논문 2007-44SD-8-5

GF(2^m)에서 삼항 기약 다항식을 이용한 약한 쌍대 기저 기반의 효율적인 지수승기

(Efficient polynomial exponentiation in GF(2^m) with a trinomial using weakly dual basis)

김 희 석*, 장 남 수**, 김 창 한***, 임 종 인****

(HeeSeok Kim, Nam Su Chang, Chang Han Kim, and Jongin Lim)

요 약

유한체 GF(2^m)에서의 다항식의 지수승 연산은 암호학(Cryptography), DSP(digital signal processing), 에러 정정 코드에서 기본적인 연산으로 사용되어진다. 기존의 방법들은 지수승 연산을 병렬처리가 가능한 Right-to-Left 이진 방법으로 구성하여 연산시간을 줄이는 방법을 사용하였다. 본 논문에서는 기존의 다항식 기저에서 Right-to-Left 이진 방법으로 구성되었던 다항식의 지수승기를 약한 쌍대 기저 기반에서 삼항 기약다항식을 이용한 Left-to-Right 이진 형태로 구성한다. 제안하는 방법은 Left-to-Right는 고정된 다항식을 곱한다는 점에 착안, 사전계산을 이용하여 연산량을 감소시킨다. 본 논문에서 제안하는 방법은 제곱기(squarer)와 곱셈기(multiplier)를 모두 수행하는 시간이 기존 지수승기의 곱셈기의 연산 시간보다 같거나 작아 Left-to-Right 형태와 Right-to-Left 형태의 기존 지수승기보다 각각 기약 다항식이 x^m+x+1의 경우 약 17%, 10%, $x^m+x^k+1(1<k<m/2)$의 경우 약 21%, 9%, $x^m+x^{m/2}+1$의 경우 15%, 1%의 시간이 단축된다.

Abstract

An exponentiation in GF(2^m) is a basic operation for several algorithms used in cryptography, digital signal processing, error-correction code and so on. Existing hardware implementations for the exponentiation operation organize by Right-to-Left method since a merit of parallel circuit. Our paper proposes a polynomial exponentiation structure with a trinomial that is organized by Left-to-Right method and that utilizes a weakly dual basis. The basic idea of our method is to decrease time delay using precomputation tables because one of two inputs in the Left-to-Right method is fixed. Since $T_{sqr}(squarer\ time\ delay)+T_{mul}(multiplier\ time\ delay)$ of our method is smaller than T_{mul} of existing methods, our method reduces time delays of existing Left-to-Right and Right-to-Left methods by each 17%, 10% for x^m+x+1 (irreducible polynomial), by each 21%, 9% for $x^m+x^k+1(1<k<m/2)$, by each 15%, 1% for $x^m+x^{m/2}+1$.

Keywords: exponentiation, dual basis, weakly dual basis, left-to-right scalar multiplication

I. 서 론

유한체 GF(2^m)에서의 지수승 연산을 수행하는 하드웨어 구조가 최근에 소개되었다[1~2,9~12]. 기존 지수승기들은 곱셈기와 제곱기를 병렬로 처리하여 수행시간을 단축시킬 수 있는 장점을 지닌 Right-to-left 형태로 구성되어 있다.

본 논문에서는 GF(2^m)에서 기존의 다항식 기저(Polynomial Basis)[8] 기반에서만 이루어졌던 다항식의

* 학생회원-주저자, ** 학생회원, **** 정회원,
고려대학교 정보경영공학전문대학원
(Graduate School of Information Management and Security, Korea University)
*** 정회원-교신저자, 세명대학교 정보통신학부
(School of Information & Communnication systems, Semyumg University)
※ "본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2006-(C1090-0603-0025))]
접수일자: 2007년4월2일, 수정완료일: 2007년7월30일

지수승 연산을 기약다항식이 삼항 다항식일 때는 기저 변환 비용이 필요 없는 약한 쌍대 기저 (Weakly Dual Basis)기반에서 Left-to-Right 이진 방법으로 효율적으로 설계한다[4~5]. 제안하는 지수승기는 Left-to-Right 이진 형태의 지수승 연산이 수행될 때, 곱셈 연산은 항상 일정한 다항식 $g(\alpha)$를 곱하는 점에 착안, 사전 계산을 이용하여 연산량과 연산 시간을 감소시킨다. Left-to-Right 이진 방식은 저장 공간이 제한된 장치에 적합한 장점에도 불구하고 제곱기와 곱셈기를 병렬 처리할 수 없는 단점이 있지만, 제안하는 지수승기는 squarer+multiplier에 소요되는 시간이 Right-to-Left 이진 형태의 지수승기에서 multiplier에 소요되는 시간보다 작아 기존보다 연산시간을 줄일 수 있다. 예를 들어, GF(2^{256})에서 160비트의 지수승을 하는 경우, 제안하는 지수승기는 L-t-R 형태와 R-t-L 형태의 기존 지수승기보다 각각 기약 다항식이 x^m+x+1의 경우 약 17%, 10%, $x^m+x^k+1(1<k<m/2)$의 경우 약 21%, 9%, $x^m+x^{m/2}+1$의 경우 15%, 1%의 시간이 단축된다.

