情報保護學會論文誌
第18卷 第2號, 2008. 4

Equally Spaced 기약다항식 기반의 효율적인 이진체 비트-병렬 곱셈기*

이 옥 석[1†], 장 남 수[1], 김 창 한[2‡], 홍 석 희[1]
[1]고려대학교 정보경영공학전문대학원, [2]세명대학교

Efficient Bit-Parallel Multiplier for Binary Field Defined by Equally-Spaced Irreducible Polynomials*

Ok Suk Lee[1†], Nam Su Chang[1], Chang Han Kim[2‡], Seokhie Hong[1]
[1]Graduate School of Information Management and Security, Korea University, [2]Semyung University

요 약

유한체 GF(2^m)의 원소를 표현하기 위한 기저선택은 곱셈기의 효율성에 영향을 미친다. 이중에서 여분표현을 이용한 곱셈기는 모듈러 감산을 빠르게 구성할 수 있는 특징을 이용하여 시간-공간의 trade-off를 효율적으로 제공한다. 따라서 여분표현을 이용한 기존의 곱셈기는 다른 기저로 표현한 곱셈기보다 시간 복잡도 상의 효율성을 제공하나 공간 복잡도가 많이 늘어나는 단점을 가진다. 본 논문에서는 다항식 지수승 연산이 많이 사용된다는 것을 감안해 Left-to-Right 형태의 지수승 환경에 적합한 시간-공간 복잡도 상의 효율성을 가지는 새로운 비트-병렬 곱셈기를 제안한다. 제안하는 곱셈기는 $T_A+(\lceil \log_2 m \rceil)T_X$ 시간 복잡도 와 $(2m-1)(m+s)$ 공간 복잡도를 요구하며 ESP(Equally Spaced Polynomial) 기약다항식 기반의 기존 여분표현 곱셈기와 비교해 공간 복잡도는 $2(ms+s^2)$감소하며, 시간복잡도는 $T_A+(\lceil \log_2(m+s) \rceil)T_X$에서 $T_A+(\lceil \log_2 m \rceil)T_X$로 감소된다.(T_A:2개의 입력에 1개의 출력인 ***AND*** 게이트 시간, T_X:2개의 입력에 1개의 출력인 ***XOR*** 게이트 시간이며 m:ESP기약 다항식 차수, s: ESP기약 다항식의 각항의 차수 간격)

ABSTRACT

The choice of basis for representation of element in GF(2^m) affects the efficiency of a multiplier. Among them, a multiplier using redundant representation efficiently supports trade-off between the area complexity and the time complexity since it can quickly carry out modular reduction. So time of a previous multiplier using redundant representation is faster than time of multiplier using others basis. But, the weakness of one has a upper space complexity compared to multiplier using others basis. In this paper, we propose a new efficient multiplier with consideration that polynomial exponentiation operations are frequently used in cryptographic hardwares. The proposed multiplier is suitable for left-to-right exponentiation environment and provides efficiency between time and area complexity. And so, it has both time delay of $T_A+(\lceil \log_2 m \rceil)T_X$ and area complexity of $(2m-1)(m+s)$. As a result, the proposed multiplier reduces $2(ms+s^2)$ compared to the previous multiplier using equally-spaced polynomials in area complexity. In addition, it reduces $T_A+(\lceil \log_2 m+s \rceil)T_X$ to $T_A+(\lceil \log_2 m \rceil)T_X$ in the time complexity.(T_A:Time delay of one ***AND*** gate, T_X:Time delay of one ***XOR*** gate, m:Degree of equally spaced irreducible polynomial, s:spacing factor)

Keywords : bit-parallel multiplier, redundant representation, Left-to-Right exponentiation, Equally spaced irreducible polynomial

Ⅰ. 서 론

유한체 $GF(2^m)$상의 연산은 코딩이론(coding theory), 컴퓨터 대수(computer algebra), 공개키 암호(public key cryptosystem)중 타원곡선 암호(elliptic curve cryptosystem)등 여러 분야에서 널리 쓰이고 있다[1-3]. $GF(2^m)$의 주요연산은 덧셈, 곱셈, 제곱, 역원 연산들이다. 이 중에서 덧셈과 제곱 연산은 단순한 과정으로 수행되지만, 곱셈과 역원 연산은 덧셈과 제곱 연산에 비해 복잡한 연산 과정을 필요로 한다. 특히, 공개키 암호 시스템에서 가장 많은 비중을 차지하는 지수승 연산의 경우 반복적인 곱셈으로 구성되므로 곱셈기의 효율성이 공개키 암호 시스템의 효율성에 크게 영향을 미친다.

