\nu} (a_i b_{j+i} + a_{j+i} b_i)(\alpha\alpha_j)^{2^i} + \sum_{i=0}^{m-1} a_i b_{\nu+1+i}(\alpha\alpha_{\nu+1})^{2^i} \quad (13)$$

Reyhani-Masoleh와 Hasan은 위의 두 식을 이용하여 그림 1에 비해 게이트 복잡도가 줄어든 선형 곱셈기를 제안하였으며, 아래 그림 2와 같은 구조를 가진다. 그림 2 역시 $m=5$인 경우로서 ONB Type 2이다.

Ⅲ. $GF(2^m)$상의 GNB Type k

m, k를 소수 $p \neq 2$에 대해, $p = mk+1$인 양의 정수라 하고, $K = <\tau>$는 $GF(p)^\times$상에서 위수(order) k인 유일한 부분군이라 하자. β가 $GF(2^{mk})$상의 p번째 원시 근 이라하면, 우리는 아래 원소

$$\alpha = \sum_{j=0}^{k-1} \beta^{\tau^j} \quad (14)$$

을 $GF(2)$상의 Type (m, k) 가우스 주기(Gauss Period)라 한다. $\mathrm{ord}_p 2$을 mod p에 대한 2의 위수라 하고, $\gcd(mk/\mathrm{ord}_p 2,\ m) = 1$이라고 가정하면, α는 $GF(2^m)$상의 정규 원소이다. 즉, $0 \le i \le m-1$에 대해, $\alpha_i = \alpha^{2^i}$라 놓으면, $\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$은 $GF(2)$상의 $GF(2^m)$에 대한 기저이고, 우리는 이를 $GF(2^m)$상의 GNB Type k라 부른다. $K = <\tau>$가 순환군 $GF(p)^\times$상의 위수 k인 부분군이기 때문에, 잉여군(quotient group) $GF(p)^\times/K$는 위수 m인 순환군이고, 군의 생성원은 $2K$이다. 따라서 우리는 $GF(p)^\times$의 coset decomposition을 식 (15)와 같이 disjoint union으로 나타낼 수 있다.

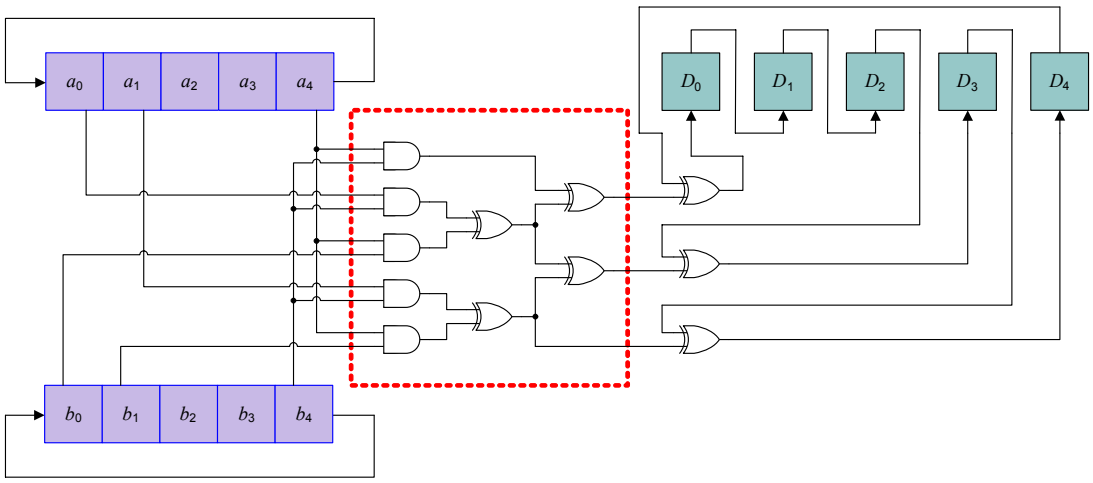

그림 2. $GF(2^5)$상의 Reyhani-Masoleh와 Hasan이 제안한 곱셈 회로도

$$GF(p)^{\times} = K_0 \cup K_1 \cup K_2 \cup \cdots \cup K_{n-1} \tag{15}$$

여기서 $K_i = 2^i K \ (0 \le i \le m-1)$ 이고 $GF(p)^{\times}$의 모든 원소는 임의의 $0 \le s \le k-1$과 $0 \le t \le m-1$에 대해, $\tau^s 2^t$로 유일하게 표현되고, $0 \le i \le m-1$에 대해 아래의 식 (16)을 얻을 수 있다.

$$\begin{aligned} \alpha\alpha_i &= \sum_{s=0}^{k-1}\beta^{\tau^s}\sum_{t=0}^{k-1}\beta^{\tau^t 2^i} = \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^{t-s}2^i)} \\ &= \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^t 2^i)} \end{aligned} \tag{16}$$

식 (16)으로부터, $1+\tau^u 2^v = 0 \in GF(p)$인 $0 \le u \le k-1$과 $0 \le v \le m-1$이 유일하게 존재한다. 만약, $t \ne u$또는 $i \ne v$이면, t와 i에 의해 결정되는 임의의 $0 \le \sigma(t,i) \le m-1$에 대하여, $1+\tau^t 2^i \in K_{\sigma(t,i)}$을 얻는다. 따라서 우리는 임의의 t'에 대해, $1+\tau^t 2^i = \tau^{t'} 2^{\sigma(t,i)}$와 같이 쓸 수 있다. 여기서 $i \ne v$일 때,

$$\begin{aligned} \alpha\alpha_i &= \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^t 2^i)} = \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(\tau^{t'}2^{\sigma(t,i)})} \\ &= \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^{s+t'}2^{\sigma(t,i)}} = \sum_{t=0}^{k-1}\alpha^{2^{\sigma(t,i)}} = \sum_{t=0}^{k-1}\alpha_{\sigma(t,i)} \end{aligned} \tag{17}$$