본 논문의 구성은 다음과 같다. Ⅱ절에서는 제안하는 지수승기의 수학적 배경으로 약한 쌍대 기저 기반에서의 곱셈연산과 기저변환을 소개한다. Ⅲ절에서는 본 논문에서 제안하는 지수승기를 제안하고 전체구조와 세부구조를 도식화하며 Ⅳ절에서 다항식 기저 기반에서의 지수승기와 본 논문에서 제안하는 지수승기의 연산량과 연산시간을 비교, 분석한다.

Ⅱ. 약한 쌍대 기저

본 절에서는 약한 쌍대 기저기반에서의 곱셈 연산과 약한 쌍대 기저와 다항식 기저간의 기저변환을 소개한다. 우선 본 논문에서 사용하는 기호를 정리하면 다음과 같다.

■ 기호
- $f(x)$: 기약다항식
- α : $f(x)$의 근
- δ_{ij} : $\begin{cases}1 & \text{if } i=j\\ 0 & \text{if } i\neq j\end{cases}$
- $Tr(x)$: $\sum_{i=0}^{m-1} x^{2^i}\in\{0,1\},\ x\in GF(2^m)$

$GF(2^m)$상에서의 기저 $\{\alpha_0,\alpha_1,\alpha_2,...,\alpha_{m-1}\}$에 대하여 $Tr(\alpha_i\beta_j)=\delta_{ij}$의 식을 만족하는 기저 $\{\beta_0,\beta_1,\beta_2,...,\beta_{m-1}\}$를 $\{\alpha_0,\alpha_1,\alpha_2,...,\alpha_{m-1}\}$에 대한 쌍대기저라 한다[6~7]. 또한 $GF(2^m)$상의 어떤 원소 λ에 대해서 $Tr(\lambda\alpha_i\beta_j)=\delta_{ij}$의 식을 만족할 경우 기저 $\{\beta_0,\beta_1,\beta_2,...,\beta_{m-1}\}$를 약한 쌍대 기저라 부른다. 본 논문에서 제안하는 지수승기는 다항식 기저기반에서 제곱연산을 쌍대기저기반에서는 곱셈연산을 수행하는 구조로 기저변환이 불가피하다. 하지만 약한 쌍대 기저 기반에서 삼항 기약 다항식을 사용했을 경우 기저변환 비용이 들지 않기 때문에 본 논문은 약한 쌍대 기저 기반에서 곱셈 연산을 수행하도록 구성한다. 우선 약한 쌍대 기저기반에서의 곱셈연산과 기본연산을 살펴본다.

1. 약한 쌍대 기저 기반의 곱셈 연산

본 절에서는 다항식 기저 기반의 다항식과 이 다항식 기저에 대한 약한 쌍대 기저 기반의 다항식의 곱셈 연산에 대해 살펴본다. 즉, 다항식 기저 $\{1,\alpha,\alpha^2,...,\alpha^{m-1}\}$에 대한 약한 쌍대 기저가 $\{\beta_0,\beta_1,\beta_2,...,\beta_{m-1}\}$일 때, $A(\alpha)=\sum_{i=0}^{m-1}a_i\alpha^i$, $B(\alpha)=\sum_{i=0}^{m-1}b_i'\beta_i$에 대한 $A(\alpha)B(\alpha)$의 연산을 수행하여 약한 쌍대 기저 기반의 다항식 $C(\alpha)$를 출력하는 곱셈 연산을 살펴본다. $A(\alpha)B(\alpha)=C(\alpha)=\sum_{j=0}^{m-1}c_j'\beta_j$라고 한다면, 다음의 식이 성립한다.

$$
\begin{aligned}
&\lambda\alpha^jA(\alpha)B(\alpha)=\lambda\alpha^j\sum_{i=0}^{m-1}c_i'\beta_i\ ,\\
\Rightarrow\ &Tr(\lambda\alpha^jA(\alpha)B(\alpha))=Tr(\lambda\alpha^j\sum_{i=0}^{m-1}c_i'\beta_i)\ ,\\
\Rightarrow\ &Tr(\lambda\alpha^jB(\alpha)\sum_{i=0}^{m-1}a_i\alpha^i)=\sum_{i=0}^{m-1}c_i'\,Tr(\lambda\alpha^j\beta_i)\ ,\\
\Rightarrow\ &\sum_{i=0}^{m-1}a_i\,Tr(\lambda\alpha^{i+j}B(\alpha))=c_j'\ .
\end{aligned}
$$

여기서 $Tr(\lambda\alpha^{i+j}B(\alpha))$의 값은 $i+j<m$이면 b_{i+j}'이고, $i+j\geq m$이면 α^{i+j}를 기약 다항식으로 감산시킨 후 이 값을 계산한다. 예를 들어 기약 다항식이 $x^5+x^3+x^2+x+1$ 이라고 한다면 $i+j\geq 5$에 대한 $Tr(\lambda\alpha^{i+j}B(\alpha))$의 값은 다음과 같이 계산하고 그 값은 b_{i+j}'이라고 정의한다.

