합성체상의 효율적인 최적정규기저 곱셈기†

권윤기*, 권순학*, 김창훈**, 김희철***
*성균관대학교 수학과
**대구대학교 컴퓨터·IT 공학부
***대구대학교 정보통신공학부
e-mail:ykkwon@skku.edu, shkwon@skku.edu, kimch@daegu.ac.kr, hckim@daegu.ac.kr

Efficient Optimal Normal Basis Multipliers Over Composite Fields

Yun Ki Kwon*, Soonhak Kwon*, Chang Hoon Kim**, Hiecheol Kim***
*Dept of Mathematics, Sungkyunkwan University
**School of Computer & Information Technology, Daegu University
***School of Information & Communication Engineering, Daegu University

요 약

최적정규기저(Optimal Normal Basis)를 이용한 $GF(2^m)$상의 곱셈은 ECC(Elliptic Curve Cryptosystems: 타원곡선 암호시스템) 및 유한체 산술 연산의 하드웨어 구현에 적합하다는 것은 잘 알려져 있다. 본 논문에서는 최적정규기저의 하드웨어적 장점을 이용하여 합성체(Composit Field)상의 곱셈기를 제안하며, 기존에 제안된 합성체상의 곱셈기와 비교 및 분석한다. 제안된 곱셈기는 최적정규기저 타입 I, II의 대칭성과 가수의 중복성을 이용한 열벡터의 재배열에 따른 XOR 연산의 재사용으로 낮은 하드웨어 복잡도와 작은 지연시간을 가진다.

1. 서론

최근 pairing 기반 암호시스템 및 ECC는 공개키 암호시스템에서 많은 관심을 가지는 분야이다. 특히 유한체 산술 연산은 ElGamal, ECC와 같은 공개키 암호시스템의 바탕이 되며 암호학의 많은 분야에서 중요하게 다루어지고 있다. Massey-Omura 비트-패러럴 곱셈기[2]가 처음 제안된 이후, 정규기저를 이용한 유한체상의 산술 연산에 대한 하드웨어 구현에 많은 연구가 이루어졌다3-5]. Massey-Omura 곱셈기에서 보듯이 정규기저 표현에 의한 $GF(2^m)$상의 산술 연산을 하드웨어로 구현할 때 최적정규기저를 이용하면 가장 효율적이다. 유한체 $GF(2^m)$은 $GF(2)$의 확장체로서 합성체 $GF(2^{20})$의 경우 $GF(2)$상의 차수가 4인 기약다항식을 이용하여 표현된 $GF(2^5)$상의 확장체로 간주할 수 있다.

$GF(2)$에 대한 $GF(2^m)$상의 정규기저는 항상 존재함은 잘 알려져 있다[1]. 또한, m, n이 서로 소인 경우 $GF(2)$에 대한 $GF(2^m)$상의 정규기저가 존재하며, $GF(2^n)$에 대한 $GF(2^{mn})$상의 정규기저가 존재한다는 것이 잘 알려져 있다[5]. 따라서 $GF(2^m)$상에 대한 $GF(2^{mn})$상 결합된 정규기저를 가진다. 특별한 경우 $GF(2)$에 대한 $GF(2^m)$상의 최적정규기저가 존재하며 m, n이 서로 소인 경우 $GF(2^m)$에 대한 $GF(2^{mn})$상의 최적정규기저도 존재한다. 위 사실로부터 최적정규기저의 성질을 이용하여 합성체 $GF(2^m)$상의 효율적인 하드웨어 구현이 가능하며 제곱 연산 및 역원 연산이 많은 pairing 기반 암호시스템 및 ECC에서 최적정규기저의 사용은 유용하다. 특히 pairing 기반 암호시스템에서 표수 2인 경우 embedding degree가 4이므로 최적정규기저를 이용한 합성체에서 효율적이다.