이다. 또한, $i = v$일 때,

$$\begin{aligned} \alpha\alpha_v &= \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^t 2^v)} \\ &= \sum_{t\ne u}\sum_{s=0}^{k-1}\beta^{\tau^s(\tau^{t'}2^{\sigma(t,v)})} + \sum_{s=0}^{k-1}\beta^{\tau^s(1+\tau^u 2^v)} \\ &= \sum_{t\ne u}\sum_{s=0}^{k-1}\beta^{\tau^{s+t'}2^{\sigma(t,v)}} + \sum_{s=0}^{k-1}1 \\ &= \sum_{t\ne u}\alpha^{2^{\sigma(t,v)}} + k = \sum_{t\ne u}\alpha_{\sigma(t,v)} + k \end{aligned} \tag{18}$$

이다. 따라서 $\alpha\alpha_i$는 $i \ne v$경우에 대해, $\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$에서 많아야 k개의 기저 원소의 합으로 계산되고, $\alpha\alpha_v$는 많아야 $k-1$개의 기저 원소와 상수 부분 $k \equiv 0,1 \in GF(2)$의합으로 계산된다.

Ⅳ. GNB를 이용한 $GF(2^m)$상의 새로운 곱셈 알고리즘

4.1 $\lambda_{ij}^{(s)}$와 λ_{ij}의 대칭성

$GF(2^m)$상에서 구현되는 ECC는 높은 안전성을 위해, 소수인 m을 요구한다. 이러한 조건은 Pohlig-Hellman 형태의 공격을 회피하기 위해 필요하다. 예를 들면, NIST와 IEEE 1363에서 ECDSA를 위해 권고하는 다섯 가지 $GF(2^m)$, $m \in \{163, 233, 283, 409, 571\}$은 모두 소수이다. 즉 m이 홀수 일 때 GNB Type k는 임의의 $k \ge 1$에 대해 항상 존재한다[15]. m이 홀수이고 $mk+1$은 소수이므로 k는 짝수이다. 따라서 우리는 m은 홀수, k는 짝수에 대한 $GF(2^m)$상의 GNB Type k에 대해서만 고려한다. 일반적으로 k가 작은값 일 때 낮은 복잡도를 가지는 곱셈기의 설계가 가능한데 $k \ge 1$ 가장 작은 짝수 정수는 2이다. 이와 같은 조건을 만족할 경우, 우리는 ONB Type 2 또는 GNB Type 2라 부른다. NIST에서 권고하는 다섯 가지 필드 사이즈 중 $m = 233$일 때 GNB Type 2가 존재한다. 나머지 필드 사이즈 중 낮은 복잡도를 가지는 GNB는 $m = 163,\ 409$로서 GNB Type 4이다. 또한 $m = 283$은 GNB Type 6이고, $m = 571$은 GNB Type 10이다[12].

$\alpha_i = \alpha^{2^i}$라 정의하고 $\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$을 $GF(2^m)$상에서 정규 기저라 하자. 또한,

$$\alpha\alpha_i = \sum_{j=0}^{m-1}\lambda_{ij}\alpha_j \tag{19}$$

라 할 때, 계수 λ_{ij}는 $GF(2)$의 원소이다. 식 (19)의 양변에 2의 거듭 제곱 연산을 반복 수행하면, 우리는 아래의 식 (20)을 얻을 수 있다.

$$\lambda_{ij}^{(s)} = \lambda_{i-j,s-j} \tag{20}$$

식 (20)에서, $\lambda_{ij}^{(s)}$는 식 (1)과 같다. 일반적으로 $\lambda_{ij}^{(s)}$은 대칭 행렬이지만 (λ_{ij})은 대칭 행렬이 아님에 유의해야한다. 그러나 k가 짝수인 GNB Type k라면 (λ_{ij})은 대칭 행렬이다.

보조 정리 1. 만약 $\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$이 k가 짝수인 GNB Type k라면 $\lambda_{ij}^{(0)} = \lambda_{i,j}$이다.

(증명) 식 (20)으로부터 $\lambda_{ij} = \lambda_{i-j,-j}$임을 보이면 된다. 식 (17), (18)로부터 만약 아래의 식 (21)을 만족하는 홀수쌍의 (s, s') (mod k)가 존재 한다면 $\lambda_{ij} = 1$임은 분명하다.

$$1 + \tau^s 2^i = \tau^{s'} 2^j \tag{21}$$

식 (21)에서 $<\tau>$는 소수 $p = mk+1$에 대해 $GF(p)^{\times}$상에서 위수 k인 유일한 곱셈 부분군이다. S

는 식 (21)을 만족하는 모든 순서쌍 (s,s') (mod k)의 집합이라 하고 같은 방법으로 $1+\tau^{i}2^{i-j}=\tau^{j}2^{-j}$를 만족하는 모든 순서쌍 (t,t') (mod k)의 집합을 T라 정의하자. $\lambda_{ij}=\lambda_{i-j,-j}$임을 증명하기 위해 집합 S와 T가 같은 기수를 가짐을 보이면 된다. 식 (21)의 양변을 $\tau^{s'}2^{j}$로 나누면 우리는 아래의 식 (22)를 얻을 수 있다.

$$\tau^{-s'}2^{-j}+\tau^{s-s'}2^{i-j}=1 \tag{22}$$

τ의 위수 k가 짝수 이므로 $-1=\tau^{\frac{k}{2}}$이다. 따라서 아래의 식 (23)은 만족한다.

$$\tau^{-s'}2^{-j}=1+\tau^{\frac{k}{2}+s-s'}2^{i-j} \tag{23}$$

사상 f_S: $S\rightarrow T$는 $f_S(s,s')=(\frac{k}{2}+s-s',-s')$에 의해 정의되고 사상 f_T: $T\rightarrow S$는 $f_T(t,t')=(\frac{k}{2}+t-t',-t')$에 의해 정의되므로 1:1대응이다. 즉, $f_S\circ f_T=id=f_T\circ f_S$이다.