$$
\begin{aligned}
b_5' &= Tr(\lambda\alpha^5B(\alpha))=Tr(\lambda\alpha^3B(\alpha))+Tr(\lambda\alpha^2B(\alpha))\\
&\quad +Tr(\lambda\alpha B(\alpha))+Tr(\lambda B(\alpha))\ ,\\
b_6' &= Tr(\lambda\alpha^6B(\alpha))=Tr(\lambda\alpha^4B(\alpha))+Tr(\lambda\alpha^3B(\alpha))\\
&\quad +Tr(\lambda\alpha^2B(\alpha))+Tr(\lambda\alpha B(\alpha)),\\
b_7' &= Tr(\lambda\alpha^7B(\alpha))=Tr(\lambda\alpha^5B(\alpha))+Tr(\lambda\alpha^4B(\alpha))\\
&\quad +Tr(\lambda\alpha^3B(\alpha))+Tr(\lambda\alpha^2B(\alpha)),\\
b_8' &= Tr(\lambda\alpha^8B(\alpha))=Tr(\lambda\alpha^6B(\alpha))+Tr(\lambda\alpha^5B(\alpha))
\end{aligned}
\qquad (1)
$$

따라서 $C(\alpha)$는 식 (2)과 같이 계산된다.

$$\begin{aligned}&(a_0b_0{}' + a_1b_1{}' + a_2b_2{}' + a_3b_3{}' + a_4b_4{}')\beta_0\\ &+ (a_0b_1{}' + a_1b_2{}' + a_2b_3{}' + a_3b_4{}' + a_4b_5{}')\beta_1\\ &+ (a_0b_2{}' + a_1b_3{}' + a_2b_4{}' + a_3b_5{}' + a_4b_6{}')\beta_2\\ &+ (a_0b_3{}' + a_1b_4{}' + a_2b_5{}' + a_3b_6{}' + a_4b_7{}')\beta_3\\ &+ (a_0b_4{}' + a_1b_5{}' + a_2b_6{}' + a_3b_7{}' + a_4b_8{}')\beta_4\,.\end{aligned} \quad (2)$$

본 논문에서는 지수승기의 연산량을 줄이기 위해 약한 쌍대기저 기반에서 기저변환의 비용이 없는 삼항 기약 다항식의 경우만을 고려하여 기술한다.

$f(x)=x^m+x^k+1(1\le k<m/2)$의 경우 곱셈 연산의 연산량은 다음과 같다. 첫 번째 단계로 식 (1)과 같이 $b_{i+j}{}'(i+j\ge m)$을 계산할 때, $m-1$번의 XOR연산이 필요하고 ($b_m{}'\sim b_{2m-2}{}'$을 연산하는데 각각 1 xor 연산 소요), 필요한 시간은 $\left\lceil \frac{m-1}{m-k} \right\rceil T_X$이다.($b_m{}'\sim b_{2m-k-1}{}'$은 병렬로 계산 가능, $b_m{}'$부터 $b_{2m-2}{}'$까지 $m-k$개 씩 묶어서 병렬로 계산 가능.) 두 번째 단계에서 식 (2)과 같이 $C(\alpha)$를 계산하는 과정은 m^2의 AND 연산, $m(m-1)$의 XOR 연산, $T_A+\lceil \log_2 m \rceil T_X$의 시간이 필요하다.

$$\begin{aligned}Tr(\lambda\alpha^{m+i}) &= Tr(\lambda\alpha^i(\alpha^k+1)) = Tr(\lambda\alpha^{i+k})\\ &\quad + Tr(\lambda\alpha^i) = 0\ (0\le i\le k-2)\,,\\ Tr(\lambda\alpha^{m+k-1}) &= Tr(\lambda\alpha^{k-1}(\alpha^k+1)) = Tr(\lambda\alpha^{2k-1})\\ &\quad + Tr(\lambda\alpha^{k-1}) = 1\,,\\ Tr(\lambda\alpha^{m+i}) &= Tr(\lambda\alpha^i(\alpha^k+1)) = Tr(\lambda\alpha^{i+k})\\ &\quad + Tr(\lambda\alpha^i) = 0\ (k\le i\le m-2)\,.\end{aligned}$$

$f(x)=x^m+x^{m/2}+1$의 경우라면, $\alpha^{3m/2}=1$을 만족하므로, 첫 번째 단계에서 $b_j{}'(j\ge m)$의 계산을 할 때, $b_{3m/2}{}'\sim b_{2m-2}{}'$은 $b_0{}'\sim b_{m/2-2}{}'$의 값과 같으므로 $b_m{}'\sim b_{3m/2-1}{}'$의 값만 계산하면 된다. 즉, $m/2$번의 XOR 연산을 필요로 하고, $b_{3m/2-1}{}'=b_{m-1}{}'+b_{m/2-1}{}'$으로 기존의 값에 XOR를 통해 계산할 수 있으므로 동시에 $b_m{}'\sim b_{3m/2-1}{}'$의 값을 구할 수 있다. 따라서 T_X의 시간을 필요로 한다. 두 번째 단계에서 필요한 계산량과 계산 시간은 $\alpha^m+\alpha^k+1(1\le k<m/2)$와 동일하다.

표 1은 삼항 기약 다항식의 형태에 따른 약한 쌍대 기저 기반의 다항식 곱셈 연산량과 연산 시간을 나타낸 것이다.