$GF(2^m)$곱셈기 설계에 있어서 원소 표현은 매우 중요하다. 주로 쓰는 원소표현 방법은 정규기저 표현(normal basis), 다항식기저 표현(polynomial basis), 쌍대기저 표현(dual basis), 여분 표현(redundant representation) 등이 있다. 이들 각각은 환경에 따라 장단점을 가지며 이중에서 여분 표현을 사용하는 경우 다른 기저들과 비교해 모듈러 감산부분을 빠르게 구성할 수 있기 때문에 시간-공간 복잡도의 trade-off를 이용해 효율성을 제공한다[8-13]. 다항식 기저에서 m차이며 항의 차수 차이가 s인 Equally Spaced 기약 다항식(ESP) $p(x)$ 의 경우 (즉, $p(x)=\sum_{i=0}^{n} x^{is}$) 복잡도는 m^2 ***AND*** 게이트, m^2-s XOR 게이트의 공간 복잡도와 $T_A+(1+\lceil \log_2 m \rceil)T_X$의 시간 복잡도가 요구된다[4-7] (여기서, T_A:2개의 입력에 1개의 출력인 ***AND*** 게이트 시간, T_X:2개의 입력에 1개의 출력인 XOR 게이트 시간). 그러나 연산 시간을 줄이기 위해 여분 표현을 적용하면 $(m+s)^2$***AND*** 게이트, $(m+s-1)(m+s)$ ***XOR*** 게이트의 공간 복잡도와 $T_A+(\lceil \log_2 m+s \rceil)T_X$의 시간 복잡도를 가진다[10-13]. 따라서 s가 최소인 AOP(All One Polynomial)가 이중에서 가장 효율적임을 알 수 있다[8],[13].

본 논문에서는 Left-to-Right 지수승 환경에서 고정된 다항식의 곱을 반복적으로 수행한다는 것을 착안하여 여분 표현으로 표현한 원소와 다항식 기저로 표현한 원소의 곱셈을 수행하는 곱셈기를 제안한다.

제안한 곱셈기는 일반적인 ESP에서 여분 표현을 사용한 기존의 곱셈기와 다르게 하나의 입력 값이 다항식 기저로 표현되기 때문에 감산하는 부분이 적어져 기존 여분 표현의 곱셈기와 비교하여 시간-공간 복잡도면에서 모두 효율성을 제공한다.

본 논문의 구성은 다음과 같다. 2절에서는 여분표현 방법에 대한 개요를 바탕으로 여분 표현을 이용한 기존 곱셈기와 원소표현 변환에 대하여 소개한다. 3절에서는 여분표현을 이용한 제안하는 곱셈기를 기술하고, 4절에서 ESP기약 다항식에서 정의된 기존 곱셈기들과 제안하는 곱셈기의 효율성을 비교한다.

Ⅱ. 여분표현의 개요

본 절에서는 여분 표현을 이용한 기존 곱셈기를 소개하며, 여분표현을 다항식기저 표현으로 변환하는 과정도 살펴본다. 우선, 소개에 앞서 유한체 $GF(2^m)$을 생성하는데 이용될 ESP 기약 다항식을 다음과 같이 정의한다.

GF(2)에서 차수가 m이고 항사이의 차가 s인 다항식 $p(x)=\sum_{i=0}^{n} x^{is}$를 ESP(Equally-Spaced Polynomial)라 한다(이때 m=ns). 그리고 α를 GF(2)위에서 차수가 m인 ESP의 근이라 하자. 그러면, $\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$은 다항식 기저를 이루고, $GF(2^m)$의 모든 원소는 $1, \alpha, \alpha^2, \cdots, \alpha^{m-1}$의 선형 결합으로 표현된다.

또한, ESP $p(x)$ 다항식의 경우 $\alpha^{m+s}=1$이므로 α^{m+s+i}차 항 이후의 곱셈 결과는 α^i차 항 이후로 더해진다. 그리고 α^m차 항부터 α^{m+s-1}차 항까지의 감산시 발생하는 시간 복잡도를 줄이기 위해, 다항식기저$\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$를 확장한 $\{1, \alpha, \alpha^2, \cdots, \alpha^{m+s-1}\}$을 고려한다. 그러면, $GF(2^m)$에 있는 임의의 원소는 $a=\sum_{i=0}^{m+s-1} a_i \alpha^i (a_i \in GF(2), 0 \le i \le m+s-1)$ 로 표현된다. 이와 같이 유한체 원소 표현에 원래 기저에 항을 추가해서 나타내는 방법을 여분 표현이라 한다. 따라서, 본 논문에서는 기약인 ESP에 의해 정의된 $GF(2^m)$에서의 효율적인 비트-병렬 곱셈기를 제안하기 위해 여분 표현의 이러한 이점을 사용한다.

