Regular paper http://dx.doi.org/10.13067/JKIECS.2013.8.8.1207

유한체위에서의 고속 최적정규기저 직렬 연산기

김용태*

Fast Sequential Optimal Normal Bases Multipliers over Finite Fields

Yong-Tae Kim*

요 약

유한체 연산은 부호이론과 암호학에 널리 쓰이고 있으므로, 유한체 연산의 복잡도를 낮출 수 있는 연산기가 절실하게 필요하다. 그런데 연산기의 복잡도는 유한체의 원소를 표현하는 방법에 달려있다. 복잡도를 줄이기 위해서, 지금까지 알려진 원소를 표현하는 가장 좋은 방법이 최적정규기저를 사용하는 것이다. 본 논문에서는 최적정규기저로 표현된 원소의 곱셈시에 구축되는 곱셈행렬의 1의 개수를 최소화하는 알고리즘을 개발하여 시간과 공간을 최소화하는 곱셈기를 제안하고자 한다.

ABSTRACT

Arithmetic operations over finite fields are widely used in coding theory and cryptography. In both of these applications, there is a need to design low complexity finite field arithmetic units. The complexity of such a unit largely depends on how the field elements are represented. Among them, representation of elements using a optimal normal basis is quite attractive. Using an algorithm minimizing the number of 1's of multiplication matrix, in this paper, we propose a multiplier which is time and area efficient over finite fields with optimal normal basis.

키워드

Finite Field, Finite Field Arithmetic, Optimal Normal Basis, Sequential Multiplier

유한체, 유한체 연산, 최적정규기저, 직렬 곱셈 연산기

I. 서 론

유한체는 ECC, XTR, AES 등에서 채택하고 있기 때문에 H/W에서의 유한체 연산의 구현 속도가 암호계의 안전성과 동시에 공격의 중요한 요인이 되고 있다. 특히 유한체 $GF(2^m)$의 원소를 정규기저로 표현하게 되면 제곱 연산은 단순히 좌표의 순환 이동이 되기 때문에 연산시간에 포함되지 않는 큰 이점이 있다. 유한체의 H/W 상에서의 연산의 시도는 Massey-Omura의 연산기[1]에서 이루어 졌으나 지연시간이 매우 긴 단점을 내포하고 있었다. 그러한 지연시간을 현저하게 줄여서 Agnew 등[2]이 최적정규기저를 갖는 유한체위에서의 직렬 연산기를 제안하였다. H/W의 계산에는 이차체의 원소를 가우스 정규기저로 표현할 때[3]보다 최적정규기저(Optimal Normal Basis, ONB)로 표현할 때의 연산 수행이 효율적이다.

* 교신저자(corresponding author) : 광주교육대학교 수학교육과 교수(ytkim@gnue.ac.kr)
접수일자 : 2013. 06. 13 심사(수정)일자 : 2013. 07. 23 게재확정일자 : 2013. 08. 23

그 후 Reyhani-Masoleh 와 Hasan[4]이 Agnew 등[2]의 제안한 연산기의 복잡도를 현저하게 줄인 ONB를 갖는 유한체위에서의 연산기를 제안하였다. 따라서 ECC의 구현이나 이차체위에서의 이진수열의 계산[5], 그들의 상호상관관계의 계산[6] 등의 H/W 상에서 구현속도를 증진하기 위해서는 ONB를 갖는 유한체를 구축하는 것이 필요하다. 본 논문에서는 최적정규기저를 가지는 이차체 $GF(2^m)$를 선택하여 ONB의 순환성을 이용하면서, 복잡도와 지연시간을 줄이기 위하여 최적정규기저로 표현된 원소끼리의 곱셈시에 생성되는 곱셈행렬의 0이 아닌 성분인 1의 개수를 최소화하는 알고리즘을 적용한 고속 직렬 연산기를 제안하기로 한다.

II. 최적정규기저(ONB)

이 장에서는 ONB의 생성과정을 간략하게 소개하기로 한다.

