Regular paper http://dx.doi.org/10.13067/JKIECS.2015.10.3.387

유한체위에서의 근점기저를 이용한 고속 타원곡선 암호법

김용태*

Fast Elliptic Curve Cryptosystems using Anomalous Bases over Finite Fields

Yong-Tae Kim*

요 약

유한체위에서 ECC를 기반으로 하는 전자상거래 또는 비밀통신에서 송수신자가 서로 다른 기저를 사용하는 경우에는 기저변환으로 인한 통신지연이 발생하게 된다. 본 논문에서는 서로 다른 기저를 사용하는 H/W와 S/W 구현 시스템 사이의 비밀통신 또는 전자서명에 소요되는 기저변환의 횟수를 분석하여, 그로 인한 통신지연을 제거하기 위해서, All One Polynomial(AOP)을 사용하는 유한체위에서 하드웨어와 소프트웨어 구현 모두에 효과적이면서, 기저변환이 필요 없는 근점 기저를 소개하였다. 제안하는 근점기저를 사용한 곱셈기의 H/W 구현 결과, 삼항식과 다항식기저를 사용하는 곱셈기보다 연산 시간이 약 25% 감소하였다.

ABSTRACT

In Electronic Commerce and Secret Communication based on ECC over finite field, if the sender and the receiver use different basis of finite fields, then the time of communication should always be delayed. In this paper, we analyze the number of bases-transformations needed for Electronic Signature in Electronic Commerce and Secret Communication based on ECC over finite field between H/W and S/W implementation systems and introduce the anomalous basis of finite fields using AOP which is efficient for H/W, S/W implementation systems without bases-transformations for Electronic Commerce and Secret Communication. And then we propose a new multiplier based on the anomalous basis of finite fields using AOP which reduces the running time by 25% than that of the multiplier based on finite fields using trinomial with polynomial bases.

키워드

ECC, AOP, Polynomial Basis, Anomalous Basis
타원곡선 암호법, 모든 계수가 1인 다항식, 다항식 기저, 근점 기저

I. 서 론

타원곡선 암호법(ECC)은 RSA나 ElGamal 암호법에 비하여 1/6정도의 키 크기로 같은 안전도를 제공하므로 메모리 용량이나 프로세서의 파워가 제한된 스마트카드 등에 더욱 효과적이기 때문에 정보공학이나 전자상거래 등에 널리 사용되고 있는 공개키 암호계이다. 타원곡선 암호법에서는 주로 유한체인

* 교신저자(corresponding author) : 광주교육대학교 수학교육과(ytkim@gnue.ac.kr)
접수일자 : 2015. 02. 12 심사(수정)일자 : 2015. 03. 13 게재확정일자 : 2015. 03. 23

$GF(2^n)$ 나 $GF(p)$ (p는 홀수인 소수)를 사용하고 있다. 유한체의 원소를 표현하는 대표적인 방법으로는 다항식 기저[1-2], 정규기저[3-4]등을 사용하거나 비관용(non-convention)기저[5]를 사용한다. 또한 유한체 $GF(2^m)$ 에서 기저표현에 필요한 이진 수열을 생성하는 방법에 관한 연구[6-7]가 최근에 진행되고 있어 기저를 효율적으로 표현하는 데에 인용되고 있다. 그런데 유한체의 연산은 기저의 선택에 따라 연산속도의 차이가 매우 크다. 즉, 유한체의 다항식기저는 S/W 구현에 용이하며 정규기저는 H/W 구현에 효과적이다. 1995년에 Schroeppel 등[8]이 키 교환 상호통신 방법을 제안한 방법에서는 H/W 시스템과 S/W 시스템 간의 유한체의 원소를 표현하는 방법이 다르기 때문에 상호통신을 할 경우에는, 반드시 기저변환을 해야만 하므로, 타원곡선 암호시스템을 이용한 통신 속도를 지연하는 원인이 되어왔다. 이러한 단점을 보완하기 위해서, SAC'98에서 Kaliski등[9]은 메모리를 줄인 효과적인 기저변환 방법을 제안하였으며, 이 방법에 의하면 한번의 기저변환에 따른 복잡도가 기저의 선택에 따라 타원곡선군의 상수곱의 10~20%에 해당한다는 것을 밝혔다. 그러나 이 방법 역시 유한체 $GF(2^n)$ 의 기저변환 행렬($n\times n$ 행렬)과 그 역행렬을 저장할 공간이 필요하게 되어, 효과적인 타원곡선 암호법의 특성에 치명적인 단점이 되고 있다. 본 논문에서는 서로 다른 기저를 사용하는 H/W와 S/W 구현 시스템 사이의 비밀통신 또는 디지털 서명의 불편함을 알아보기 위해서, 소요되는 기저변환의 횟수를 계산하였으며, 그로 인한 통신지연을 제거하기 위하여, H/W와 S/W 구현에 모두 효과적이고 기저변환이 필요 없는 근점(anomalous)기저를 소개하고, 근점기저를 이용하는 S/W 구현 과정과 이를 적용한 H/W 설계 과정을 구체적으로 설명한 다음, 설계에 따른 근점기저 곱셈기를 제안하고, 제안한 곱셈기와 삼항식을 적용한 유한체위에서의 다항식기저에 기반한 곱셈기와의 구현 결과 비교 표를 제시하기로 한다.