2. 기저변환

다항식의 기저는 다항식 기저, 정규 기저, 쌍대 기저와 같이 여러 종류의 기저가 있다. 이러한 여러 종류의 기저가 연구되어지는 이유는 환경에 따라 기저들의 장

표 1. 약한 쌍대 기저기반에서의 $f(x)$에 따른 곱셈 연산 비용

Table 1. The cost of multiplier corresponding to $f(x)$ based on weakly dual basis.

	x^m+x+1	x^m+x^k+1 $(1<k<\frac{m}{2})$	$x^m+x^{m/2}+1$
연산 시간	$T_A+(1+\lceil \log_2 m \rceil)T_X$	$T_A+(2+\lceil \log_2 m \rceil)T_X$	$T_A+(1+\lceil \log_2 m \rceil)T_X$
XOR gate	m^2-1	m^2-1	$m^2-\frac{m}{2}$
AND gate	m^2	m^2	m^2

단점이 있기 때문이다. 따라서 연산을 수행할 때 그 연산에 장점을 갖는 기저로 $GF(2^m)$의 원소를 변환하는 작업은 불가피하다. 본 절에서는 임의의 다항식 $A(\alpha)$를 다항식 기저에서 약한 쌍대 기저 기반으로, 또는 약한 쌍대 기저에서 다항식 기저 기반으로 변환하는 기저변환에 대해 알아본다.

$A(\alpha)=\sum_{i=0}^{m-1}a_i\alpha^i=\sum_{j=0}^{m-1}a_j{}'\beta_j$일 때, 다음 식을 만족한다.

$$a_j{}' = Tr(\lambda\alpha^j A) = \sum_{i=0}^{m-1} a_i\, Tr(\lambda\alpha^{i+j})\,.$$

다항식 기저에서 약한 쌍대 기저로 변환하는 기저변환 행렬을 계산하면 다음과 같다.

$$\begin{pmatrix} a_0{}' \\ a_1{}' \\ \vdots \\ a_{m-1}{}' \end{pmatrix} = \begin{pmatrix} Tr(\lambda) & Tr(\lambda\alpha) & \cdots & Tr(\lambda\alpha^{m-1}) \\ Tr(\lambda\alpha) & Tr(\lambda\alpha^2) & \cdots & Tr(\lambda\alpha^m) \\ \vdots & \vdots & \ddots & \vdots \\ Tr(\lambda\alpha^{m-1}) & Tr(\lambda\alpha^m) & \cdots & Tr(\lambda\alpha^{2m-2}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{pmatrix}$$

기저 변환 행렬은 $Tr(\lambda), Tr(\lambda\alpha), \ldots, Tr(\lambda\alpha^{m-1})$의 값에 따라 구성되어 지는 행렬이다.

본 논문에서 다루는 삼항 기약 다항식의 형태가 x^m+x^k+1인 경우, $Tr(\lambda\alpha^{k-1})=1$이고 $Tr(\lambda\alpha^i)=0$ $(0\le i\le m-1, i\ne k-1)$을 만족하는 λ에 대하여 다음 식이 성립한다.

이러한 방법으로 기저 변환 행렬을 구성하면 이 행렬 곱셈은 연산을 필요로 하지 않는 치환(일대일 대응)형태로 구성된다. 즉 기저의 역변환 행렬도 같은 행렬(치환)형태로 구성되어진다.

삼항 기약 다항식에서 $Tr(\lambda\alpha^{k-1})=1$과 $Tr(\lambda\alpha^i)=0$ $(0\le i\le m-1, i\ne k-1)$을 만족할 때, $Tr(\lambda\alpha^i\beta_j)=\delta_{ij}$을 만족하는 $\{1,\alpha,\alpha^2,\ldots,\alpha^{m-1}\}$의 약한 쌍대 기저 표현은

$\{\alpha^{\sigma(0)},\alpha^{\sigma(1)},\alpha^{\sigma(2)},...,\alpha^{\sigma(m-1)}\}(\sigma(t)=k-1-t \bmod m)$로 표현되어질 수 있다. 예를 들어, $f(x)=x^m+x^2+1$의 경우 쌍대 기저 표현은 $\{\alpha^1,\alpha^0,\alpha^{m-1},\alpha^{m-2},...,\alpha^2\}$의 형태가 된다.

III. 제안하는 지수승기 구조

본 논문에서는 효율적인 다항식 지수승기를 설계한다. 알고리즘 1은 본 논문에서 제안하는 다항식 기저와 약한 쌍대 기저 기반의 지수승 연산을 수행하는 알고리즘이다.

알고리즘 1.
Polynomial exponentiation MSB-first

입력 $g(\alpha)$, $H=(h_{n-1}h_{n-2}...h_1h_0)_2$, $h_{n-1}=1$
출력 $Q(\alpha)=g(\alpha)^H \bmod f(\alpha)$
$(Q(\alpha)=[Q_{m-1}, Q_{m-2}, \cdots, Q_1, Q_0]$,
$g(\alpha)=[g_{m-1}, g_{m-2}, \cdots, g_1, g_0])$

1. $G=[g_{2m-2}', g_{2m-1}', \cdots, g_1', g_0']$
   $=p.computer([g_{m-1}, g_{m-2}, \cdots, g_1, g_0])$
2. $Q(\alpha)=g(\alpha)$
3. For $i=n-2$ down to 0 do :
   3.1. $Q(\alpha)=squarer(Q(\alpha))$
   3.2. If $h_i=1$, $Q(\alpha)=multiplier(Q(\alpha),G)$
4. Return $[Q_{m-1}, Q_{m-2}, \cdots, Q_1, Q_0]$

알고리즘 1의 $squarer$는 다항식 기저 기반의 $Q(\alpha)^2 \bmod f(\alpha)$ 연산을 수행하는 연산기이며, $multiplier$는 약한 쌍대 기저 기반의 $Q(\alpha)g(\alpha) \bmod f(\alpha)$ 연산과 약한 쌍대 기저에서 다항식 기저로의 기저 변환을 동시에 수행하는 연산기이다. $p.computer$는 $multiplier$의 연산량을 줄이기 위한 사전계산기로 $Q(\alpha)g(\alpha) \bmod f(\alpha)$의 연산이 일정한 $g(\alpha)$를 곱한다는 점을 이용해 설계되어진다. 제안하는 지수승기를 도식화하면 그림 1과 같다.