2.1. 최적정규기저

모든 자연수 l에 대하여, 유한체 $GF(2^l)$은 기초체 $GF(2)$위에서 언제나 정규기저(normal basis)를 가진다는 것은 널리 알려진 사실이다. 즉, $GF(2^l)$의 한 원소 β가 항상 존재하여, $N=\{\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{l-1}}\}$가 $GF(2)$위에서 $GF(2^l)$ 의 한 정규기저가 된다. 이러한 ★를 정규기저 N의 생성자(generator)라고 부르며, 모든 $A \in GF(2^l)$는 이 정규기저로 다음과 같이 표현되고

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

보통 좌표로 $A=(a_0, a_1, \ldots, a_{l-1})$로 표기하거나 T가 행렬의 전치, $\bar{\star}=[\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{l-1}}]$, $\bar{a}=[a_0, a_1, \cdots, a_{l-1}]$ 는 행벡터 일 때, 벡터 형태인

$$A=\bar{a} \times \bar{\star}^T = \bar{\star} \times \bar{a}^T$$

와 같이 표기하기도 한다. $GF(2^l)$의 원소 A를 정규기저로 표현할 때의 장점은 A^2 의 계산은 A의 좌표를 단 한번 우측순환이동(Right Cyclic Shift, RCS)으로 수행된다는 것이다. 즉,

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

$A, B \in GF(2^l)$의 곱셈 $C=AB$ 절차, 곱셈과정에서 발생하는 곱셈행렬의 내용과 다음 정리는Gao등[7]과 Kim[8]을 참조하였다.

정리 1. C_N을 곱셈행렬의 0이 아닌 성분의 개수라고 하면, $C_N \geq 2l-1$ 이다[7].

특히, $C_N = 2l-1$인 정규기저를 최적정규기저라고 한다.

2.2. 최적정규기저의 종류

유한체의 H/W 와 S/W 에서의 연산을 수행하기 위해서는 복잡도가 낮은 정규기저를 사용해야 한다. 유한체에는 많은 ONB가 존재하지만 결국은 다음 두 종류의 ONB와 동치이다.

정리 2. (유형 I ONB) 유한체 $GF(2^n)$이 유형 I ONB 를 가질 필요충분조건은 다음과 같다.

(1) n+1이 솟수이고 $GF(n+1)^*$이 2를 생성자로 갖는 순환군 $GF(n+1)^*=<2>$ 이거나 또는

(2) 기약다항식 $x^n+x^{n-1}+\ldots+x+1$ 의 어떤 근이 ONB 의 생성자이다[7].

정리 3.(유형 II ONB) 유한체 $GF(2^n)$이 유형 II ONB 를 가질 필요충분조건은 다음과 같다.

(1) 2m+1이 솟수이고 $GF(2m+1)^*=<2>$ 또는

(2) 만일 $2m+1 \equiv 3 \bmod 4$ 이고

$GF(2m+1)^*=<2>$이면 $\beta=\gamma+\gamma^{-1}$ 가 $GF(2)$위에서 $GF(2^m)$의ONB의 생성자이다. 단, Γ는 2m+1의 원시근이다[7].

스마트카드에 가장 많이 사용되는 ECC에서는 ONB를 주로 사용하기 때문에 ANSI[9]에서는 $m=191, 239$인 유형 II ONB를 권장하고 있으며, NIST[10]에서는 $m=233$ 인 유형 II ONB를 권장하고 있다. 따라서 본 논문에서는 m이 홀수이고 ONB

를 가지는 유한체 $GF(2^m)$만을 다루기로 한다.

III. ONB를 갖는 $GF(2^m)$에서의 직렬 연산기

이 장에서는 유형 II의 ONB를 갖는 유한 이차체 $GF(2^m)$위에서의 직렬 연산기를 소개하려고 한다.