† 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임

2. 관련연구

유한체 $GF(2^m)$는 2^m개의 원소를 가지는 $GF(2)$상의 m차원 벡터 공간이다. 임의의 원소 α에 대해, $N=\{\alpha,\alpha^2,\cdots,\alpha^{2^{m-1}}\}$ 형태의 기저를 $GF(2^m)$의 정규기저(Normal Basis)라 한다. 이 때, 임의의 원소 α는 정규기저의 생성원이며 모든 $m\geq 1$에 대해, $GF(2^m)$상의 정규기저가 존재함은 잘 알려져 있다[1]. 이러한 N에 대해, $\alpha_i=\alpha^{2^i}$라 놓으면, $N=\{\alpha_0,\alpha_1,\cdots,\alpha_{m-1}\}$과 같이 표현할 수 있다. $GF(2^m)$상의 임의의 원소 A는

$$A=a_0\alpha_0+a_1\alpha_1+\cdots+a_{m-1}\alpha_{m-1},\ a_i\in GF(2) \qquad (1)$$

로 정규기저로 표현할 수 있고, 이것을 벡터 형태로 표현하면

$$A=\mathbf{a}\cdot N^T,\quad \mathbf{a}=[a_0,a_1,\cdots,a_{m-1}],\ N=[\alpha_0,\alpha_1,\cdots,\alpha_{m-1}] \qquad (2)$$

이다. 여기서, T는 벡터의 전치 행렬을 나타낸다. $GF(2^m)$상의 제곱 연산은 벡터 $\mathbf{a}$의 오른쪽 순환 쉬프트로 간단하게 얻을 수 있다. 즉,

$$A^2=a_{m-1}\alpha_0+a_0\alpha_1+\cdots+a_{m-1}\alpha_{m-1} \qquad (3)$$

이다. 하지만, 곱셈 연산은 복잡하다.
$GF(2^m)$상의 임의의 두 원소 A, B의 곱을 $C=A\cdot B$라 하면,

$$\begin{aligned} C&=(\mathbf{a}\cdot N^T)\cdot(\mathbf{b}\cdot N^T)^T\\ &=(\mathbf{a}\cdot N^T)\cdot(N\cdot\mathbf{b}^T)\\ &=\mathbf{a}\cdot(\lambda_{i,j})\cdot\mathbf{b}^T \end{aligned} \qquad (4)$$

이며, 여기서 곱행렬(multiplication matrix) $(\lambda_{i,j})$은

$$(\lambda_{i,j}) = \begin{bmatrix} \alpha_0\alpha_0 & \alpha_0\alpha_1 & \cdots & \alpha_0\alpha_{m-1} \\ \alpha_1\alpha_0 & \alpha_1\alpha_1 & \cdots & \alpha_1\alpha_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m-1}\alpha_0 & \alpha_{m-1}\alpha_1 & \cdots & \alpha_{m-1}\alpha_{m-1} \end{bmatrix} \quad (5)$$

, ($0 \le i, j \le m-1$)으로 정의된다. N은 기저이므로, $\alpha_i\alpha_j$는

$$\alpha_i\alpha_j = \sum_{s=0}^{m-1} \lambda_{i,j}^{(s)}\alpha_s,\ \lambda_{i,j}^{(s)} \in GF(2) \quad (6)$$

와 같이 정규기저를 이용하여 나타낼 수 있고, 곱행렬 (λ_{ij})은

$$(\lambda_{i,j}) = (\lambda_{i,j})^{(0)}\alpha_0 + (\lambda_{i,j})^{(1)}\alpha_1 + \cdots + (\lambda_{i,j})^{(m-1)}\alpha_{m-1} \quad (7)$$

로 부울 행렬(Boolean Matrix)을 이용하여 나타낼 수 있으며, 부울 행렬 $(\lambda_{ij})^{(s)}$ ($0 \le s \le m-1$)의 원소는 $GF(2)$의 원소이다. 그러면 모든 $s(0 \le s \le m-1)$에 대해, $\alpha_i\alpha_j$을 정규 기저로 표현하면 $(\lambda_{ij})^{(s)}$의 i번째 행, j번째 열의 원소는 α_s의 계수를 포함한다. 일반적으로 $(\lambda_{ij})^{(s)}$는 대칭 행렬이다. 여기서, $(\lambda_{ij})^{(s)} = (\lambda_{i-1,j-1})^{(s-1)}$이므로, 식 (4)와 (7)로부터 두 원소의 곱 C의 계수 c_s에 대해 고려하면,

$$\begin{aligned} c_s &= \mathbf{a} \cdot (\lambda_{i,j})^{(s)} \cdot \mathbf{b}^T \\ &= \mathbf{a}^{(s)} \cdot (\lambda_{i,j})^{(0)} \cdot \mathbf{b}^{(s)\,T}, \\ & 0 \le s \le m-1 \end{aligned} \quad (8)$$

으로 나타낼 수 있고 여기서, $\mathbf{a}^{(s)}$는 $\mathbf{a}$를 s번 오른쪽 순환 쉬프트한 벡터이다. N에 대응되는 곱행렬에 대한 복잡도(complexity) C_N은 다음과 같이 정의된다.

$$C_N = \left|\left\{(i,j) | \lambda_{i,j}^{(0)} = 1,\ 0 \le i,j \le m-1\right\}\right|. \quad (9)$$

임의의 정규기저 N에 대해, $C_N \ge 2m-1$임이 잘 알려져 있다[1]. 특히, $C_N = 2m-1$이면 기저 N을 최적정규기저(Optimal Normal Basis)라 부른다.