2.1. 여분 표현을 이용한 기존 곱셈기

α가 ESP $p(x)$의 근이라 하면, $p(\alpha)=0$ 이다. 따라서

접수일: 2007년 11월 8일; 채택일: 2007년 12월 28일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2007-(C1090-0701-0025))
† 주저자, oslee@cist.korea.ac.kr
‡ 교신저자, chkim@semyung.ac.kr

유한체 $GF(2^m)$에 있는 임의의 원소 a와 b는 $a=\sum_{i=0}^{m+s-1} a_i \alpha^i$, $b=\sum_{j=0}^{m+s-1} b_j \alpha^j$ 로 표현된다 (단, $a_i, b_j \in GF(2)$). 두 원소의 곱셈 $a \cdot b$을 수행하는 연산 과정에는 다음과 같은 두 가지 방법이 있다. 첫 번째 방법은 곱셈단계와 모듈러 감산단계를 분리해서 계산하는 일반적인 방법이며, 두 번째 방법은 곱셈단계와 모듈러 감산을 하나의 행렬로 구성해 처리하는 마스트로 비토 곱셈방법이다. 이중 두 번째인 마스트로 비토 곱셈방법의 경우 병렬 처리가 가능해 고속 연산에 적합하다 [4-6]. 따라서 유한체 $GF(2^m)$에 대한 고속 병렬 곱셈기를 설계하기 위해 이 방법을 사용한다. 그러면 계산 과정은 다음과 같다. 우선, a와 b를 $GF(2^m)$의 임의의 두 원소라 하자. a와 b는 $a=\sum_{i=0}^{m+s-1} a_i \alpha^i$, $b=\sum_{j=0}^{m+s-1} b_j \alpha^j$로 표현된다. 이제 두 원소를 단순히 곱하는 과정을 식으로 나타내면 다음과 같다.

$$\begin{aligned} c &= a \cdot b \\ &= \left(\sum_{i=0}^{m+s-1} a_i \alpha^i\right) \cdot \left(\sum_{j=0}^{m+s-1} b_j \alpha^j\right) \\ &= \sum_{i=0}^{2m-2+2s} c_i \alpha^i. \end{aligned}$$

이제 위의 다항식 표현의 곱셈을 나타내기 위해 다음과 같이 행렬 M을 정의한다.

$$M=\begin{pmatrix} a_0 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ a_{m-1} & \cdots & \cdots & a_0 & 0 & 0 \\ \vdots & \ddots & & & \vdots & \vdots \\ a_{m+s-1} & \cdots & & \cdots & a_1 & a_0 \\ 0 & a_{m+s-1} & \cdots & & a_2 & a_1 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & a_{m+s-1} & a_{m+s-2} \\ 0 & 0 & 0 & 0 & 0 & a_{m+s-1} \end{pmatrix}.$$

그러면 정의된 행렬 M을 이용해 다항식 곱셈을 다음과 같은 행렬의 표현식으로 나타낼 수 있다[4-6].

$$\vec{c}=M \cdot \vec{b}$$

이며, 이때, $M:(2m-1+2s)\times(m+s)$ 행렬, $\vec{c}=(c_0, c_1 \cdots, c_{2m-2+2s})^T$, $\vec{b}=(b_0, \cdots, b_{m+s-1})^T$ 열벡터이다. 이제, $\vec{c}$을 여분 표현으로 나타내기 위해서 다음과 같은 두 가지 과정이 필요하다.

2.1.1 행렬 감산 과정

$\alpha^{m+s}=1$임을 이용하면 다항식 c의 $m+s+i$ 차 항 이후가 i차 항 이후로 더해진다. 이 과정을 위의 행렬 표현으로 보면 행렬 M의 $(m+s+i)$행 이후가 i행 이후로 더해진다. 따라서 행렬 M의 특성상 추가적인 ***XOR*** 게이트 없이 행렬 $M:(2m-1+2s)\times(m+s)$은 행렬 $M':(m+s)\times(m+s)$으로 구성할 수 있다[4],[5],[6].

$$M'=\begin{bmatrix} a_0 & a_{m+s-1} & \cdots & & a_{s+1} \\ a_1 & a_0 & a_{m+s-1} & \cdots & a_{s+2} \\ \vdots & & \ddots & & \\ a_{m-2} & \cdots & & a_0 & a_{m+s-1} \\ a_{m-1} & \cdots & & & a_0 \\ \vdots & & & & \\ a_{m+s-1} & \cdots & & & a_s \end{bmatrix}.$$

2.1.2 행렬 곱셈 과정

다항식 c을 $p(\alpha)$로 감산한 다항식을 c'로 정의한다. 그러면, 다항식 $p(\alpha)$로 감산한 과정은 행렬 감산 과정과 관련되어 있으며, 위에서 정의된 M'을 이용해 모듈러 감산을 다음과 같은 행렬의 표현식으로 나타낼 수 있다 [4],[5],[6].

$$\vec{c}=M' \cdot \vec{b}$$

이며, 이때, $M':(m+s)\times(m+s)$ 행렬, $\vec{c}=(c'_0, c'_1, \cdots, c'_{m+s-1})^T$, $\vec{B}=(b_0, \cdots, b_{m+s-1})^T$.

이제까지 살펴본 여분 표현의 기존 곱셈기의 효율성은 다음과 같다. 행렬 M'에 열벡터 $\vec{b}$을 곱할 때 각 행마다 $(m+s)$개의 곱과 합이 수행되고 총 행의 개수가 $(m+s)$개 이므로, $(m+s)^2$ ***AND*** 게이트, $(m+s-1)(m+s)$ ***XOR*** 게이트의 공간 복잡도와 $T_A+(\lceil \log_2(m+s) \rceil)T_X$ 시간 복잡도가 소요된다.