3.1. Reyhani-Masoleh 와 Hasan(RMH)의 직렬 연산기

먼저 α를 $GF(2^m)$의 유형 II ONB의 생성자라고 하자. 즉, $\alpha_i=\alpha^{2^i}$이면 $\{\alpha_0,\alpha_1,\cdots,\alpha_{m-1}\}$은 $GF(2^m)$의 유형 II ONB이다. 그러면 $\alpha\alpha_i=\sum_{j=0}^{m-1}\lambda_{ij}\alpha_j$, 단 $\lambda_{ij}\in GF(2)$이다. 이제 $GF(2^m)$의 임의의 두 원소 A,B를 유형 II ONB $\{\alpha_0,\alpha_1,\cdots,\alpha_{m-1}\}$에 관하여

$A=\sum_{i=0}^{m-1}a_i\alpha_i$, $B=\sum_{j=0}^{m-1}b_j\alpha_j$ 로 표현할 때, 두 원소의 곱 $C=AB$를 계산하기 위하여 Reyhani-Masoleh 와 Hasan[4]은 다음과 같은 조건하에서 정리 4를 증명하였다.

즉, $\delta_j=\alpha\alpha_j$, $v=\lfloor\frac{m-1}{2}\rfloor$ 이고, $1\le j\le v$에 대하여 $F_i(A,B)=a_{i-g}b_{i-g}\alpha+\sum_{j=1}^{v}z_{ij}\delta_j$, 단,

$g=0$이면 $z_{ij}=(a_i+a_{i+j})(b_i+b_{i+j})$이고,

$g=1$이면 $z_{ij}=a_ib_{j+i}+a_{i+j}b_i$.

정리 4. ([4, Theorem 1] 참조)

$A,B\in GF(2^m)$, $C=AB$이면

$C=(((F_{m-1}^2+F_{m-2})^2+F_{m-3})^2+\cdots+F_1)^2+F_0$.

이다.

따라서 δ_j는 곱셈행렬 (λ_{ij})에 의해서 결정되며, 곱셈행렬 (λ_{ij})의 첫째 행과 첫째 열을 제외한 모든 행과 열에는 꼭 두 개의 1이 있게 된다. 이 때 행렬 (λ_{ij})의 i번 째 행의 두 개의 1 사이에 있는 0의 개수를 $l(i)$로 정의하자. 그러면 m이 홀수이므로, $l(i)$가 홀수인 경우에는 $l(i)$는 $m-l(i)$이 되고, 짝수이면 $l(i)$ 그대로이다.

3.2 $GF(2^5)$위에서의 RMH의 직렬 연산기

기약다항식 z^5+z^2+1에 의해서 생성되는 유한체를 $GF(2^5)$, α를 기약다항식의 근이라고 하자. 그러면 $\beta=\alpha^5$으로 놓으면 $\{\beta,\ \beta^2,\ \beta^4,\ \beta^8,\ \beta^{16}\}$은 $GF(2^5)$의 정규기저임을 알 수 있다. 이 정규기저를 정리 4에 적용하면 곱셈행렬은

$$\begin{pmatrix}0&0&1&0&1\\0&0&1&1&0\\1&1&0&0&0\\0&1&0&1&0\\1&0&0&0&0\end{pmatrix}$$ 이다.

이제, $\delta_1=\beta+\beta^8$과 $\delta_2=\beta^8+\beta^{16}$ 을 3.1 절에서 소개한 RMH에 적용하여 구축한 연산기는 그림 1과 같다.

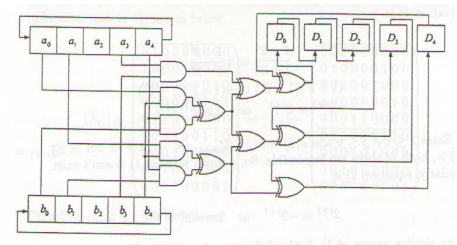

그림 1. RMH 연산기
Fig. 1 RMH multiplier

IV. 제안하는 고속 직렬연산기

3.1 절에서 소개한, 유형 II의 ONB를 갖는 유한 이차체 $GF(2^m)$에서의 $l(i)$의 중요한 성질을 다음의 보조정리에서 증명하고자 한다.