정리 1[1].
(1) $m+1$이 소수이고 2가 $GF(m+1)$상의 원시근이면 $GF(2^m)$은 최적정규기저를 가진다.
(2) $2m+1$이 소수이고 2가 $GF(2m+1)$상의 원시근이면 $GF(2^m)$은 최적정규기저를 가진다.
(3) $2m+1$이 소수이고 $2m+1 \equiv 3 \pmod 4$을 만족하며 2가 $GF(2m+1)$상의 이차 잉여류이면 $GF(2^m)$은 최적정규기저를 가진다.

정의 2[1]. 정리 1의 성질 (1)을 만족하는 최적정규기저를 **최적정규기저 타입 I**이라 하고 정리 1의 성질 (2) 또는 (3)을 만족하는 최적정규기저를 **최적정규기저 타입 II**라 정의한다.
$GF(2^m)$상에서 최적정규기저 타입 I을 만족하는 m의 값은 2, 4, 10, 12, 18, 28, 36, 52, 등이며 최적정규기저 타입 II를 만족하는 m의 값은 2, 3, 5, 6, 9, 11, 14, 18, 23, 26, 29, 30, 33, 35, 39, 41,50, 51, 등이다.

정리 3[5]. 정리 1의 성질 (1)을 만족하는 최적정규기저 타입 I의 곱행렬에서 부울 행렬 $(\lambda_{ij})^{(0)}$의 원소는 다음 조건을 만족한다.

$$\lambda_{i,j}^{(0)} = \begin{cases} 1 & 2^i + 2^j \equiv 0 \text{ or } 1 \pmod{m+1} \\ 0 & \text{otherwise} \end{cases}, \quad 0 \le i,j \le m-1 \quad (10)$$

여기서, 지수 i, j는 mod m에 대해 고려한다.

정리 4[3]. 정리 1의 성질 (2), (3)을 만족하는 최적정규기저 타입 II의 곱행렬에서 부울 행렬 $(\lambda_{ij})^{(0)}$의 원소는 다음 조건을 만족한다.

$$\lambda_{i,j}^{(0)} = \begin{cases} 1 & 2^i \pm 2^j \equiv \pm 1 \pmod{2m+1} \\ 0 & \text{otherwise} \end{cases}, \quad 0 \le i,j \le m-1 \quad (11)$$

여기서, 지수 i, j는 mod m에 대해 고려한다.

일반적으로 최적정규기저는 타입 I, II로 나누어진다. 따라서 합성체 $GF(2^{mn})$은 확장된 최적정규기저를 이용하여 다음과 같이 3가지 경우로 분류할 수 있다.

합성체 타입 I. 부분체 $GF(2^m)$-최적정규기저 타입 II, 확장체 $GF(2^{mn})$-최적정규기저 타입 I.
합성체 타입 II. 부분체 $GF(2^m)$-최적정규기저 타입 I, 확장체 $GF(2^{mn})$-최적정규기저 타입 II.
합성체 타입 III. 부분체 $GF(2^m)$-최적정규기저 타입 II, 확장체 $GF(2^{mn})$-최적정규기저 타입 II.

3. 최적정규기저를 이용한 $GF(2^m)$상의 곱셈 알고리즘

두 원소의 곱 $C = AB = \sum_{s=0}^{m-1} c_s\alpha_s$는

$$\begin{aligned} C &= \sum_{i,j} a_i b_j \alpha_i\alpha_j \\ &= \sum_{i,j} a_i b_j \sum_{s=0}^{m-1} \lambda_{ij}^{(s)} \\ &= \sum_{s=0}^{m-1} \left(\sum_{i,j} a_i b_j \lambda_{ij}^{(s)}\right)\alpha_s \end{aligned} \quad (12)$$