4.2 GNB를 이용한 $GF(2^m)$상의 선형 곱셈기

식 (6)과 보조 정리 1로부터 $C=\sum_{s=0}^{m-1}c_s\alpha_s$의 계수 c_s는

$$\begin{aligned} c_s=\sum_{i,j}a_{i+s}b_{j+s}\lambda_{ij}^{(0)}&=\sum_{i,j}a_{i+s}b_{j+s}\lambda_{ij}\\ &=\sum_{j=0}^{m-1}(\sum_{i=0}^{m-1}a_{i+s}\lambda_{ij})b_{j+s}\end{aligned} \tag{24}$$

와 같다. 대응되는 행렬 $X=(x_{st})$에 대해 $GF(2)$상의 원소 x_{st}, $0\le s,t\le m-1$을 아래의 식 (25)와 같이 정의하면,

$$x_{st}=(\sum_{i=0}^{m-1}a_{i+s}\lambda)b_{t+s} \tag{25}$$

X의 t번째 열벡터 X_t는 아래와 같다.

$$X_t=(x_{0t},x_{1t},\cdots,x_{m-1,t})^T \tag{26}$$

여기서, $(x_{0t},x_{1t},\cdots,x_{m-1,t})^T$는 행벡터 $(x_{0t},x_{1t},\cdots,x_{m-1,t})$의 전치 행렬이다. 또한, $\sum_{t=0}^{m-1}x_{xt}=c_s$이기 때문에 모든 열벡터 X_t, $t=0,1,\cdots,m-1$의 합은 정확히 식 (27)과 같다.

$$(c_0,c_1,\cdots,c_{m-1})^T \tag{27}$$

여기서, 우리는 식 (26)의 열벡터 X_t를 재배열하고 계산 과정의 부분합 신호를 재사용함으로써 게이트 복잡도 및 최대 처리기 지연시간을 줄일 수 있다. $m-1=2\nu$라 하고 $Y=(y_{st})$를 X의 열벡터 치환에 의한 $m\times m$행렬이라 정의하면, ν가 홀수일 때 Y는 식 (28)과 같이 정의되고,

$$\begin{aligned}&(X_\nu,\cdots,X_3,X_1,X_{m-1},X_{m-3},\cdots,X_{m-\nu},\\&X_{\nu-1},\cdots,X_2,X_0,X_{m-2},\cdots,X_{m-\nu+1})\end{aligned} \tag{28}$$

ν가 짝수일 때, Y는 식 (29)와 같이 각각 정의된다.

$$\begin{aligned}&(X_\nu,\cdots,X_2,X_0,X_{m-2},\cdots,X_{m-\nu},X_{\nu-1},\\&\cdots,X_3,X_1,X_{m-1},X_{m-3},\cdots,X_{m-\nu+1})\end{aligned} \tag{29}$$

여기서, $Y_t=(y_{0t},y_{1t},\cdots,y_{m-1,t})^T$인 Y의 모든 열벡터 Y_t, $t=0,1,\cdots,m-1$의 합은 $(c_0,c_1,\cdots,c_{m-1})^T$인 X의 모든 열벡터 X_t, $t=0,1,\cdots,m-1$의합과 같다. 따라서 우리는 패러럴 입·출력 곱셈기를 설계하기위해 Y의 열벡터 합을 계산하는 대신 Y의 순환 쉬프트 된 대각 벡터의 합을 계산한다. 즉, 행렬 Y의 표현에서 벡터 X_t와 X_{m-t}사이에 정확히 $t-1$개의 열이 존재한다. 또한, X_t의 s번째 원소와 X_{m-t}의 $s+t$번째 원소는 그들의 가수(summand)에서 동일한 a_{is}를 가진다. 다시 말하면, 우리는 식 (25)로부터 아래의 식 (30)을 얻을 수 있다.

$$\begin{aligned} x_{s+t,m-t}&=(\sum_{i=0}^{m-1}a_{i+s+t}\lambda_{i,-t})b_s\\ &=(\sum_{i=0}^{m-1}a_{i+s}\lambda_{i-t,-t})b_s\\ &=(\sum_{i=0}^{m-1}a_{i+s}\lambda)b_s\end{aligned} \tag{30}$$

식 (30)에서 세 번째 표현식은 아래 첨자 i에서 합의 재배열로 나오고 마지막 식 표현은 보조 정리 1에서 언급된 $\lambda_{ij}=\lambda_{i-j,-j}$로부터 나온다. 따라서 x_{st}와 $x_{s+t,m-t}$는 같은 항 $\sum_{i=0}^{m-1}a_{i+s}\lambda$을 가진다. 이는 결국 AB의 계산에 있어 부분합 신호의 재사용을 의미한다. 지금까지 설명한 내용을 바탕으로 우리는 아래와 같은 $GF(2^m)$상의 새로운 GNB 곱셈 알고리즘을 얻을 수 있다.

알고리즘 1. GNB를 이용한 $GF(2^m)$상의 새로운 곱셈 알고리즘

Input : $A, B \in GF(2^m)$
Output : D, $D_i = c_i$ for all $0 \le i \le m-1$,
where $AB = \sum_{i=0}^{m-1} c_i \alpha_i$.
Initial : $A \leftarrow (a_0, a_1, \cdots, a_{m-1})$
$B \leftarrow (b_0, b_1, \cdots, b_{m-1})$
$D \leftarrow (D_0, D_1, \cdots, D_{m-1}) \leftarrow (0,0,\cdots,0)$.
1. For $t=0$ to $m-1$
2. For $s=0$ to $m-1$
3. $D_{s+t+1} \leftarrow y_{s,s+t} + D_{s+t}$
4. End for
5. End for
6. Return D

알고리즘 1에서 초기 루프($t=0$)에서 $D_{s+1} = D_s + y_{ss}$는 모든 $0 \le s \le m-1$에 대해 다음과 같이 동시에 계산된다. $D_1 = y_{00}$, $D_2 = y_{11}, \cdots, D_0 = y_{m-1,m-1}$이다. 또한 $t=1$일 때, $D_{s+2} = D_{s+1} + y_{s,s+1}$은 모든 $0 \le s \le m-1$에 대해 아래의 식 (31)과 같이 동시에 계산되며,

$$\begin{aligned} D_2 &= D_1 + y_{01} = y_{00} + y_{01}, \\ D_3 &= D_2 + y_{12} = y_{11} + y_{12}, \\ &\vdots \\ D_1 &= D_0 + y_{m-1,0} = y_{m-1,m-1} + y_{m-1,0} \end{aligned} \tag{31}$$