보조정리 1. 유형 II의 ONB를 갖는 유한 이차체 $GF(2^m)$에서는 $1\le i\ne j\le v$ 인 경우에는$l(i)\ne l(j)$ 이다.

(증명) $GF(2^m)$가 유형 II의 ONB를 갖는 경우에는, 곱셈행렬 (λ_{ij})에서 성분 λ_{ij}=1 이기 위한 필요충분조건은 $2^i\pm2^j\equiv\pm1 \mod 2m+1$ 이므로 $1\le i\ne i'\le v$, $0\le j,j'\le m-1$에 대하여 $k<m-1$인 k에 대해서는 다음의 두 식이 성립하지 않는다는 사실을 증명하면 충분하다. 또한 아래의 모든 합동식은 $km+1$을 법으로 계산한다.

① $\begin{cases}2^i \pm 2^j \equiv \pm 1 \\ 2^i \pm 2^{j+k} \equiv \pm 1\end{cases}$, ② $\begin{cases}2^{i'} \pm 2^{j'} \equiv \pm 1 \\ 2^{i'} \pm 2^{j'+k} \equiv \pm 1\end{cases}$.

먼저 식 ①에서 $2^i \pm 2^j \equiv \pm 1$과 $2^i \pm 2^{j+k} \equiv \pm 1$이 동시에 성립하게 되면, $2^i \pm 2^k(\pm 2^i \pm 1) \equiv \pm 1$ 즉, $2^i \pm 2^{k+i} \pm 2^i \equiv \pm 1$이므로, 복호에 따라 8가지 경우가 발생하게 된다. 이들 중 4가지경우는 $0 < i \le v$일 때, $2^i \equiv \pm 1 \mod km+1$이 되므로 모순이다. 따라서 식 ①과 ②에서 다음과 같이 각각 4 가지의 가능성만 남게 된다.

$$\begin{cases}2^i + 2^{k+i} + 2^k \equiv 1 \cdots (a) \\ 2^i + 2^{k+i} - 2^k \equiv -1 \cdots (b) \\ 2^i - 2^{k+i} + 2^k \equiv -1 \cdots (c) \\ 2^i - 2^{k+i} - 2^k \equiv 1 \cdots (d)\end{cases},$$

$$\begin{cases}2^{i'} + 2^{k+i'} + 2^k \equiv 1 \cdots (a') \\ 2^{i'} + 2^{k+i'} - 2^k \equiv -1 \cdots (b') \\ 2^{i'} - 2^{k+i'} + 2^k \equiv -1 \cdots (c') \\ 2^{i'} - 2^{k+i'} - 2^k \equiv 1 \cdots (d')\end{cases}.$$

두 종류의 합동식을 조합하면 모두 16가지의 연립합동식이 생성된다.

첫째, 두 합동식 (a)와 (a')으로 만든 연립합동식 $\begin{cases}2^i(1+2^k) \equiv 1-2^k \\ 2^{i'}(1+2^k) \equiv 1-2^k\end{cases}$ 에서 $2^{i-i'} \equiv 1$을 얻게 되지만 $1 \le i \ne i' \le v$과 유형 II의 최적정규기저의 성질에 모순이다.

둘째, 두 합동식 (a)와 (b')으로 만든 연립합동식 $\begin{cases}2^i(1+2^k) \equiv 1-2^k \\ 2^{i'}(1+2^k) \equiv -(1-2^k)\end{cases}$ 에서 $2^{i-i'} \equiv -1$을 얻게 되지만, 첫째와 마찬가지로 $1 \le i \ne i' \le v$과 유형 II의 최적정규기저의 성질에 모순이다.