2.2. 원소표현 변환

유한체 원소 표현에는 다항식기저 표현, 정규기저 표현, 쌍대기저 표현, 여분 표현 등 여러 가지 종류가 있다. 이들은 각각의 환경에 따라 장단점을 가진다. 따라서 연산을 수행할 때 그 연산에 장점을 갖는 원소 표현으로 변환하는 일은 필수적이다. 본 논문에서 제안한 곱

셈기는 ESP 기약 다항식의 특성 $\alpha^{m+s}=1$을 고려해 여분 표현을 선택한다. 그리고 연산 결과를 우리가 원하는 다항식 기저로 표현하려면 α^{m}차 항부터 α^{m+s-1}차 항까지를 $p(x)$로 다음과 같이 감산하며 상세한 설명은 다음과 같다.

$$a=\sum_{i=0}^{m+s-1} a_i\alpha^i=\sum_{i=0}^{m-1} a_i\alpha^i+\sum_{i=m}^{m+s-1} a_i\alpha^i$$

$$=\sum_{i=0}^{m-1} a_i\alpha^i+\sum_{i=0}^{s-1} a_{m+i}\alpha^{m+i} \quad (1)$$

$$=\sum_{i=0}^{m-1} a_i\alpha^i+\sum_{i=0}^{s-1} a_{m+i}(\alpha^{m-s+i}+\cdots+\alpha^{s+i}+\alpha^i) \quad (2)$$

$$=\sum_{i=0}^{m-1} a_i\alpha^i+(\sum_{i=0}^{s-1} a_{m+i}\alpha^i+\sum_{i=s}^{2s-1} a_{m-s+i}\alpha^i+\cdots+\sum_{i=m-s}^{m-1} a_{s+i}\alpha^i) \quad (3)$$

수식 (1)은 α^{m+i}항 이후를 감산하기 위한 유도 과정을 나타내며, 수식 (2)은 정의된 기약 다항식의 $p(x)$근이 α 이므로 $\alpha^m=\alpha^{m-s}+\cdots+\alpha^s+1$라는 특성에 의해 유도된 과정이다. 마지막으로 수식(3)은 인덱스 맞춰 정리한 식을 나타낸다.

따라서, 유한체 $GF(2^m)$원소를 여분 표현에서 다항식 기저 표현으로 변환하는데 다음과 같은 복잡도가 요구된다. 공간 복잡도의 경우 s개의 항이 순서대로 n개씩 더해지므로 m ***XOR*** 게이트, 시간 복잡도의 경우 더해지는 부분이 한 번씩 병렬로 구성되므로 $1T_X$ 시간 지연이 일어난다.

Ⅲ. 여분 표현을 이용한 제안하는 효율적인 비트-병렬 곱셈기

본 장에서는 여분 표현을 이용한 기존 곱셈기와 다른 형태의 효율적인 비트-병렬 곱셈기를 소개한다. 본 곱셈기는 다항식 지수승 연산이 많이 쓰인다는 것을 감안한 Left-to-Right 지수승 환경에 적합한 곱셈기로 고정된 형태의 인자가 반복적으로 곱해지는 점을 착안하여 고정된 형태의 입력 값을 다항식 기저로 표현한 원소를 다른 하나의 입력 값은 여분 표현을 이용한 원소를 곱하는 형태이다. 임의의 원소 a, b가 각각 여분 표현과 다항식 기저로 표현된다면, $a=\sum_{i=0}^{m+s-1} a_i\alpha^i$, $b=\sum_{j=0}^{m-1} b_j\alpha^j$ $(a_i, b_j\in GF(2))$이다. 두 원소 a와 b을 곱하는 과정을 식으로 나타내면 다음과 같다.

$$c=a\cdot b$$
$$=(\sum_{i=0}^{m+s-1} a_i\alpha^i)\cdot(\sum_{j=0}^{m-1} b_j\alpha^j)$$
$$=\sum_{i=0}^{2m-2+s} c_i\alpha^i.$$

이제 위의 다항식 곱셈과정을 표현하기 위해 다음과 같이 행렬 M을 정의한다.

$$M=\begin{pmatrix} a_0 & 0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ a_{m-1} & \cdots & \cdots & a_1 & a_0 \\ \vdots & \ddots & & & \vdots \\ a_{m+s-1} & \cdots & & a_{s+1} & a_s \\ 0 & a_{m+s-1} & \cdots & a_{s+2} & a_{s+1} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & a_{m+s-1} & a_{m+s-2} \\ 0 & 0 & 0 & 0 & a_{m+s-1} \end{pmatrix}.$$

그러면 정의된 행렬 M을 이용해 다항식 곱셈을 다음과 같은 행렬의 표현식으로 나타낼 수 있다[4],[5],[6].

$$\vec{c}=M\cdot\vec{b}$$

이고, $\vec{c}=(c_0, c_1, \cdots, c_{2m-2+s})^T$, $\vec{b}=(b_0, \cdots, b_{m-1})^T$ 열벡터이며, $\vec{c}$는 c의 다항식 계수로 이루어진 $(2m-1+s)\times 1$인 열벡터이고, $\vec{b}$는 b의 다항식 계수로 이루어진 $m\times 1$인 열벡터이다. 또한, 정의한 행렬 M은 $(2m-1+s)\times m$인 행렬이고 a라는 다항식 계수로 이루어져 있다.