II. 타원곡선 암호시스템 분석

이 장에서는 서로 다른 기저를 사용한 두 타원곡선 암호시스템 사이에서 비밀통신과 디지털 서명이 수행될 때 필요한 기저변환의 횟수를 계산하여 통신지연시간을 단축시킬 방법을 제시하기로 한다. 유한체의 연산에 관한 내용은 Menezes 등[10]을 참조하기로 한다.

2.1. 비밀통신

비밀통신 방법을 American National Standard X9.63의 표준을 기준으로 제시하면 다음과 같다.

앨리스(A)가 밥(B)에게 비밀스럽게 메시지 M을 전달하기를 원하는 경우에 다음과 같이, A와 B가 주어진 유한체의 기저가 같은 경우와 다른 경우에 거치게 되는 절차는 각각 다음과 같다.

2.1.1 기저가 같은 경우

a. Setup

1. $E(a,b): y^2+xy=x^3+ax^2+b$ 는 유한체 $K=GF(2^n)$ 위에 주어진 타원곡선
2. P 는 $E(K)$ 에서 소수 위수 N을 갖는 점

b. 키 생성

1. $1<d<N$ 인 정수 d를 선택
2. $Q=dP$ 를 계산
3. B는 d를 비밀키로 하고, P, Q는 공개키로 공개

c. 암호화

1. $1<k<N$ 를 만족하는 k를 임의로 선택
2. $P'=kP,\ Q'=kQ,\ C=Q'\oplus M$ 을 계산
3. B에게 (P', C)를 전송

d. 복호화

1. $Q'=dP'$ 를 계산
2. $M=Q'\oplus C$를 계산

2.1.2 기저가 다른 경우

A와 B는 각각 서로 다른 기저 B_0와 B_1을사용하며, B_0에서 B_1으로 기저변환 행렬을 Γ 라 하면, 사용자 B는 기저 B_1을 사용하므로 B가 사용하는 타원곡선 $\Gamma(E(a,b))$ 를 편의상 $\overline{E(a,b)}$ 로 표현하기로 한다. 따라서 $\Gamma(P)=\overline{P}$, $\Gamma(Q)=\overline{Q}$ 로 나타내기로 한다.

d′. 복호화

1. P'을 $\overline{P'}$ 로 변환하고 $d\overline{P'}$ 를 계산

2. Γ^{-1}를 이용하여 $d\overline{P}$을 $Q' = dP'$로 변환
3. $M = Q' \oplus C$를 계산

그런데 메시지 M을 복원하기 위하여 6번의 유한체 원소의 변환이 필요하다. 유한체 $GF(2^n)$에서 기저변환과 곱셈은 각각 $2n^2 - n$, n^2번의 비트 연산이 필요하다. 그러므로 다른 기저를 사용한 비밀통신을 수행할 때, 같은 기저를 사용할 때보다 12번의 유한체의 곱셈연산과 Γ 와 Γ^{-1}를 저장하는 공간이 더 필요하게 된다.