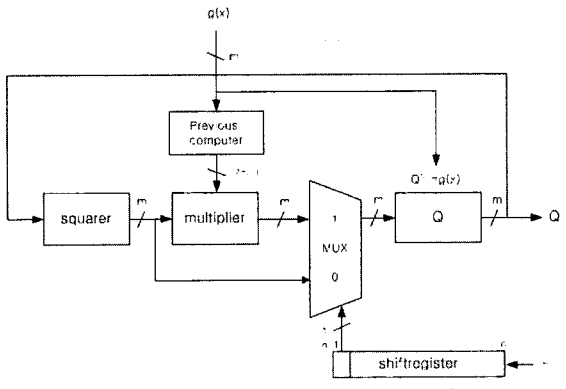

그림 1. $GF(2^m)$에서의 제안하는 지수승기 구조
Fig. 1. Proposed exponentiator in $GF(2^m)$.

입력값 $g(\alpha)$는 Q_register와 $p.computer$에 입력 값으로 들어가 $multiplier$의 연산을 줄일 수 있는 약한 쌍대 기저 기반의 원소들(이 원소들의 역할은 $multiplier$에서 상세히 다룬다),$g_{2m-2}', g_{2m-3}', \cdots, g_1', g_0'$을 생성한다. 생성된 이 $2m-1$ 비트의 값은 g_register에 저장된다. 또한 지수 H는 H_register에 저장되어 한 비트씩 shift 연산을 수행하며 FSM (finite state machine)에 입력 값으로 들어가 회로의 상태를 결정한다.

1. 제곱기($squarer$)

본 절에서는 삼항 기약 다항식 $f(x)$의 형태에 따른 제곱기를 설계한다. 우선 제곱기를 설계하기 위해 다항식의 자승을 수식화하면 다음과 같다.

경우 1. $f(x)=x^m+x+1$ 기약 다항식 $f(x)$가 x^m+x+1의 형태일 때, 다항식 $Q(\alpha)=\sum_{i=0}^{m-1} q_i\alpha^i$의 $Q(\alpha)^2 \bmod f(\alpha)$의 연산을 살펴보면 다음과 같다.

$$
\begin{aligned}
Q(\alpha)^2 \bmod f(\alpha) &= \sum_{i=0}^{m-1} q_i\alpha^{2i} = \sum_{i=0}^{\lfloor \frac{m-1}{2} \rfloor} q_i\alpha^{2i} + \sum_{i=\lfloor \frac{m+1}{2} \rfloor}^{m-1} q_i\alpha^{2i} \\
&= \sum_{i=0}^{\lfloor \frac{m-1}{2} \rfloor} q_i\alpha^{2i} + \sum_{i=\lfloor \frac{m+1}{2} \rfloor}^{m-1} q_i(\alpha^{2i-m+1}+\alpha^{2i-m}) \\
&= \begin{cases} \sum_{i=0}^{\frac{m-2}{2}} (q_i+q_{\frac{m}{2}+i})\alpha^{2i} + \sum_{i=0}^{\frac{m-2}{2}} q_{\frac{m}{2}+i}\alpha^{2i+1} & m: even \\ q_0 + \sum_{i=1}^{\frac{m-1}{2}} (q_i+q_{\frac{m-1}{2}+i})\alpha^{2i} + \sum_{i=0}^{\frac{m-3}{2}} q_{\frac{m+1}{2}+i}\alpha^{2i+1} & m: odd \end{cases}
\end{aligned}
\tag{3}
$$

따라서 기약 다항식 $f(x)$가 x^m+x+1의 형태일 때, 다항식 기저 기반의 제곱 연산에는 식 (3)과 같이 $\lfloor \frac{m}{2} \rfloor$ 번의 XOR연산과 $1 \cdot T_X$(T_X : XOR연산을 한 번할 때, 필요한 시간)의 시간이 소요된다.

경우 2. $f(x)=x^m+x^k+1(1<k\le\frac{m}{2})$ $f(x)$가 $x^m+x^k+1(1<k\le\frac{m}{2})$의 형태일 때, $Q(\alpha)$의 제곱 연산은 다음과 같이 표현된다. 수식의 편의상 m은 짝수

이고, k는 홀수인 경우만을 고려하였다. 식 (4)의 앞의 두 항은 짝수 차수의 값이고, 뒤의 두 항은 홀수 차수의 값이다.