셋째, 두 합동식 (a)와 (c')으로 만든 연립합동식 $\begin{cases}2^i(1+2^k) \equiv 1-2^k \\ 2^{i'}(1-2^k) \equiv 1+2^k\end{cases}$ 에서는 $2^{i+i'} \equiv -1$이 되므로 위와 같은 이유로 모순이다.

넷째, 두 합동식 (a)와 (d')으로 만든 연립합동식의 결과는 $2^{i+i'} \equiv 1$이고 모순이다. 결국 16가지 경우 모두 모순이 되는 사실을 확인할 수 있다. 따라서 $(\alpha\alpha_i)^{2^s} = \alpha\alpha_i$이 되는 s는 존재하지 않게 된다.

4.1. 고속 알고리즘

먼저 앞으로의 논의에 필요한 기호를 도입하기로 한다.

정의 1. 행렬 (λ_{ij})의 i번째 행을 $\lambda_i = (\lambda_{i0}, \lambda_{i1}, \cdots, \lambda_{i,m-1})$라 할 때, λ_i를 γ_i번 우측 교대(right shift)한 결과를 $(\lambda_i \rightarrow \gamma_i)$로 표기한다.

그러면 다음의 보조정리를 얻게 된다.

보조정리 2. $1 \le i \le v$에 대하여, γ_i를 행렬 (λ_{ij})의 i번째 행을 우측교대 횟수라고 하자. 만일 행렬 (λ_{ij})의 i번째 행에 γ_i번 우측교대를 시행하여 만든 행렬 (λ'_{ij})에서 1행에서 v행까지는 오직 한 개의 1을 갖는다.

(증명) 위에서 알아 본바와 같이 $l(i)$들은 서로 다른 짝수이다. 또한 행렬 (λ_{ij})의 i번째 행에 γ_i번 우측교대를 시행하면 다음과 같게 된다.

$$(\lambda'_{ij}) = \begin{pmatrix} \lambda_0 \\ (\lambda_1 \rightarrow \gamma_1) \\ \vdots \\ (\lambda_v \rightarrow \gamma_v) \\ \lambda_{v+1} \\ \vdots \\ \lambda_{m-1} \end{pmatrix}$$

$$= \begin{pmatrix} \lambda_0 \\ (\cdots 1\ l(1)\text{개의}\ 0\ 1 \cdots 0) \\ \vdots \\ (\cdots 1\ l(v)\text{개의}\ 0\ 1 \cdots 0) \\ \lambda_{v+1} \\ \vdots \\ \lambda_{m-1} \end{pmatrix}$$

그러면 (λ'_{ij})의 1행에서 v행의 마지막 열의 성분은 모두 0이 되고 $\lambda_0 = (010 \cdots 0)$이므로, γ_0를 $m-2$로 치환하면 행렬 $(\lambda'_{ij})_{1 \le i \le v, 0 \le j \le m-1}$의 각 열에는 꼭 한 개의 1이 남게 된다.

따라서 보조정리 2를 이용하면 $GF(2^m)$에서의 곱셈은 다음과 같이 개선된다.

정리 5. $A, B \in GF(2^m)$의 곱 $C = AB$일 때,

$G_i(A,B) = a_{i+\gamma_0} b_{i+\gamma_0} \alpha_{\gamma_0+1} + \sum_{j=1}^{v} z_{i+\gamma_j, j} \delta^{2^{\gamma_j}}$ 라 놓으

면 $C=((G^2+G_{m-2})^2+\cdots+G_1)^2+G_0$ 이다.

(증명) 정리 4의 계산과정에 보조정리 1,2를 적용하면 C의 계산은 다음과 같다.

$$C=\sum_{i=0}^{m-1}a_ib_i\alpha_{i+1}+\sum_{i=0}^{m-1}\sum_{j=1}^{v}z_{ij}\delta_j^{2^i}$$
$$=\sum_{i=0}^{m-1}a_{i+\gamma_0}b_{i+\gamma_0}\alpha_{i+\gamma_0+1}+\sum_{i=0}^{m-1}\sum_{j=1}^{v}z_{i+\gamma_j}\delta^{2^{i+\gamma_j}}$$
$$=\sum_{i=0}^{m-1}(a_{i+\gamma_0}b_{i+\gamma_0}\alpha_{\gamma_0+1}+\sum_{j=1}^{v}z_{i+\gamma_j,j}\delta^{2^{\gamma_j}})^{2^i}.$$

따라서 $C=((G^2+G_{m-2})^2+\cdots+G_1)^2+G_0$이다.