열벡터 $\vec{c}$를 여분표현으로 나타내기 위한 감산 과정은 기약 ESP $p(x)$와 연관되어 나타난다.

$c'=\sum_{i=0}^{m+s-1} c'_i\alpha^i$ $(c'_i\in GF(2))$로 표현된다. 여기서, c'는 기약 ESP $p(x)$의 특성인 $\alpha^{m+s}=1$을 이용한 결과 값이다(α: $p(x)$의 근).

이제, 위의 식을 행렬로 표현하기 위해서 다음과 같은 두 가지 과정을 거친다.

3.1. 행렬 감산 과정

$\alpha^{m+s}=1$인 특성을 이용해 감산하면 다항식 c의 $m+s+i$ 차 항 이후가 각각 i차 항 이후로 더해진다. 그러면, 행렬 M은 추가적인 ***XOR*** 게이트 없이 행렬

$M'(:(m+s)\times m)$으로 구성 할 수 있고,변화 과정은 다음과 같다[4],[5],[6].

$$M[m+s+i \rightarrow i] \quad (i=1,\cdots,m-1).$$

여기서, $M[i \rightarrow j]$는 i행 구성 원소을 j행의 구성 원소와 ***XOR*** 연산하여 j행으로 이동 시키는 것으로 정의한다. 그러면, 위의 변화 과정으로 $M((2m-1+s)\times m$인 행렬) 에서 $M'((m+s)\times m$행렬)으로 바뀐다.

$$M' = \begin{bmatrix} a_0 & a_{m+s-1} & \cdots & & a_{s+1} \\ a_1 & a_0 & a_{m+s-1} & \cdots & a_{s+2} \\ \vdots & & \ddots & & \\ a_{m-2} & \cdots & & a_0 & a_{m+s-1} \\ a_{m-1} & \cdots & & & a_0 \\ \vdots & & & & \\ a_{m+s-1} & \cdots & & & a_s \end{bmatrix}$$

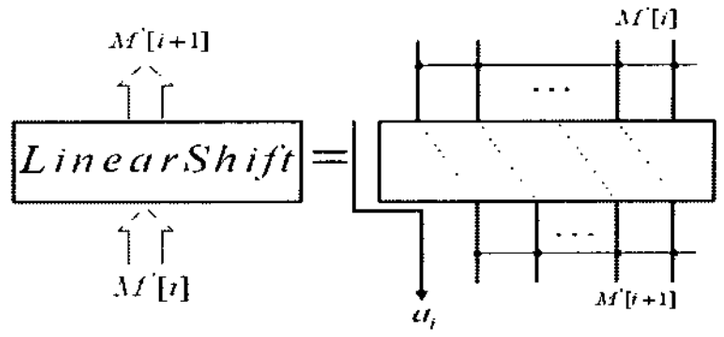
3.2. 행렬 곱셈 과정

여분표현으로 나타낸 다항식 c은 행렬 감산 과정에서 얻은 행렬 M'을 이용해 모듈러 감산을 다음과 같은 행렬의 표현식으로 나타낼 수 있다[4],[5],[6].