마지막으로, m번째 주기 ($t=m-1$)에서 $D_s = D_{s-1} + y_{s,s-1}$은 식 (32)와 같이 동시에 계산된다.

$$\begin{aligned} D_0 &= D_{m-1} + y_{0,m-1} = y_{00} + y_{01} + \cdots + y_{0,m-1} = c_0, \\ D_1 &= D_0 + y_{10} = y_{11} + y_{12} + \cdots + y_{10} = c_1, \\ &\vdots \\ D_{m-1} &= D_{m-2} + y_{m-1,m-2} \\ &= y_{m-1,m-1} + y_{m-1,0} + \cdots + y_{m-1,m-2} \\ &= c_{m-1} \end{aligned} \tag{32}$$

다시 말하면, 고정된 s에 대해 마지막 값 D_s는 아래의 식 (33)의 순서에 따라 순차적으로 계산된다.

$$\begin{aligned} D_s &= \underbrace{\overbrace{y_{ss}}^{D_{s+1}} + y_{s,s+1}}_{D_{s+2}} + y_{s,s+2} + \cdots + y_{s,s-1} \\ &= \sum_{i=0}^{m-1} y_{s,s+i} = c_s \end{aligned} \tag{33}$$

식 (32)에서 $0 \le s \le m-1$에 대한 $y_{s-1,s}$와 $y_{s,s}$는 행렬 Y의 같은 열 Y_s로부터 온다. 여기서, Y는 행렬 X의 열 치환에 의해 얻어졌으므로 s에 따라 임의의 s'에 대해 $y_{s-1,s} = x_{s-1,s'}$이고 $y_{s,s} = x_{s,s'}$임을 알 수 있다. 또한 식 (25)로부터 우리는 식 (34)를 얻을 수 있다.

$$\begin{aligned} x_{ss'} &= (\sum_{i=0}^{m-1} a_{i+s}\lambda_{is'}) b_{s'+s}, \\ x_{s-1,s'} &= (\sum_{i=0}^{m-1} a_{i+s-1}\lambda_{is'}) b_{s'+s-1} \end{aligned} \tag{34}$$

식 (34)에서 $x_{s-1,s'}(y_{s-1,s})$는 $x_{s,s'}(y_{s,s})$로부터 벡터 a_i와 b_i의 오른쪽으로 1-비트 순환 쉬프트를 의미하기 때문에 연산에 있어 어떠한 논리 게이트도 필요하지 않다. 따라서 곱셈에 있어 식 (35) 연산만이 논리 게이트를 필요로 한다.

$$D_{s+1} = Ds + y_{ss}, \ 0 \le s \le m-1 \tag{35}$$

다시 말하면, 치환 연산이기 때문에 식 (36)과 같이 각 s에 대응하는 s'이 있다.

$$y_{ss} = x_{ss'} = (\sum_{i=0}^{m-1} a_{i+s}\lambda_{is'}) b_{s'+s} \tag{36}$$

만약 $s' \neq 0$이면, 즉 $x_{ss'}$이 X의 0번째 열이 아니라면 식 (25), (30)으로부터 $x_{ss'}$과 $x_{s+s',m-s'}$(행렬 Y의 대각 원소)를 계산하기 위한 XOR 게이트는 공유될 수 있다. $x_{ss'} = (\sum_{i=0}^{m-1} a_{i+s}\lambda_{is'}) b_{s'+s}$ 연산에 필요한 게이트는 다음과 같다. 식 (17)에 나타나듯이 GNB 타입 k의 곱셈 행렬 (λ_{ij})는 각 열(행)에 대해 많아야 k개의 0이 아닌 원소를 가지므로 하나의 AND 게이트와 많아야 $k-1$개의 XOR 게이트가 필요하다. 따라서 $s' \neq 0$ 경우 모든 $y_{s,s} = x_{s,s'}$을 계산하기 위해 필요한 총 게이트 수는 $m-1$개의 AND 게이트와 $\frac{m-1}{2}(k-1)$개의 XOR 게이트이다. 또한, $s'=0$ 경우, $0 \le i \le m-1$에 대해, λ_{i0}의 0이 아닌 원소의 개수는 $\alpha\alpha_0 = \alpha^2 = \alpha_1$이기 때문에 1이다. 따라서 $s'=0$일 때 $x_{s,s'}$ 계산에 필요한 게이트는 단지 AND 하나이다. 식 (35)에서 덧셈 $D_s + y_{ss}$는 $0 \le s \le m-1$에 대해 하나의 XOR 게이트만 필요하다. 따라서 알고리즘 1의 곱셈에 있어 필요한 총 게이트 수는 m개의 AND 게이트와 많아야 $m + \frac{m-1}{2}(k-1)$개의 XOR 게이트이다. 또한 최대