2.2. 디지털 서명

디지털 서명이 필요한 암호계에서, A가 B에게 메시지 M과 서명 (r,s)를 보내기를 원하는 경우에도 비밀통신과 마찬가지로, A와 B가 주어진 유한체의 기저가 같은 경우와 다른 경우에 거치게 되는 절차는 각각 다음과 같다.

2.2.1 기저가 같은 경우

Setup과 키 생성 과정은 비밀통신의 경우와 같다.

c. 서명 생성
1. $1 < k < N$ 를 만족하는 k를 선택
2. $kP = (x_1, y_1)$ 을 계산
3. $r = x_1 \bmod N$을 계산
4. 해쉬함수 H를 이용하여 $e = H(M)$ 을 계산
5. $s = k^{-1}(e + dr) \bmod N$ 을 계산
6. (r,s)와 M을 A에게 전송

d. 서명 확인
1. $e = H(M)$ 을 계산
2. $s^{-1} \bmod N$을 계산
3. $u_1 = es^{-1} \bmod N$과 $u_2 = rs^{-1} \bmod N$ 을 계산
4. $u_1 P + u_2 Q =$ (x_1, y_1)를 계산
5. $v = x_2 \bmod N$ 을 계산
6. $v = r$인 경우 서명 확인

2.2.2 기저가 다른 경우

비밀통신의 경우와 같이 기저와 변환행렬을 사용하고, 서명 (r,s)를 확인하기 위하여는 비밀통신과 마찬가지로 Γ를 이용하여 $E(a,b)$, P, Q를 $\overline{E(a,b)}$, $\overline{P}, \overline{Q}$ 로 변환해야 한다.

d′ 서명 확인
1. $e = \mathrm{H(M)}$ 을 계산
2. $s^{-1} \bmod N$을 계산
3. $u_1 = es^{-1} \bmod N$과 $u_2 = rs^{-1} \bmod N$ 을 계산
4. $u_1 \overline{P} + u_2 \overline{Q} = (\overline{x_2}, \overline{y_2})$를 계산
5. Γ^{-1}을 이용하여 $\overline{x_2}$를 x_2로 변환
6. $v = x_2 \bmod N$ 을 계산
7. $v = r$인 경우 서명 확인

따라서 기저가 다른 경우는 같은 경우보다 14번의 유한체의 곱셈연산을 더 수행해야 하며, 기저변환 행렬 Γ를 저장하는 공간도 필요하게 된다.

타원곡선 암호시스템을 구현할 때, 시스템의 속도는 중요한 문제이다. 그런데 유한체 $GF(2^n)$ 상에서는 H/W는 정규기저를 사용할 때, S/W는 다항식기저를 사용할 때 가장 효과적으로 구현되기 때문에, 서로 다른 기저를 사용하는 타원곡선 암호시스템 사이의 통신이 불가피하게 되므로, 기저변환에 따른 암호시스템의 통신 속도가 지연되는 원인이 된다.

III. 근점 기저 곱셈기

II장에서는 A와 B가 유한체의 서로 다른 기저를 사용할 때 발생하는 단점을 알아보았다. 이 장에서는 A와 B가 유한체의 서로 다른 기저를 사용하여 비밀통신을 행할 때 발생하는 유한체의 원소의 곱셈 12번, 서명 확인때에 발생하는 14번의 곱셈을 줄이면서 동시에 기저변환행렬을 저장하지 않아도 되고, 휴대전화기, 스마트카드, 휴대용 컴퓨터(HPC)등에 사용되는 S/W와 H/W에서 효율적으로 구현되는 근점기저를 제안하고, 유한체위에서 근점기저를 이용한 ECC의 구축과정을 제시하며, 그에 따른 근점기저 곱셈기를 제안하고, 근점기저 곱셈기와 다항식기저 곱셈기의 구현결과를 비교하여 제시하기로 한다.

정의 1. 0이 아닌 $\alpha \in GF(2^n)$ 에 대하여, $GF(2^n)$

의 부분집합 $B=\{\alpha,\alpha^2,\cdots,\alpha^n\}$ 을 $GF(2)$ 위에의 $GF(2^n)$ 의 근점기저라고 한다.