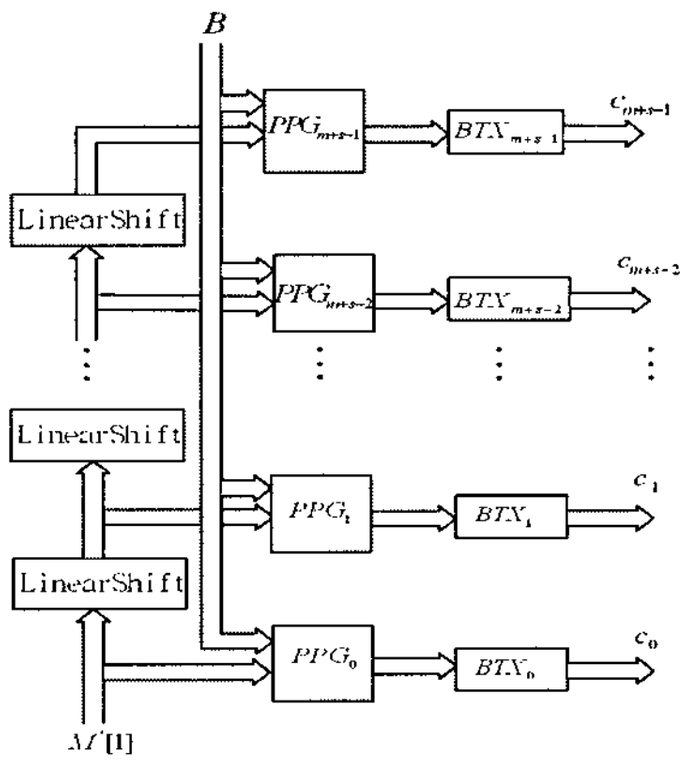
$$\vec{c} = M' \cdot \vec{b}$$

이고, $M':(m+s)\times m$ 행렬, $\vec{c}=(c_0,c_1,\cdots,c_{m+s-1})^T$, $\vec{b}=(b_0,\cdots,b_{m-1})^T$ 이다. 이때, $\vec{c}$은 c'의 다항식 계수로 이루어진 $(m+s)\times 1$인 열벡터, $\vec{b}$는 b의 다항식 계수로 이루어진 $m\times 1$인 열벡터이다.

따라서, 제시한 행렬의 계산과정을 도식화 해보면, 다음과 같다. 그리고 각각의 [그림 1,2]에서 PPG_i (Parallel Product Generator)는 m개의 2입력 ***AND*** 게이트의 병렬 연산구조로써 행렬 M'의 i행의 구성원소와 $\vec{b}$ 다항식 계수를 곱하는 역할을 수행하고 BTX_i (Binary Tree ***XOR***)는 m개의 2입력 ***XOR*** 게이트의 이진트리 연산구조로써 PPG_i에서 계산한 결과들을 더해 다항식 c'의 i번째 계수를 생성하는 역할을 한다. $\otimes$는 2입력 ***AND*** 게이트, $\oplus$는 2입력 ***XOR*** 게이트를 나타낸다.



〔그림 1〕 제안하는 곱셈기의 각 부분의 Linear Shift 확대부분

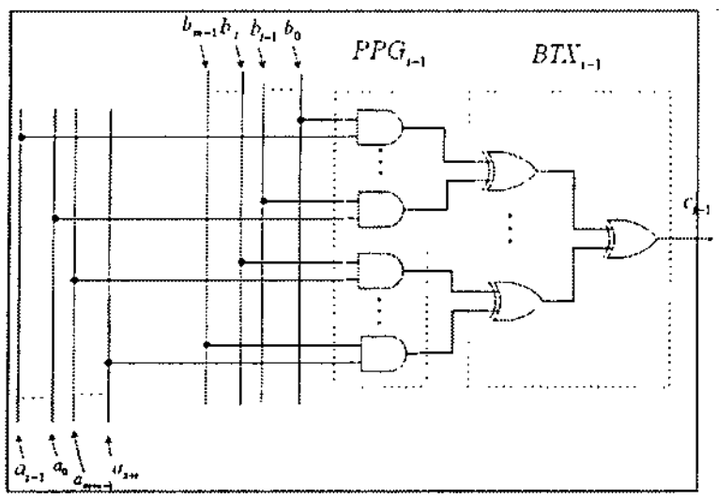


〔그림 2〕 $GF(2^m)$상의 제안한 곱셈기 구조

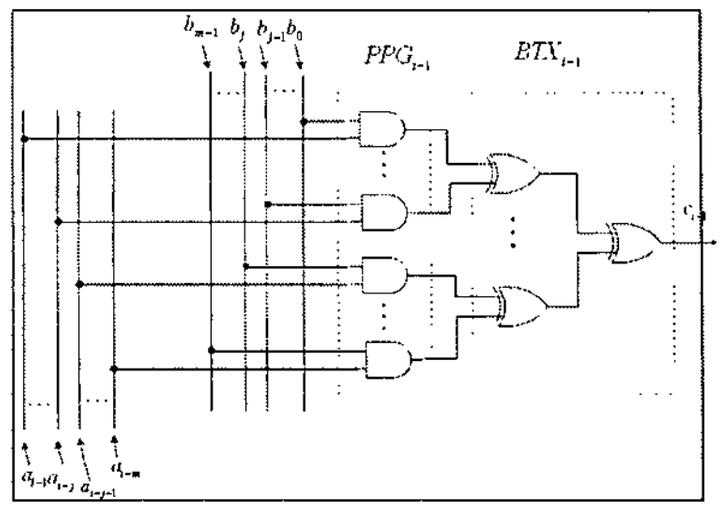
따라서, 그림을 통해 설계된 곱셈기의 복잡도를 살펴보면 다음과 같다. 행렬 M'에 b을 곱할 때 행렬 M'의 각 행은 병렬로 구성되고 총 행의 개수가 $(m+s)$ 개 이므로 공간 복잡도는 ***AND*** 게이트 $m(m+s)$, ***XOR*** 게이트 $(m-1)(m+s)$이고, 시간 복잡도는 $T_A+(\lceil \log_2 m \rceil)T_X$가 소요된다.