그런데 정리 5에서

$G_{m-t}(A,B)=G_{m-1}(A^{2^{t-1}},B^{2^{t-1}})$이므로 다음과 같은 곱셈 알고리즘이 성립한다.

Algorithm 1.

INPUT : $A=(a_0,a_1,\cdots,a_{m-1})$,

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

OUTPUT : $C=(c_0,c_1,\cdots,c_{m-1})$.

1. $1\le i\le v$에 대하여, 행렬 $(\lambda'_{ij})_{1\le i\le v,0\le j\le m-1}$의 각 열에 꼭 한 개의 1을 갖도록 γ_i를 정한다.

2. A,B를 m비트 register에 탑재하고,

모든 중간 값 $D_0,D_1,\cdots,D_{m-1}$ 은 0으로 놓는다.

3. $i=0,\cdots,m-1$에 대하여 다음을 실행한다.

3.1. $D=D^2+G_{m-1}(A,B)$.

3.2. $A\leftarrow A^2$, $B\rightarrow B^2$.

4. m번 반복 후에 $1\le i\le m$에 대하여 $AB=\sum_{i=0}^{m-1}c_i\alpha_i$인 $D_i=c_i$를 얻는다.

4.2. 제안하는 직렬 연산기

3.2 절에서와 같이, α를 z^5+z^2+1의 근, $\beta=\alpha^5$, $\{\beta,\ \beta^2,\ \beta^4,\ \beta^8,\ \beta^{16}\}$은 $GF(2^5)$의 정규기저라고 하자. 그러면 보조정리 2에 의하여 곱셈행렬$(\lambda_{5,5})$과 $(\lambda'_{5,5})$는 다음과 같다.

$$(\lambda_{5,5})=\begin{pmatrix}01000\\10010\\00011\\01100\\00101\end{pmatrix}$$

$$(\lambda'_{5,5})=\begin{pmatrix}00001\\10010\\01100\\01100\\00101\end{pmatrix}$$

그러면 $i=0,1,2$에 대하여,

$\gamma_i=\begin{cases}3, i=0;\\0, i=1;\\3, i=2.\end{cases}$ 이고,

$m-1=4$ 이므로, $v=2$이다. 따라서 정리 5에 의하여 $G_4(A,B)=a_2b_2\alpha_4+z_{4,1}\delta_1+z_{2,2}\delta_2^{2^2}$이다.

이와 같이 $GF(2^5)$위에서 Algorithm 1을 적용하여 개선한 직렬 연산기를 구성하면 그의 구조는 그림 2와 같다.

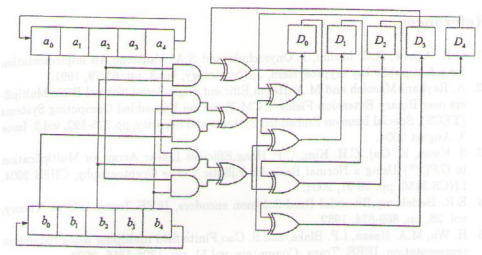

그림 2. 제안하는 직렬 연산기
Fig. 2 Proposed sequential multiplier

V. 복잡도

T_A는 AND delay이고, T_X 는 XOR delay로 표기하고, 세 연산기의 복잡도를 비교하면 다음과 같다.