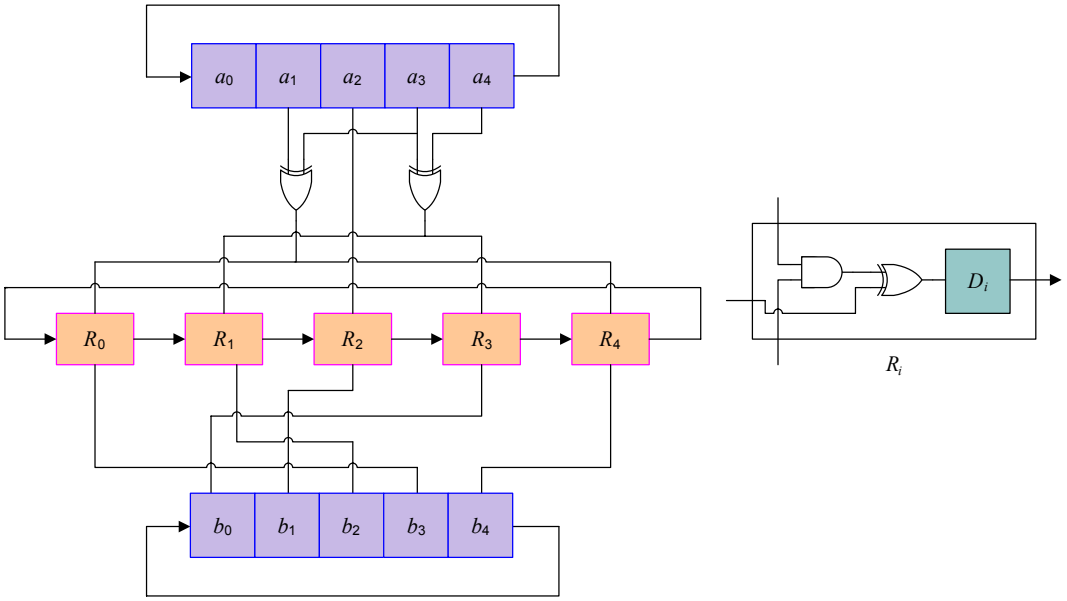

그림 3. $GF(2^5)$상의 새로운 GNB Type 2 곱셈기

처리기 지연 시간은 식 (35)과 (36)으로부터 $T_A+(1+\lceil \log_2 k \rceil)T_X$임을 알 수 있다. 표 1에 기존에 제안된 정규기저 곱셈기와 본 논문에서 제안된 곱셈기를 시간 및 하드에어 복잡도 측면에서 비교하였다. 표 1에서 C_N은 행렬 $(\lambda_{ij}^{(0)})$에서 0이 아닌 원소의 개수이다. k가 홀수이면 $C_N < mk+m-k$이고, k가 짝수이면 $C_N \le mk-1$임은 잘 알려져 있다[3]. 우리가 제안한 곱셈기는 ECC의 적용을 위해 m이 홀수인 경우로서 식 (17)과 (18)로부터 $C_N \le mk-k+1$이다. 따라서 표 1의 Reyhani-Masoleh와 Hasan 곱셈기의 $(C_N+1)/2+\lfloor m/2 \rfloor$는

$$\frac{mk-k+2}{2}+\frac{m-1}{2}=\frac{2m+mk-m-k+1}{2}$$

$$=m+\frac{m-1}{2}(k-1)$$

이다. 따라서 표 1에 기술된바와 같이 본 논문에서 제안된 곱셈기는 Agnew등이 제안한 곱셈기와 동일한 최대 처리기 지연 시간을 가지지만 낮은 하드웨어 복잡도를 가지고 Reyhani-Masoleh와 Hasan 곱셈기와 동일한 하드웨어 복잡도를 가지지만 낮은 최대 처리기 지연시간을 가진다.

V. ECC를 위한 $GF(2^m)$상의 GNB Type 2, 4 곱셈기

이번 절에서는 지금까지 기술한 내용을 바탕으로

표 1. $GF(2^m)$상의 정규기저를 이용한 곱셈기의 성능분석

	최대 처리기 지연 시간 (ONB Type 2인 경우)	AND	XOR (ONB Type 2인 경우)	플립플롭
Massey and Omura [7]	$\le TA+\lceil \log_2(mk) \rceil TX$ $(TA+\lceil \log_2(2m) \rceil TX)$	C_N	$\le C_N-1$ $(2m-2)$	$2m$
Agnew등. [1]	$\le TA+(1+\lceil \log_2 k \rceil)TX$ $(TA+2TX)$	m	$\le C_N$ $(2m-1)$	$3m$
Reyhani-Masoleh와 Hasan [3]	$\le TA+(1+\lceil \log_2(k+2) \rceil)TX$ $(TA+3TX)$	m	$\le (C_N+1)/2+\lfloor m/2 \rfloor$ $((3m-1)/2)$	$3m$
제안된 곱셈기	$\le TA+(1+\lceil \log_2 k \rceil)TX$ $(TA+2TX)$	m	$\le m+(m-1)(k-1)/2$ $((3m-1)/2)$	$3m$

$GF(2^m)$상의 GNB Type 2와 4에 대한 실제적인 VLSI 곱셈 회로 구현 예를 보이며, 동일한 방법으로 모든 GNB Type에 대해 적용할 수 있다.

5.1 $GF(2^m)$상의 GNB Type 2 곱셈기

$p=2m+1$가 $\gcd(2m/\mathrm{ord}_p 2, m)=1$인 소수라 하자. 즉, 2가 (mod p)에 대한 원시근이거나 m이 홀수이고 $\mathrm{ord}_p 2=m$이면, 원소 $\alpha=\beta+\beta^{-1}$는 $GF(2^m)$상의 NB $\{\alpha_0,\alpha_1,\cdots,\alpha_{m-1}\}$을 형성한다. 여기서 β는 $GF(2^{2m})$상에서 p번째 원시근이다. 위와 같은 경우를 우리는 ONB Type 2 혹은 GNB Type 2라 부른다. GNB Type 2에 있어 $\alpha\alpha_i$의 곱셈 행렬 (λ_{ij})는 다음과 같은 속성을 가진다.

$$\lambda_{ij}=1 \text{ iff } 1\pm 2^i \equiv \pm 2^j \pmod{p} \tag{37}$$

식 (37)은 III장의 GNB의 기본 속성으로부터 m은 $\mathrm{ord}_p 2$을 나누기 때문에 $1\pm 2^i \equiv 0 \pmod{p}$를 만족하는 $i=0 \pmod{m}$은 유일하다. 즉, $\alpha\alpha_0=\alpha_1$이고, (λ_{ij})의 0번째 행은 $(0,1,0,\cdots,0)$이다. $i\neq 0$이면 $1\pm 2^i \neq 0 \pmod{p}$이고 (λ_{ij})의 $i(\neq 0)$번째 행은 정확히 두 개의 0이 아닌 원소를 포함한다. 그러므로 GNB Type 2의 경우에 대해 m개의 AND 게이트와 $m+\frac{m-1}{2}=\frac{3m-1}{2}$개의 XOR 게이트를 필요로 한다. 또한, 최대 처리기 지연 시간은 T_A+2T_X이다. 명확한 설명을 위해 우리는 아래에 GNB Type 2에 대한 예를 보인다.