이다. 이 때, a, b, λ의 아래 첨자는 mod m에 기약된다. 그러므로 $(\lambda_{ij})^{(s)} = (\lambda_{i-1,j-1})^{(s-1)}$인 성질을 이용하여 C의 계수 c_s는 다음과 같이 표현할 수 있다.

$$\begin{aligned} c_s &= \sum_{i,j} a_i b_j \lambda_{ij}^{(s)} \\ &= \sum_{i,j} a_i b_j \lambda_{i-s,j-s}^{(0)} \\ &= \sum_{i,j} a_{i+s} b_{j+s} \lambda_{ij}^{(0)} \\ &= \sum_{j=0}^{m-1} \left(\sum_{i=0}^{m-1} a_{i+s}\lambda_{i,j}^{(0)}\right) b_{j+s} \end{aligned} \quad (13)$$

이 때, 대응되는 행렬 $X = (x_{st})$에 대해 $GF(2)$상의 원소 x_{st} ($0 \le s, t \le m-1$)을 다음 식과 같이 정의하면,

$$x_{st} = \left(\sum_{i=0}^{m-1} a_{i+s}\lambda_{i,j}^{(0)}\right) b_{t+s} \quad (14)$$

이고 행렬 X의 t번째 열벡터 X_t는 $X_t = (x_{0t}, x_{1t}, \cdots, x_{m-1,t})^T$이다. 또한, $c_s = \sum_{t=0}^{m-1} x_{st}$이므로 모든 열벡터 X_t ($0 \le t \le m-1$)의 합은 $(c_0, c_1, \cdots, c_{m-1})^T$와 같다. 열벡터 X_t의 재배열과 계산 과정의 부분합 신호를 재사용함으로써 공

간 복잡도 및 최대 처리기 지연 시간을 줄일 수 있다.

3.1 최적정규기저 타입 I의 열벡터 치환

$m=2v$라 하고 $Y=(y_{st})$를 X의 열벡터 치환에 의한 $m\times m$ 행렬로 정의하면 Y는 다음 식 (15)와 같이 정의된다.

$$\begin{aligned}
&\text{1st}\quad \text{3rd}\quad \text{5th}\quad \cdots\quad \text{2nd}\quad \text{4th}\quad \text{6th}\quad \cdots\\
&0\quad 1\quad 2\quad \cdots\quad \frac{m}{2}\quad \frac{m}{2}+1\quad \frac{m}{2}+2\quad \cdots\\
&(X_v,\ X_{v-u},\ X_{v-2u},\cdots,\ X_{v''},\ X_{(v-u)''},\ X_{(v-2u)''},\cdots),\\
&u=\begin{cases}\lceil \frac{v}{2} \rceil & v=\text{odd},\\ v-1 & v=\text{even},\end{cases}\\
&2^{t'}\equiv m+2^t \pmod{m+1},\ t''=t'-\frac{m}{2}\pmod{m}.
\end{aligned} \tag{15}$$

여기서, u는 gcd(m, u)=1인 가장 작은 소수로 정한다. 또한, $Y_t=(y_{0t},\ y_{1t},\ \cdots,\ y_{m-1,t})^T$인 Y의 모든 열벡터 Y_t, ($0\le t\le m-1$)의 합은 $(c_0,\ c_1,\ \cdots,\ c_{m-1})^T$인 X의 모든 열벡터 X_t, $0\le t\le m-1$의 합과 같다. 따라서 패러럴 입·출력 곱셈기를 설계하기 위해 Y의 열벡터 합을 계산하는 대신 Y의 순환 쉬프트된 대각 벡터의 합을 계산한다. 즉, 행렬 Y의 표현에서 열벡터 X_t와 $X_{t''}$ 사이에 정확히 $m/2$개의 열이 존재한다. 또한, 열벡터 X_0와 X_{m-1}을 제외한 나머지($t=0$에 대응하는 $t''=m-1$이다.) 열벡터에서 X_t의 s번째 원소와 $X_{t''}$의 $s+m/2$번째 원소는 그들의 가수(summand)에서 동일한 a_{i+s}를 가진다. 다시 말해, 식 (14)로부터 다음 식을 얻을 수 있다.

$$\begin{aligned}
x_{s+\frac{m}{2},t''}&=(\sum_{i=0}^{m-1}a_{i+s+\frac{m}{2}}\lambda_{i,t''}^{(0)})b_{t''+s+\frac{m}{2}}\\
&=(\sum_{i=0}^{m-1}a_{i+s}\lambda_{i,t}^{(0)})b_{t+s}
\end{aligned} \tag{16}$$

따라서 $x_{s,t}$와 $x_{s+\frac{m}{2},t''}$은 같은 항 $\sum_{i=0}^{m-1}a_{i+s}\lambda_{i,t}^{(0)}$을 가진다. 이는 결국 $C=AB$의 계산에 있어 부분합 신호의 재사용을 의미한다. 지금까지 설명한 내용을 바탕으로 다음과 같은 최적정규기저 타입 I을 이용한 $GF(2^m)$상의 곱셈 알고리즘 1을 얻을 수 있다.