〔표 1〕 ESP 다항식에 정의된 $GF(2^m)$의 곱셈기들의 복잡도 비교

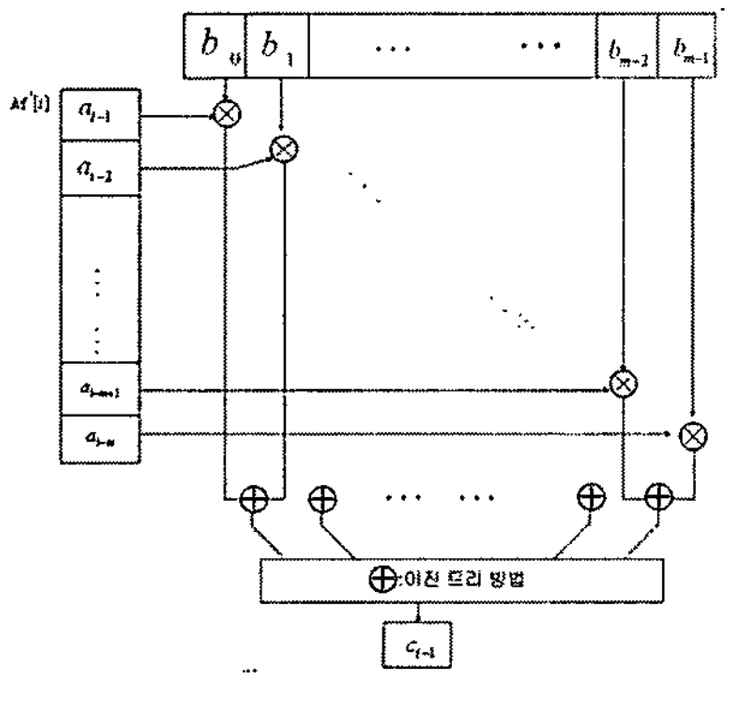
참 고 문 헌	공 간 복 잡 도		시간 복잡도	기저
	#AND	#XOR		
Itoh-Tsujii[15]	$(m+s)^2$	$(m+s)^2-s$	$T_A+(\lceil \log_2 m \rceil+\lceil \log_2(m+s+1) \rceil)T_X$	다항식기저
Hasan et al.[14]	m^2	m^2+m-2s	$T_A+(\frac{m}{s}+\lceil \log_2 m \rceil)T_X$	다항식기저
Mastrovito[16,17]	m^2	$\frac{2s+1}{2s}m^2-\frac{3}{2}m$	$T_A+(1+\lceil \log_2 m \rceil)T_X$	다항식기저
Masoleh AND Hasan [7]	m^2	m^2-s	$T_A+(1+\lceil \log_2 m \rceil)T_X$	다항식기저
기존 방법[10,11,12,13]	$(m+s)^2$	$(m+s-1)(m+s)$	$T_A+(\lceil \log_2(m+s) \rceil)T_X$	여분표현
제안한 방법	$m(m+s)$	$(m-1)(m+s)$	$T_A+(\lceil \log_2 m \rceil)T_X$	여분표현



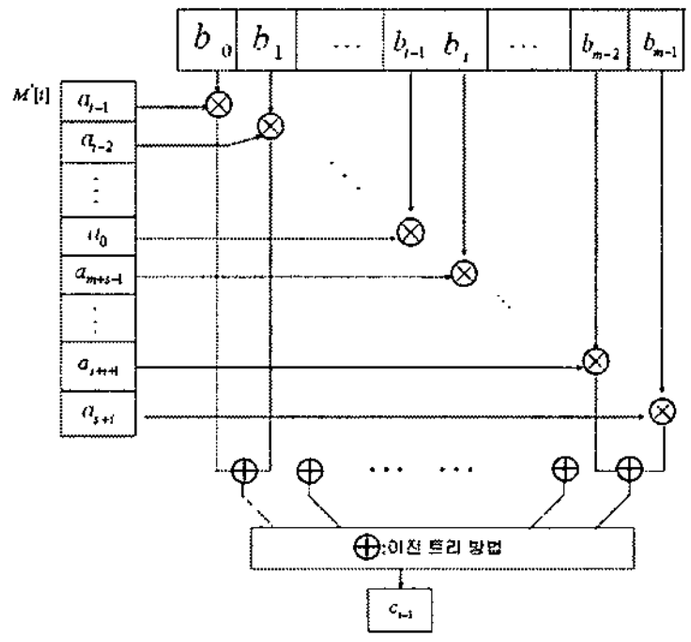
〔그림 3〕 제안하는 곱셈기의 각 부분의 PPG_{i-1} 와 BTX_{i-1} $(1 \le i \le m-1)$



〔그림 4〕 제안하는 곱셈기의 각 부분의 PPG_{i-1}와 BTX_{i-1} $(m \le i \le m+s)$



〔그림 5〕 $GF(2^m)$상의 제안한 곱셈기의 전체구조 중 $M'[i]\,(1 \le i \le m-1)$



〔그림 6〕 $GF(2^m)$상의 제안한 곱셈기의 전체구조 중 $M'[i]\,(m \le i \le m+s)$

Ⅳ. 결론

본 논문은 여분표현을 사용한 기약 ESP에 의해 정의된 $GF(2^m)$에서의 효율적인 비트-병렬 곱셈기를 제안하였다. 제안하는 곱셈기와 기존의 곱셈기들의 공간 및 시간 복잡도를 비교하면 위의 [표 1]과 같다. 본 논문에서 제안하는 곱셈기는 [표 1]에서 보는 것 처럼 시간 복잡도는 다른 곱셈기들과 비교해 가장 효율적 이고 같은 여분표현을 사용한 기존 결과와 비교해 공간-시간 모두 효율성을 가진다. 또한 Left-to-Right 지수승 환경에서와 같이 하나의 곱셈 입력 값이 고정된 형태에서 장점을 가진다.