$$
\begin{aligned}
Q(\alpha)^2 \bmod f(\alpha) &= \sum_{i=0}^{\frac{m}{2}-1} q_i \alpha^{2i} + \sum_{i=0}^{\frac{m}{2}-1} q_{\frac{m}{2}+i} \alpha^{m+2i} \\
&= \sum_{i=0}^{\frac{m}{2}-1} q_i \alpha^{2i} + \sum_{i=0}^{\frac{m}{2}-1} q_{\frac{m}{2}+i} (\alpha^{k+2i} + \alpha^{2i}) \\
&= \sum_{i=0}^{\frac{m}{2}-1} q_i \alpha^{2i} + \sum_{i=0}^{\frac{m-k-1}{2}} q_{\frac{m}{2}+i} (\alpha^{k+2i} + \alpha^{2i}) \\
&\quad + \sum_{i=\frac{m-k+1}{2}}^{\frac{m}{2}-1} q_{\frac{m}{2}+i} (\alpha^{2k+2i-m} + \alpha^{k+2i-m} + \alpha^{2i}) \\
&= \sum_{i=0}^{\frac{m}{2}-1} (q_i + q_{\frac{m}{2}+i}) \alpha^{2i} + \sum_{i=\frac{k+1}{2}}^{k-1} q_{m-k+i} \alpha^{2i} \\
&\quad + \sum_{i=0}^{\frac{k-3}{2}} q_{m-\frac{k-1}{2}+i} \alpha^{2i+1} + \sum
\end{aligned}
\tag{4}
$$

식 (4)와 유사한 방법으로 m, k를 각각 짝수인 경우와 홀수인 경우로 나눠 생각하면, 기약 다항식 $f(x)$가 $x^m + x^k + 1 (2 < k \leq \frac{m}{2})$의 형태일 때, 다항식 기저 기반의 제곱 연산에는 $\left\lfloor \frac{m+k-1}{2} \right\rfloor$ 번의 XOR연산과 $2 \cdot T_X$의 시간이 소요된다.

한 예로, $f(x) = x^7 + x^3 + 1$의 형태일 때 제곱기를 (4)의 식에 따라 설계하면 그림 2와 같다.

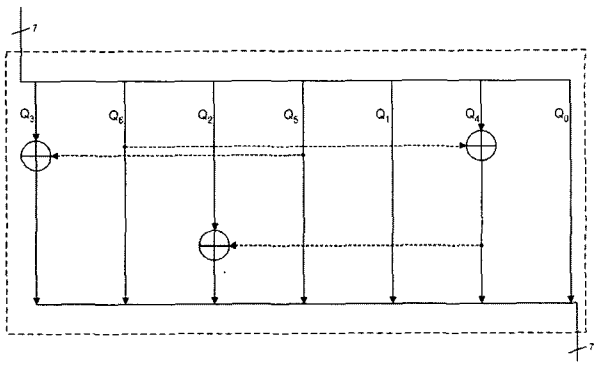

그림 2. 제곱기 : $f(x) = x^7 + x^3 + 1$
Fig. 2. squarer : $f(x) = x^7 + x^3 + 1$.

2. 사전계산기($p.computer$)와 곱셈기($multiplier$)

본 논문에서 제안하는 지수승기는 삼항 기약 다항식에서의 다항식 기저와 약한 쌍대 기저 사이의 기저변환은 연산이 필요하지 않다는 사실과 약한 쌍대 기저 기반에서 어떤 다항식에 대해 같은 다항식을 여러 번 곱하는 환경에서 사전 계산이 가능하다는 사실을 이용해 설계되어진다. 사전계산은 식(1)과 같이 b_{i+j}'의 연산을 지수승이 수행되기 전에 사전에 계산하는 단계로, 알고리즘 1의 단계 3.2에서 곱셈기(multiplier)가 $Q(\alpha)g(\alpha)$의 연산에서 항상 일정한 $g(\alpha)$를 곱하는 점을 이용한다. 예를 들어, $f(x) = x^7 + x^3 + 1$일 때 사전계산은 다음과 같이 수행된다.

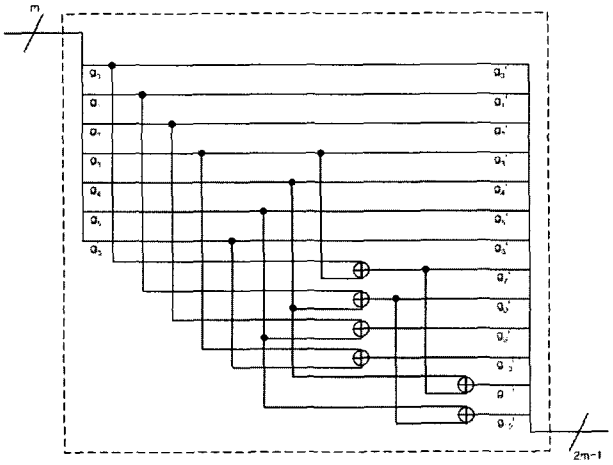

그림 3. 사전계산기($p.computer$) : $f(x) = x^7 + x^3 + 1$
Fig. 3. Pre-computer($p.computer$) : $f(x) = x^7 + x^3 + 1$.