알고리즘 1. 최적정규기저를 이용한 $GF(2^m)$상의 비트-레벨 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

3.2 최적정규기저 타입 II의 열벡터 치환

$m-1=2v$라 하고 $Y=(y_{st})$를 X의 열벡터 치환에 의한 $m\times m$ 행렬로 정의하면 v가 홀수일 때 Y는 식 (17)과 같이 정의되고,

$$\begin{aligned}
&(X_\nu,\cdots,X_3,X_1,X_{m-1},X_{m-3},\cdots,\\
&X_{m-\nu},X_{\nu-1},\cdots,X_2,X_0,X_{m-2},\cdots,X_{m-\nu+1})
\end{aligned} \tag{17}$$

v가 짝수일 때, Y는 식 (18)과 같이 각각 정의된다.

$$\begin{aligned}
&(X_\nu,\cdots,X_2,X_0,X_{m-2},\cdots,X_{m-\nu},X_{\nu-1},\cdots,\\
&X_3,X_1,X_{m-1},X_{m-3},\cdots,X_{m-\nu+1})
\end{aligned} \tag{18}$$

여기서, $Y_t=(y_{0t},\ y_{1t},\ \cdots,\ y_{m-1,t})^T$인 Y의 모든 열벡터 Y_t, ($0\le t\le m-1$)의 합은 $(c_0,\ c_1,\ \cdots,\ c_{m-1})^T$인 X의 모든 열벡터 X_t, $0\le t\le m-1$의 합과 같다. 따라서 패러럴 입·출력 곱셈기를 설계하기 위해 Y의 열벡터 합을 계산하는 대신 Y의 순환 쉬프트된 대각 벡터의 합을 계산한다. 즉, 행렬 Y의 표현에서 벡터 X_t와 X_{m-t} 사이에 정확히 $t-1$개의 열이 존재한다. 또한, 열벡터 X_t의 s번째 원소와 X_{m-t}의 $s+t$번째 원소는 그들의 가수(summand)에서 동일한 a_{i+s}를 가진다. 다시 말해, 식 (14)로부터 다음 식을 얻을 수 있다.

$$\begin{aligned}
x_{s+t,m-t}&=(\sum_{i=0}^{m-1}a_{i+s+t}\lambda_{i,-t}^{(0)})b_s\\
&=(\sum_{i=0}^{m-1}a_{i+s}\lambda_{i-t,-t}^{(0)})b_s\\
&=(\sum_{i=0}^{m-1}a_{i+s}\lambda_{i,t}^{(0)})b_s
\end{aligned} \tag{19}$$

따라서 $x_{s,t}$와 $x_{s+t,n-t}$는 같은 항 $\sum_{i=0}^{m-1}a_{i+s}\lambda_{i,t}^{(0)}$을 가진다. 이는 결국 $C=AB$의 계산에 있어 부분합 신호의 재사용을 의미한다. 지금까지 설명한 내용을 바탕으로 최적정규기저 타입 II을 이용한 $GF(2^m)$상의 곱셈 알고리즘에서도 재배열 규칙만 다를 뿐, 최적정규기저 타입 I의 경우와 같은 알고리즘 1을 얻을 수 있다.