예제 1. β를 $GF(2^{10})$상에서 11번째 원시근이라 하고 $\alpha=\beta+\beta^{-1}$은 $GF(2^5)$상의 GNB Type 2 원소라 하면, $0\leq i\leq 4$에 대한 $\alpha\alpha_i$의 계산은 표 2로부터 쉽게 얻을 수 있다. 각 블록 K와 K'은, 엔트리 (s,t)는, $(0\leq s\leq 1,\ 0\leq t\leq 4)$, 각각 $\tau^s 2^t$와 $1+\tau^s 2^t$의 값을 가진다. 여기서, $\langle\tau\rangle=\langle -1\rangle$은 $GF(11)^{\times}$상의 위수 2인 유일한 곱셈 부분군이다.

표 2로부터 $\alpha\alpha_0=\alpha_1$이고 나머지 부분은 식 (38)과 같다.

표 2. $GF(2^5)$ 상의 GNB Type 2를 사용한 K_i와 K_i'의 계산

K_0	K_1	K_2	K_3	K_4	K_0'	K_1'	K_2'	K_3'	K_4'
1	2	3	4	5	2	3	5	9	6
-1	-2	-3	-4	-5	0	-1	-3	-7	-4

$$\begin{aligned}\alpha\alpha_1&=\alpha_0+\alpha_3,\ \alpha\alpha_2=\alpha_3+\alpha_4,\\ \alpha\alpha_3&=\alpha_1+\alpha_2,\ \alpha\alpha_4=\alpha_2+\alpha_4.\end{aligned} \tag{38}$$

예를 들어, $\alpha\alpha_3$의 계산은 다음과 같이 할 수 있다. 블록 K_3'을 보면, $9\equiv -2 \pmod{11}$은 K_1블록에, $-7\equiv 4 \pmod{11}$은 K_2블록에서 각각 찾을 수 있다. 따라서 $\alpha\alpha_3=\alpha_1+\alpha_2$이다. 식 (38)로부터 완전한 $GF(2^5)$상의 곱셈 행렬 (λ_{ij})은 식 (39)와 같다.

$$(\lambda_{ij})=\begin{pmatrix}0&1&0&0&0\\1&0&0&1&0\\0&0&0&1&1\\0&1&1&0&0\\0&0&1&0&1\end{pmatrix} \tag{39}$$

위의 식 (39)와 식 (24), (25), (28), (29)를 사용하여 $A=\sum_{i=0}^{4}a_i\alpha_i$와 $B=\sum_{i=0}^{4}b_i\alpha_i$의 곱 $C=\sum_{i=0}^{4}c_i\alpha_i$는 아래의 식 (40)과 같이 계산 될 수 있다. 식 (40)으로부터 $GF(2^5)$상의 GNB Type 2를 이용한 곱셈기는 그림 3과 같다. 식 (40)에서 밑줄 친 원소가 첫 번째 클럭 사이클에 계산되고 쉬프트된 대각 원소는 동일한 a_i를 가진다.

5.2 $GF(2^m)$상의 GNB Type 4 곱셈기

NIST 및 IEEE 1363에서 권고하는 다섯 가지 $GF(2^m)$, $m\in\{163, 233, 283, 409, 571\}$ 중 GNB Type 2는 단지 $GF(2^{233})$ 하나뿐이다. 또한 GNB Type 2를 이용한 곱셈 회로는 많이 알려져 있지만, GNB Type $k\geq 4$ 경우에 대한 곱셈 회로는 현재까지 알려져 있지 않다. 본 절에서는 GNB Type $k\geq 4$경우 중 GNB Type 4에 대한 회로 설계의 예를 보이며, 나머지 GNB Type에 대해서도 동일하게 적용될 수 있다. 참고로 $GF(2^{163})$와 $GF(2^{409})$는 GNB Type 4이다. 보다 명확한 설명을 위해 $m=7$인 경우에 대한 예를 제시한다.

$$\begin{aligned}
c_0&=\underline{(a_3+a_4)b_2}+a_1b_0+(a_1+a_2)b_3+(a_0+a_3)b_1+(a_2+a_4)b_4\\
c_1&=(a_4+a_0)b_3+\underline{a_2b_1}+(a_2+a_3)b_4+(a_1+a_4)b_2+(a_3+a_0)b_0\\
c_2&=(a_0+a_1)b_4+a_3b_2+\underline{(a_3+a_4)b_0}+(a_2+a_0)b_3+(a_4+a_1)b_1\\
c_3&=(a_1+a_2)b_0+a_4b_3+(a_4+a_0)b_1+\underline{(a_3+a_1)b_4}+(a_0+a_2)b_2\\
c_4&=(a_2+a_3)b_1+a_0b_4+(a_0+a_1)b_2+(a_4+a_2)b_0+\underline{(a_1+a_3)b_3}
\end{aligned} \tag{40}$$