표 1. 연산기의 복잡도 비교
Table 1. Comparision of complexities

	Path delay	AND	XOR	flip-flop
Agnew	T_A+2T_X	m	2m-1	3m
Reyhani et al.	T_A+3T_X	m	$\frac{3m-1}{2}$	3m
Proposed	T_A+2T_X	m	$\frac{3m-1}{2}$	3m

VI. 결 론

본 논문에서는 유한체에서의 곱셈속도를 줄이기 위해서 원소를 제곱할 때 단순히 좌표의 우측순환이 되는 최적정규기저를 사용하면서 연산 속도를 향상시키기 위한 Algorithm을 개발하여 구축한 새로운 연산기를 제안하였다. 이 연산기는 스마트카드에 사용되는 ECC위에서의 연산 또는 CDMA에서 필요한 트레이스 함수[11]의 H/W 상에서의 실행속도를 줄이는데 도움이 될 것이다. IV장에 간단한 예를 들어 제안한 연산기의 H/W 구현결과를 수록하였으며, 제안한 연산기와 기존의 연산기의 복잡도를 분석한 결과 Agnew의 연산기[2]와 비교하면 Path delay는 동일하지만 XOR gate 수를 줄였으며, Reyhani-Masolleh 과 Hasan의 연산기[4]와 비교하면 Path delay가 줄었으며 XOR gate 수는 동일하였다.

감사의 글

본 논문은 광주교육대학교 2013년도 학술연구비 지원에 의한 것임

참고 문헌

[1] J. Massey, J. Omura, "Computational methd and apparatus for finite field arithmetic", US Patent No. 4587627, 1986.

[2] G. Agnew, R. Mullin, S. Vanstone, "An implementation for a fast public key cryptosystem", Journal of Cryptography, Vol. 3, pp. 63-79, 1999.

[3] Y. Kim, "Efficient Serial Gaussian Normal Basis Multipliers over Binary Extension Fields", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 4, No. 3, pp. 197-203, 2009.

[4] A. Reyhani-Maslleh and M.H. Hasan, "Efficient Digit Serial Normal Basis Multiplier over Binary Extension Fields", ACM Trans. on Embedded Systems and Security Vol. 3, pp. 575-592, 2004.

[5] S.J. Cho, J.G. Kim, U.S. Choi, S.T. Kim, "Cross-correlation of linear and nonlinear GMW-sequences generated by the same primitive polynomial on $GF(2^p)$", The Korea Institute of Electronic Communication Sciences 2011 Spring Conference Vol. 5, No. 1, pp. 155-158, 2011.

[6] Han-Doo Kim, Sung-Jin Cho, Min-Jeong Kwon, Hyun-Ju An, " A study on the cross-correlation function of extended Zeng sequences", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 7, No. 1, pp. 61-67, 2012.

[7] S. Gao Jr. and H.W. Lenstra, "Optimal normal bases", Designs, Codes and Cryptology, Vol. 2, pp. 315-323, 1992.

[8] Y. Kim, "A Fast Multiplier of Composite fields over finite fields", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 6, No. 3, pp. 389-395, 2011.

[9] ANSI, "Public Key Cryptology for the Financial Service Industry : The Elliptic Curve Digital Signature Algorithm(ECDSA)", ANSI x9.62, 1998.

[10] NIST, Digital Signature Standard, FIPS Publication, 186-2, 2000.

[11] U.S. Choi, S.J. Cho, "Design of Binary Sequence with optimal Cross-correlation Values", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 6, No. 4, pp. 539-544, 2011.

저자 소개

김용태(Yong-Tae Kim)

1976년 2월 공주사범대학 수학교육과(이학사)

1986년 2월 고려대학교 대학원 수학과(이학석사)

1991년 2월 고려대학교대학원 수학과(이학박사)

2000년 8월 서울대학교 대학원 수학교육과(교육학석사)

2008년 2월 서울대학교 대학원 수학교육과(박사과정 수료)

1992년 3월~현재 광주교육대학교 수학교육과 교수

※ 관심분야 : ECC, 정수론적 암호학, 공개키암호학