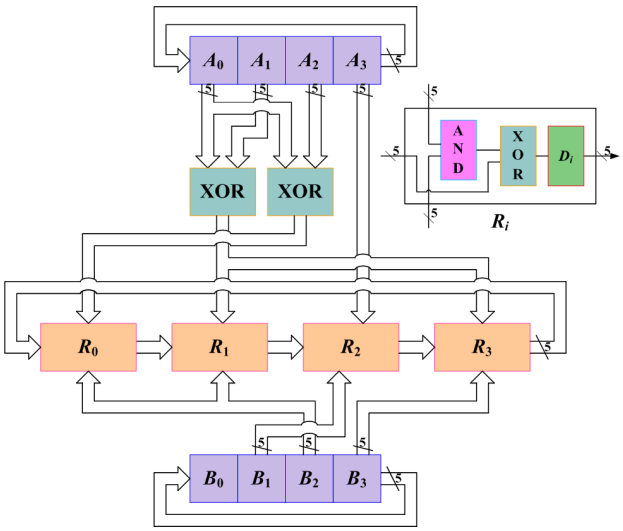
4. 최적정규기저를 이용한 합성체 $GF(2^{mn})$상의 곱셈기

앞에서 언급한 최적정규기저 타입 I, II를 이용하여 제시된 3가지 합성체 중 합성체 타입 I에 대한 곱셈기를 하드웨어로 구현한다. 합성체 타입 I에서 언급했듯이 부분체 $GF(2^m)$상의 곱셈기는 최적정규기저 타입 II을 이용한 곱셈기이고 $GF(2^{mn})$상의 곱셈기는 $GF(2^m)$에 대한 최적정규기저 타입 I을 이용한다. 본 논문에서는 합성체 $GF(2^{5\times4})$상의 곱셈기를 예를 들어 설명할 것이다.

합성체 $GF(2^{5\times4})$의 경우 부분체 $GF(2^5)$은 최적정규기저 II가 존재하며 부분체 $GF(2^5)$에 대한 확장체로 $GF(2^4)$을 간주할 수 있어 최적정규기저 타입 I이 존재한다. 따라서 합성체 $GF(2^{5\times4})$상의 곱셈 연산을 수행하기 위해 최적정규기저 타입 I을 이용한 $GF(2^4)$상의 곱셈기를 이용하며 이 때 부분체 $GF(2^5)$상의 연산이 수행되야 한다. 따라서 클럭당 5-비트의 연산을 수행해야 하며 이를 (그림 1)과 같이 나타낼 수 있다.

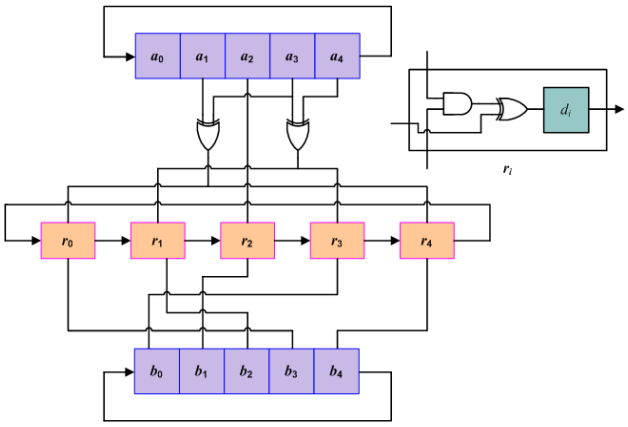
(그림 1)에서 보듯이 레지스터 A_i, B_i, R_i는 5-비트씩 저장하며 5-비트씩 쉬프트시킨다. 특히 R_i는 $GF(2^5)$상의 연산을 수행하게 된다. 따라서 레지스터 R_i 외부의 XOR 셀과 레지스터 R_i 내부의 XOR 셀은 두 개의 5-비트 입력 값을 받아 각각 XOR 게이트를 이용하여 출력하게 된다.

또한 레지스터 D_i 셀은 5-비트의 입력값을 받아서 저장하는 역할을 수행한다. 이와 달리 레지스터 R_i의 AND 셀은 두 개의 5-비트 입력값을 받아 $GF(2^5)$상의 곱셈 연산을



(그림 1) 합성체 $GF(2^{5\times4})$상의 곱셈기

수행한 결과값을 출력해야 한다. 따라서 최적정규기저 타입 I을 이용한 $GF(2^5)$상의 곱셈기를 이용하는데 다음 (그림 2)와 같이 나타낼 수 있다.



(그림 2) AND 셀 - 최적정규기저 타입 II 곱셈기

(그림 2)에서는 (그림 1)과 달리 레지스터 a_i, b_i, r_i는 1-비트씩 저장 및 쉬프트시키며 XOR 게이트와 AND 게이트는 2개의 1-비트 값을 입력받아 1-비트 값을을 출력한다.