참고문헌

[1] R. Lidl and H. Niederreiter, Introduction to finite fields and their applications, New York: Cambridge Univ. Press, 1994.

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

[3] A. J. Menezes, Elliptic Curve Public Key Cryptosystems. Boston; Kluwer Academic, 1993.

[4] A. Halbutogullari and C.K. Koc, “Mastrovito Multiplier for General Irreducible Polynomials,” IEEE Trans. Computers, vol. 49, no. 5, pp. 503-518, May 2000

[5] Tong Zhang, Keshab K. Parhi, "Systematic Design of Original and Modified Mastrovito Multipliers for General Irreducible Polynomials" IEEE Trans. Computers, vol. 50, no.7, July 2001

[6] B. Sunar and C. K. Koc, "Mastrovito Multiplier for All Trinomials," IEEE Trans. Computers, vol. 48, no. 5, pp. 522-527, May. 1999.

[7] A. Reyhani-Masoleh and M.A. Hasan, “Low

Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over $GF(2^m)$," IEEE Trans. Computers, vol. 53, no. 8, pp. 945-959, Aug. 2004

[8] K.-Y. Chang, D. Hong, and H.-Y. Cho, "Low complexity bit-parallel multiplier for $GF(2^m)$ defined by all-one polynomials using redundant representation," IEEE Trans. Computers, vol. 54, no. 12, pp. 1628-1630, Dec. 2005.

[9] G. Drolet, "A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Complexity Arithmethic Circuits," IEEE Trans. Computers, vol. 47, no. 9, pp. 938-946, Sep. 1998.

[10] W. Geiselmann and R. Steinwandt "A Redundant Representation of $GF(q^m)$ for Designing Arithmetic Circuits," IEEE Trans. Computers, vol. 52, no. 7, pp. 848-853, July 2003.

[11] W. Geiselmann and H. Lukhaub, "Redundant Representation of Finite Fields,"Proc. Public Key Cryptography, pp. 339-352, 2001.

[12] H. Wu, M. A. Hasan, I. F. Blake, and S. Gao,"Finite field multiplier using redundant representation," IEEE Trans. Computers, vol. 51, no. 11, pp. 1306-1316, Nov. 2002.

[13] C.-Y. Lee, E.-H. Lu, and J.-Y. Lee,"Bit-parallel systolic multipliers for $GF(2^m)$ fields defined by all-one and equally spaced polynomials."IEEE Trans. Computers, vol. 50, no. 5, pp. 385-393, May 2001.

[14] M.A. Hasan, M.Z. Wang, and V.K. Bhargava, "Modular Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields $GF(2^m)$," IEEE Trans. Computers, vol. 41, no. 8, pp. 962-971, Aug. 1992.

[15] T. Itoh and S. Tsujii, "Structure of Parallel Mutipliers for a Class of Fields $GF(2^m)$," Information and Computation, vol. 83, pp. 21-40, 1989.

[16] E.D. Mastrovito, ""VLSI Designs for Multiplication over Finite Fields $GF(2^m)$," Proc. Sixth Symp. Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC-6), pp. 297-309, July 1988.

[17] E.D. Mastrovito, "VLSI Architectures for Computation in Galois Fields," PhD thesis, Linkoping Univ., Linkoping, Sweden, 1991.

〈著者紹介〉

이 옥 석 (Ok Suk Lee) 학생회원
2006년 8월 : 경원대학교 수학과 학사
2006년 9월 ~ 현재 : 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 공개키 암호, 암호칩 설계 기술

김 창 한 (Chang Han Kim) 정회원
1985년 2월 : 고려대학교 수학과 학사
1987년 2월 : 고려대학교 수학과 석사
1992년 2월 : 고려대학교 수학과 박사
1992년 3월~현재 : 세명대학교 정보통신학부 교수
<관심분야> 정수론, 공개키 암호, 프로토콜

장 남 수 (Nam Su Chang) 학생회원
2002년 2월 : 서울 시립대학교 수학과 학사
2004년 8월 : 고려대학교 정보보호 대학원 공학석사
2005년 3월~현재 : 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 인수분해, 공개키 암호, 암호칩 설계 기술

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
1999년 8월~2004년 2월 : (주)시큐리티 테크놀로지스 선임연구원
2003년 2월~2004년 2월 : 고려대학교 시간강사
2004년 4월~2005년 2월 : K.U.Leuven 박사후연구원
2005년 3월~현재 : 고려대학교 정보경영공학전문대학원 조교수
<관심분야> 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식