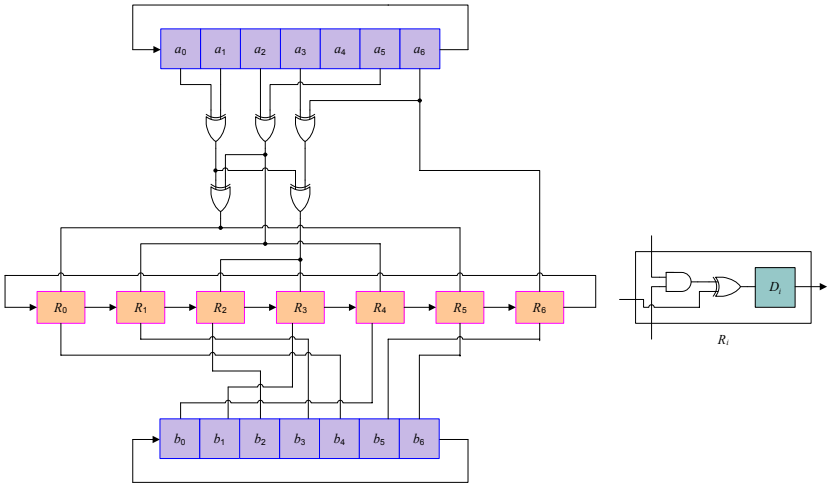

그림 4. $GF(2^7)$상의 새로운 GNB Type 4 곱셈기

예제 2. $GF(2^7)$이 GNB Type 4라 하면, $m=7$, $k=4$, 즉 $p=29=mk+1$이다. 이 경우, $GF(29)^{\times}$상에서 위수 4인 유일한 순환군은 $K=\{1,2^7,2^{14},2^{21}\}=\{1,12,28,17\}$이다. 여기서 β를 $GF(2^{28})$상에서 29번째 원시근이라하고, $\tau=12$로 두면, $GF(2^7)$상의 정규 원소 α는 $\alpha=\beta+\beta^{12}+\beta^{17}+\beta^{28}$로 표현되고, $\{\alpha_0,\alpha_1,\cdots,\alpha_6\}$는 정규 기저이다. $0\le i\le 6$에 대해, $\alpha\alpha_i$의 계산은 표 3으로부터 얻을 수 있다. 각 블록 K와 K'은, 엔트리 (s,t), $0\le s\le 3$, $0\le t\le 6$에 대해 각각 $\tau^s 2^t$와 $1+\tau^s 2^t$의 값을 가진다.

표 3. $GF(2^7)$ 상의 GNB 타입 4를 사용한 K_i와 K_i'의 계산

K_0	K_1	K_2	K_3	K_4	K_5	K_6	K_0'	K_1'	K_2'	K_3'	K_4'	K_5'	K_6'
1	2	4	8	16	3	6	2	3	5	9	16	4	7
12	24	19	9	18	7	14	13	25	20	10	19	8	15
28	27	25	21	13	26	23	0	28	26	22	14	27	24
17	5	10	20	11	22	15	18	6	11	21	12	23	16

위의 표로부터 $\alpha\alpha_0=\alpha_1$이고, 나머지 부분은 아래의 식 (41)과 같다.

$$\begin{aligned}
\alpha\alpha_1&=\alpha_0+\alpha_2+\alpha_5+\alpha_6,\\
\alpha\alpha_2&=\alpha_1+\alpha_3+\alpha_4+\alpha_5,\\
\alpha\alpha_3&=\alpha_2+\alpha_5,\\
\alpha\alpha_4&=\alpha_2+\alpha_5,\\
\alpha\alpha_5&=\alpha_0+\alpha_2+\alpha_5+\alpha_6,\\
\alpha\alpha_6&=\alpha_1+\alpha_3+\alpha_4+\alpha_5.
\end{aligned}\tag{41}$$

예를 들어, $\alpha\alpha_2$에 대한 블록 K_2'은 다음과 같이 할 수 있다. 블록 K_2'의 원소는 5, 20, 26, 11이다. K_i의 블록들을 보면, $5\in K_1$, $20\in K_3$, $26\in K_5$, $11\in K_4$에서 찾을 수 있다. 따라서 $\alpha\alpha_2=\alpha_1+\alpha_3+\alpha_4+\alpha_5$이다. 식 (41)로부터 곱셈 행렬 (λ_{ij})은 아래의 식 (42)와 같다.

$$(\lambda_{ij})=\begin{pmatrix}
0&1&0&0&0&0&0\\
1&0&1&0&0&1&1\\
0&1&0&1&1&1&0\\
0&0&1&0&0&1&0\\
0&0&1&0&0&0&1\\
0&1&1&1&0&0&1\\
0&1&0&0&1&1&1
\end{pmatrix}\tag{42}$$

식 (42)의 곱셈 행렬과 식 (24), (25), (28), (29)으로부터 우리는 아래의 식 (43)과 같은 곱셈 결과 $C=AB=\sum_{i=0}^{6}c_i\alpha_i$을 얻을 수 있다. 식 (43)에서

$$\begin{aligned}
c_0&=\underline{(a_2+a_5)b_3}+a_{0256}b_1+a_{1456}b_6+(a_2+a_6)b_4+a_{1345}b_2+a_1b_0+a_{1236}b_5\\
c_1&=(a_3+a_6)b_4+\underline{a_{1360}b_2}+a_{2560}b_0+(a_3+a_0)b_5+a_{2456}b_3+a_2b_1+a_{2340}b_6\\
c_2&=(a_4+a_0)b_5+a_{2401}b_3+\underline{a_{3601}b_1}+(a_4+a_1)b_6+a_{3560}b_4+a_3b_2+a_{3451}b_0\\
c_3&=(a_5+a_1)b_6+a_{3512}b_4+a_{4012}b_2+\underline{(a_5+a_2)b_0}+a_{4601}b_5+a_4b_3+a_{4562}b_1\\
c_4&=(a_6+a_2)b_0+a_{4623}b_5+a_{5123}b_3+(a_6+a_3)b_1+\underline{a_{5012}b_6}+a_5b_4+a_{5603}b_2\\
c_5&=(a_0+a_3)b_1+a_{5034}b_6+a_{6234}b_4+(a_0+a_4)b_2+a_{6123}b_0+\underline{a_6b_6}+a_{6014}b_3\\
c_6&=(a_1+a_4)b_2+a_{6145}b_0+a_{0345}b_5+(a_1+a_5)b_3+a_{0234}b_1+a_0b_6+\underline{a_{0125}b_4}
\end{aligned}\tag{43}$$

a_{ijkl}은 $a_{ijkl}=a_i+a_j+a_k+a_l$을 의미하다. 식 (43)의 밑줄 친 원소가 첫 번째 클럭 사이클에 계산되고 쉬프트된 대각 원소는 동일한 a_i를 가진다. 지금까지 기술한 내용을 바탕으로 우리는 그림 4와 같은 $GF(2^7)$상의 GNB Type 4 곱셈기를 설계할 수 있다.