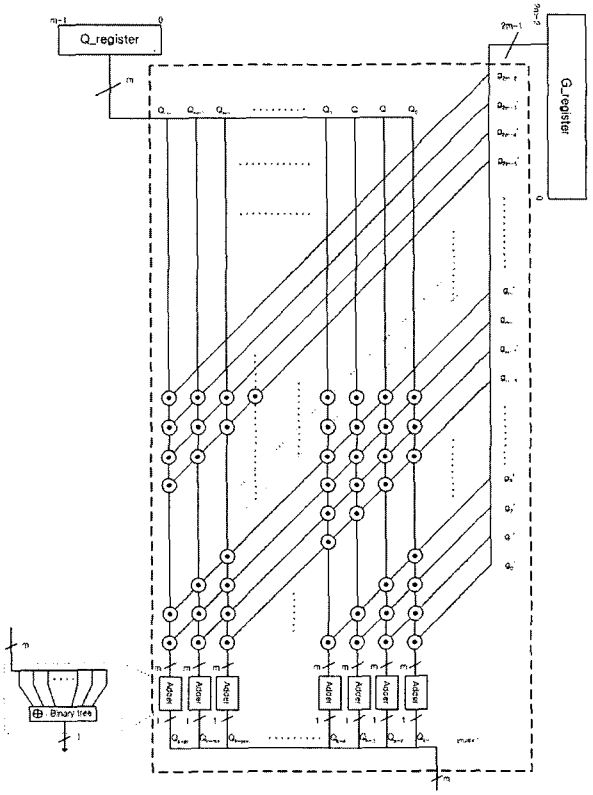

그림 4. 곱셈기
Fig. 4. multiplier.

$$g_7' = Tr(\lambda\alpha^7 B) = Tr(\lambda\alpha^3 B) + Tr(\lambda B) = g_3' + g_0'$$
$$g_8' = Tr(\lambda\alpha^8 B) = Tr(\lambda\alpha^4 B) + Tr(\lambda\alpha B) = g_4' + g_1'$$
$$g_9' = Tr(\lambda\alpha^9 B) = Tr(\lambda\alpha^5 B) + Tr(\lambda\alpha^2 B) = g_5' + g_2'$$
$$g_{10}' = Tr(\lambda\alpha^{10} B) = Tr(\lambda\alpha^6 B) + Tr(\lambda\alpha^3 B) = g_6' + g_3'$$
$$g_{11}' = Tr(\lambda\alpha^{11} B) = Tr(\lambda\alpha^7 B) + Tr(\lambda\alpha^4 B) = g_7' + g_4'$$
$$g_{12}' = Tr(\lambda\alpha^{12} B) = Tr(\lambda\alpha^8 B) + Tr(\lambda\alpha^5 B) = g_8' + g_5'$$

식 (5)를 이용해 삼항 기약 다항식 $f(x) = x^7 + x^3 + 1$일 때의 사전계산기를 도식화하면 그림 3과 같다.

약한 쌍대 기저 기반에서의 $Q(\alpha)g(\alpha)$의 곱셈 연산은 사전계산이 되어졌다면 본 계산에서는 식 (2)와 같은 연산과 본 논문의 2.2에서 소개했던 약한 쌍대기저에서 다항식 기저 기반으로의 기저 변환이 필요하다. 하지만 $f(x) = x^m + x^k + 1$의 기약다항식에서의 기저 변환은 치환형태로 수행되어지기 때문에 별도의 연산은 필요 없고, 따라서 본 논문에서 제안하는 *multiplier*는 식 (4)에서의 연산과 같이 단지 $T_A + \lceil \log_2 m \rceil T_X$의 시간만을 필요로 하게 된다. 삼항 기약다항식에서의 치환을 포함한 곱셈기(multiplier)를 도식화하면 그림 4와 같다.

Ⅳ. 비 교

GF(2^m)에서 기약다항식이 삼항 다항식일 경우 제안하는 지수승기를 사용하여 지수의 길이가 n비트일 때의 연산 시간과 연산량을 비교해 보면 표 2와 같다.

Ⅴ. 결 론

본 논문에서 제안하는 지수승기를 Wu[3]가 제안한 곱셈기와 제곱기를 이용해 다항식 기저 기반으로 구성했을 경우와 비교해보면 표 3과 같다.

표 2. $f(x)$의 형태에 따른 제안하는 지수승기의 비용

Table 2. The cost of proposed exponentiator corresponding to $f(x)$.

			$x^m + x + 1$	$x^m + x^k + 1$ $(1 < k < \frac{m}{2})$	$x^m + x^{m/2} + 1$
사전계산	소요시간		T_X	$2T_X$	T_X
	연산량	xor	$m-1$	$m-1$	$\frac{m}{2}$
		and	0	0	0
제곱승기	소요시간		$(n-1)T_X$	$2(n-1)T_X$	$2(n-1)T_X$
	연산량	xor	$\lfloor \frac{m}{2} \rfloor$	$\lfloor \frac{m+k-1}{2} \rfloor$	$\lfloor \frac{3m-2}{4} \rfloor$
		and	0	0	0
곱셈기	소요시간		$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$	$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$	$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$
	연산량	xor	$m(m-1)$	$m(m-1)$	$m(m-1)$
		and	m^2	m^2	m^2

본 논문에서 제안하는 지수승기의 gate 수는 기존 방법들과 AND gate는 모든 경우 m^2으로 같고, XOR gate는 삼항 기약 다항식이 $x^m + x + 1$의 경우 $m^2 - 1 + \lfloor \frac{m}{2} \rfloor$ 개, $x^m + x^k + 1 (1 < k < \frac{m}{2})$의 경우 $m^2 - 1 + \lfloor \frac{m+k-1}{2} \rfloor$ 개, $x^m + x^{m/2} + 1$의 경우 $m^2 + \lfloor \frac{m-2}{4} \rfloor$ 개로 동일하지만 표 3에서 보는

표 3. $f(x)$에 따른 지수승기의 비교(n:지수의 비트수, method 1: 다항식 기저기반의 L-t-R 지수승기, method 2: 다항식 기저기반의 R-t-L 지수승기, method 3: 제안하는 지수승기)

Table 3. The comparison of exponentiators corresponding to $f(x)$(n: the length of exponent, method 1: Left-to-Right method based on polynomial basis, method 2: Right-to-Light method based on polynomial basis, method 3: Proposed method).