5. 성능분석

본 논문에서 제안한 합성체 $GF(2^{mn})$상의 곱셈기는 최적정규기저를 이용하여 $GF(2^m)$을 원소로 가지는 $GF(2^n)$상의 곱셈 연산을 수행하며 각각의 레지스터 R_i 내부의 AND 셀에서 부분체 $GF(2^m)$상의 곱셈 연산을 수행한다. (그림 1)에서 보듯이 합성체 타입 I의 경우, XOR 셀은 $3n/2$개 존재한다. 이 XOR 셀은 2개의 m-비트 입력값을 XOR 연산하므로 각 XOR 셀은 단지 m개의 XOR 게이트로 구성되어있다. 따라서 레지스터 R_i의 외부에 있는 XOR 셀의 처리지연시간은 T_X(T_X : XOR 게이트의 처리시간)이고 레지스터 R_i의 내부에 있는 XOR 셀의 처리지연시간 또한 T_X이다. 레지스터 R_i의 AND 셀은 $GF(2^m)$상의 곱셈 연산을 수행하기 위한 곱셈기로써 합성체 타입 I인 경우이므로 최적정규기저 타입 II를 이용한 곱셈기이다. 따라서 각 AND 셀은 $(3m-1)/2$개의 XOR 게이트와 m개의 AND 게이트를 가지며 처리지연시간은 T_A+2T_X.(T_A : AND 게이트의 처리시간)이다. 따라서 합성체 타입 I에 대한 곱셈기에 대해 살펴보면 최대처리기지연시간은 $T_A+(1+1+2)T_X$이고 AND 게이트는 $m\times n$개 존재하며 XOR 게이트는 $m\times(3n/2)+n\times(3m-1)/2=n(6m-1)/2$개 존재한다.

<표 1>은 기존의 합성체상의 곱셈기와 제안된 곱셈기를 비교하였다. 표에서 나타나듯이 제안된 합성체상의 곱셈기는 기존의 곱셈기보다 작은 하드웨어 복잡도와 처리기 지연시간을 가짐을 알 수 있다.

<표 1> 합성체상의 곱셈기 비교

회로\항목	최대처리기 지연시간	AND	XOR
Oh[5]	$T_A+(2+\lceil \log_2 m \rceil +\lceil \log_2(n-1) \rceil)T_X$	m^2n^2	$m(2mn^2-n^2-1)$
제안된 곱셈기	T_A+4T_X	mn	$\frac{n(6m-1)}{2}$

6. 결론

본 논문에서는 유한체 산술 연산 중 매우 복잡한 곱셈 연산시 최적정규기저를 이용한 합성체상의 곱셈기를 제안하였다. 앞에서 언급했듯이 정규기저는 모든 유한체 $GF(2^m)$에 대해 항상 존재하지만 최적정규기저가 존재하는 유한체는 제한적이다. 만약 유한체 $GF(2^{5\times4})$의 경우 최적정규기저가 존재하지 않지만 부분체와 확장체간의 관계를 이용하여 최적정규기저를 이용한 효율적인 곱셈기를 구현할 수 있다. 또한 ECC에서의 곱셈 연산시 합성체상의 곱셈기를 이용하거나 pairing 기반 암호시스템에서 표수가 2인 경우 embedding degree가 4이므로 합성체 타입 I에 대한 곱셈기를 이용할 수 있다. 따라서 제안된 곱셈기는 ECC 및 pairing 기반 암호시스템에서의 곱셈기로 매우 적합하다.

참고문헌

[1] A. J. Menazes, I. F. Blake, S. Gau, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian, *Applications of Finite Fields*, Kluwer Academic Publisher, 1993.

[2] J. L. Massey and J. K. Omura, "Computational Method and Apparatus for Finite Field Arithmetic," *US Patent* No. 4587627, 1986.

[3] S. Kwon and H. Ryu, "Efficient Bit Serial Multiplication Using Optimal Normal Bases of Type II in $GF(2^m)$," *ISC 02, Lecture Notes in Computer Science*, Vol. 2433, pp. 353-356, 1998.

[4] 권윤기, 김창훈, 권순학, 황병곤, "최적정규기저 타입 I을 이용한 $GF(2m)$상의 효율적인 비트-시리얼 곱셈기", 2008년도 한국멀티미디어학회 추계학술발표대회, Vol. 11, No. 2, pp. 136, 2008.

[5] S. Oh, C. H. Kim, J. Lim, and D. H. Cheon, "Efficient Normal Basis Multipliers in Composite Fields," *IEEE Trans. Computers*, Vol. 49, No. 10, pp. 1133-1138, 2000.