Ⅵ. 결 론

본 논문에서는 가우시안 정규기저를 이용하여, m이 홀수인 경우의 $GF(2^m)$상의 새로운 곱셈 알고리즘 및 VLSI 구조를 제안하였다. 지금까지 기술한 바와 같이 타원곡선 암호 시스템응용에 적용한다면, 실제적으로 m이 홀수이어야 한다는 제약조건은 거의 없는 것과 같다. 효율적인 알고리즘의 설계를 위해, 우리는 정규기저 원소의 대칭성이용과 계수의 인덱스를 변형을 이용하였다. 기존의 다른 $GF(2^m)$상의 NB 곱셈 알고리즘에 비해, 본 논문에서 제안한 곱셈 알고리즘은 동일한 방식으로 NIST 및 IEEE 1363에서 권고하는 다섯 가지 $GF(2^m)$, $m\in$ {163, 233, 283, 409, 571}, 모두에 적용 할 수 있다. 제안된 곱셈알고리즘에 기반한 VLSI 구조는 기존의 $GF(2^m)$상의 Agnew등이 제안한 곱셈기 보다 낮은 하드웨어 복잡도를 가지지만 동일한 최대 처리기 지연시간을 가지고 Reyhani-Masoleh와 Hasan이 제안한 곱셈기와 거의 동일한 하드웨어 복잡도를 가지지만 낮은 최대 처리기 지연시간을 보였다. 또한 본 논문에서 제안한 곱셈기의 구조는 매우 규칙적일 뿐만 아니라 GNB 타입만 동일하면 서로 다른 m에 대해서도 동일하게 적용되기 때문에 높은 확장성을 제공하다. 따라서 본문에서 제안된 곱셈기는 GNB상에서 구현되는 타원곡선 암호 프로세서의 곱셈기로 매우 적합하다 할 수 있다.

참 고 문 헌

〔1〕 G.B. Agnew, R.C. Mullin, I. Onyszchuk, and S.A. Vanstone, “An implementation for a fast public key cryptosystem,” *J. Cryptology*, vol. 3, pp. 63-79, 1991.

〔2〕 G.B. Agnew, R.C. Mullin, and S.A. Vanstone, “Fast exponentiation in $GF(2^n)$,” *Eurocrypt 88*, *Lecture Notes in Computer Science*, vol. 330, pp. 251-255, 1998.

〔3〕 A. Reyhani-Masoleh and M.A. Hasan, “Low complexity sequential normal basis multipliers over $GF(2^m)$,” *16th IEEE Symposium on Computer Arithmetic*, vol. 16, pp. 188-195, 2003.

〔4〕 A. Reyhani-Masoleh and M.A. Hasan, “A new construction of Massey-Omura parallel multiplier over $GF(2^m)$,” *IEEE Trans. Computers*, vol. 51, pp. 511-520, 2002.

〔5〕 A. Reyhani-Masoleh and M.A. Hasan, “Efficient multiplication beyond optimal normal bases,” *IEEE Trans. Computers*, vol. 52, pp. 428-439, 2003.

〔6〕 A.J. Menezes, I.F. Blake, S. Gau, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian, “Applications of Finite Fields,” *Kluwer Academic Publisher*, 1993.

〔7〕 J.L. Massey and J.K. Omura, “Computational method and apparatus for finite field arithmetic,” *US Patent No. 4587627*, 1986.

〔8〕 C. Parr, P. Fleschmann, and P. Roelse, “Efficient multiplier architectures for Galois fields $GF(2^{4n})$,” *IEEE Trans. Computers*, vol. 47, pp. 162-170, 1998.

〔9〕 E.R. Berlekamp, “Bit-serial Reed-Solomon encoders,” *IEEE Trans. Inform. Theory*, vol. 28. pp. 869-874, 1982.

〔10〕 B. Sunar and C.K. Koc, “An efficient optimal normal basis type II multiplier,” *IEEE Trans. Computers*, vol 50, pp. 83-87, 2001.

〔11〕 H. Wu, M.A. Hasan, I.F. Blake, and S. Gao, “Finite field multiplier using redundant representation,” *IEEE Trans. Computers*, vol 51, pp. 1306-1316, 2002.

〔12〕 S. Gao, J. von zur Gathen, and D. Panario, “Orders and cryptographical applications,” *Math. Comp.*, vol. 67, pp. 343-352, 1998.

〔13〕 J. von zur Gathen and I. Shparlinski, “Orders of Gauss periods in finite fields,” *ISAAC 95, LNCS*, vol. 1004, pp. 208-215, 1995.

〔14〕 S. Gao, S. Vanstone, “On orders of optimal normal basis generators,” *Math. Comp.*, vol. 64, pp. 1227-1233, 1995.

〔15〕 S. Feisel, J. von zur Gathen, and M. Shokrollahi, “Normal bases via general Gauss periods,” *Math. Comp.*, vol. 68, pp. 271-290, 1999.

〔16〕 NIST, “Digital Signature Standard,” *FIPS Publication*, 186-2, Feb. 2000.

〔17〕 D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer Verlag, 2004.

〔18〕 IEEE 1363, “IEEE Standard specifications for public-key cryptography,” Jan. 2000.

권 순 학 (Soonhak Kwon) 정회원
한국통신학회 논문지 제31권 11C호 참조
현재 성균관대학교 수학과 부교수

김 희 철 (Hiecheol Kim) 정회원
한국통신학회 논문지 제27권 3C호 참조
현재 대구대학교 정보통신공학과 부교수

김 창 훈 (Chang Hoon Kim) 정회원
한국통신학회 논문지 제31권 11C호 참조
현재 대구대학교 정보통신공학과 BK21 연구교수

홍 춘 표 (Chun Pyo Hong) 정회원
한국통신학회 논문지 제31권 11C호 참조
현재 대구대학교 정보통신공학부 교수