$f(x)$	method	$x^m + x + 1$	$x^m + x^k + 1 (1 < k < \frac{m}{2})$	$x^m + x^{m/2} + 1$
TIME DELAY	1	$(n-1)\{T_A + (3 + \lceil \log_2 m \rceil)T_X\}$	$(n-1)\{T_A + (5 + \lceil \log_2 m \rceil)T_X\}$	$(n-1)\{T_A + (4 + \lceil \log_2 m \rceil)T_X\}$
	2	$n\{T_A + (2 + \lceil \log_2 m \rceil)T_X\}$	$n\{T_A + (3 + \lceil \log_2 m \rceil)T_X\}$	$n\{T_A + (2 + \lceil \log_2 m \rceil)T_X\}$
	3	$(n-1)\{T_A + (1 + \lceil \log_2 m \rceil)T_X\} + T_X$	$(n-1)\{T_A + (2 + \lceil \log_2 m \rceil)T_X\} + 2T_X$	$(n-1)\{T_A + (2 + \lceil \log_2 m \rceil)T_X\} + T_X$

것처럼 연산 시간이 감소한다. 예를 들어, GF(2^{256})에서 160비트의 지수승을 하는 경우, 제안하는 지수승기는 L-t-R 형태와 R-t-L 형태의 기존 지수승기보다 각각 기약 다항식이 x^m+x+1의 경우 약 17%, 10%, $x^m+x^k+1(1<k<m/2)$의 경우 약 21%, 9%, $x^m+x^{m/2}+1$의 경우 15%, 1%의 시간이 단축된다.

참 고 문 헌

[1] Lee K-J. and Yoo K-Y. "Linear systolic multiplier/squarer for fast exponentiation." Information Processing Letters. 2000, vol.76, pp. 105-111.

[2] M. A. G. Martinez, G. M. Luna, F. R. Henriquez. "Hardware Implementation of the Binary Method for Exponentiation" in GF(2m), IEEE Trans, on Computers, 2003, ENC'03

[3] Huapeng Wu, "Efficient computations in Finite Fields with Cryptographic Significance", PhD thesis, University of Waterloo, 1998.

[4] M. Morii, M. Kasahara, and D.L. Whiting. "Efficient bit-serial multiplication and discrete-time Wiener-Hoph equation over finite fields", IEEE Trans. IT, 35:1177-1184, 1989.

[5] M. Wang and I.F.Blake. "Bit serial multiplication in finite fields", SIAM Discrete Mathematics, 3(1):140-148, 1990.

[6] S.T.J. Fenn, M. Benaissa, and D.Taylor. "GF(2-m) multiplication and division over the dual basis.", IEEE Trans. Comput., 45(3):319-327, 1996.

[7] C. Paar. "Efficient VLSI architectures for Bit-Parallel Computation in Galois Fields.", VDI-Verlag, Dusseldorf, 1994. Ph.D Thesis.

[8] J.L. Massey and J.K. Omura. "Computational method and apparatus for finite field arithmetic." U.S. Patent No. 4587627, 1984.

[9] Wang, C.L. "Bit-Level Systolic Array for Fast Exponentiation in GF(2m)," IEEE Trans. on Comp.,1994, vol.43(7), pp. 838-841.

[10] Kovac M. and Ranganathan N. "A VLSI Chip for Galois Field GF(2m) Based Exponentiation." IEEE Trans. on Circuits and Systems-II, 1996, vol. 43, no. 4, pp. 289-297.

[11] Blum T. and Paar C. "Montgomery Modular Exponentiation on Reconfigurable Hardware." 14th IEEE Symposium on Computer Arithmetic. 1999, Adelaide, Australia.

[12] Paar C. and Soria-Rodriguez P. "Fast Arithmetic for Public-Key Algorithms in Galois Fields with Composite Exponents." IEEE Trans. on Computers, 1999, vol. 48, no. 10, pp. 1025-1034.

저 자 소 개

김 희 석(학생회원)
2006년 2월 연세대학교 수학과 학사
2006년~현재 고려대학교 정보경영공학전문대학원 석사과정

<주관심분야 : 부채널 공격, 공개키 암호시스템 안전성 분석 및 고속구현, 타원곡선>

김 창 한(정회원)
1985년 2월 고려대학교 수학과 학사
1987년 2월 고려대학교 수학과 석사
1992년 2월 고려대학교 수학과 박사
2002년 2월~현재 세명대학교 정보통신학부 부교수

<주관심분야: 정수론, 공개키암호, 암호프로토콜>

장 남 수(학생회원)
2002년 2월 서울시립대학교 수학과 학사
2005년 2월 고려대학교 정보경영공학전문대학원 석사 과정
2005년~현재 고려대학교 정보경영공학전문대학원 박사과정

<주관심분야: 공개키암호 암호칩설계기술 부채널 공격방법론>

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

<주관심분야 : 암호이론, 정보보호정책>