만일 f가 $GF(2)$ 위에서 n차의 모닉(monic)인 기약다항식이고 $f(\alpha)=0$ 이면 집합 $B=\{\alpha,\alpha^2,\alpha^3,\cdots,\alpha^n\}$은 $GF(2)$ 위에서 일차독립인 것은 잘 알려진 사실이다. B는 다항식 기저 $\{1,\alpha,\cdots,\alpha^{n-1}\}$와 다르고, f 가 모든 계수가 1인 다항식(AOP) $x^n+x^{n-1}+\cdots+x+1$ 일 때에는 정규기저 $\{\alpha,\alpha^2,\alpha^{2^2},\cdots,\alpha^{2^{n-1}}\}$와 집합으로는 같지만, 기저의 성분의 순서가 다르므로 서로 다른 기저이다. AOP가 기약다항식이 되는 차수 n은 162, 172, 178, 180, 196 등이 있다. 이제 f가 AOP 인 경우에, 근점 기저를 이용하여 S/W와 H/W를 구현하는 과정을 설명하기로 한다.

3.1. S/W 구현

α가 n차 기약다항식 AOP의 근이고 $B=\{\alpha,\alpha^2,\alpha^3,\cdots,\alpha^n\}$를 $GF(2)$ 위에서 $GF(2^n)$ 의 근점기저라 하자. $GF(2^n)$ 의 원소 **a** 를 근점기저를 이용하여 나타내면 식 (1)과 같다.

$$\mathsf{a}=\sum_{i=1}^{n}a_{i-1}\alpha^i,\quad a_i\in GF(2). \tag{1}$$

또는 근점기저의 성분인 α^i의 계수만을 이용하여 n차원 벡터형태인 다음의 식 (2)와 같이 표현된다.

$$\mathsf{a}=(a_0,a_1,\cdots,a_{n-1}) \tag{2}$$

$GF(2^n)$ 상에서, AOP가 기약다항식이면 n은 짝수이므로 $n=2m$ 으로 놓으면, $\alpha^{n+1}=1$ 이므로 **a**의 제곱은 다음과 같이 계산된다.

$$\begin{aligned}\mathsf{a}^2&=(\sum_{i=1}^{n}a_{i-1}\alpha^i)^2\\&=\sum_{i=1}^{2m}a_{i-1}\alpha^{2i}\\&=\sum_{i=1}^{m}a_{i-1}\alpha^{2i}+\sum_{i=m+1}^{2m}a_{i-1}\alpha^{2i}\\&=\sum_{i=1}^{m}a_{i-1}\alpha^{2i}+\sum_{i=1}^{m}a_{m+i-1}\alpha^{2i-1}.\end{aligned} \tag{3}$$

예를 들어 $n=10$ 인 경우, 즉 $\mathsf{a}=(a_0,a_1,\cdots,a_9)$ 이라면,

$$\mathsf{a}^2=(a_5,a_0,a_6,a_1,a_7,a_2,a_8,a_3,a_9,a_4) \tag{4}$$

이 된다. 즉, 근점기저를 이용하면 제곱연산은 단순히 좌표의 순환이동(Cyclic Shift)이므로 시간 소요가 거의 없다.

다항식 기저로 표현한 유한체 원소의 역원을 구하는 가장 개선된 알고리즘은 Almost Inverse 알고리즘[3]으로 알려져 있다. 이 알고리즘을 근점기저에 적용하여 유한체의 0이 아닌 원소의 역원을 구하는 과정은 다음과 같다.

$$GF(2^n)\cong GF(2)[x]/(f) \tag{5}$$

이므로 차수가 n보다 작은 0이 아닌 다항식 $A(x)$의 역원 $C(x)$는 다음과 같이 표현된다.

$$A(x)C(x)\equiv 1 \bmod f,\ \deg(C(x))<n. \tag{6}$$

Almost Inverse 알고리즘은

$$A(x)B(x)\equiv x^k \bmod f, \deg(B(x))<n, 0\le k. \tag{7}$$

를 만족하는 $B(x)$ 와 k를 구하여 $A(x)$의 역원을 구하는 알고리즘이다. 또한 $GF(2^n)$ 의 0이 아닌 원소 $A(x)$를 근점 기저 $\{x,x^2,\cdots,x^n\}$으로 표현하면 $A(x)$의 차수는 n보다 작거나 같고 상수항은 존재하지 않는다. 즉,

$A(x)=x^{k_0}\cdot A_1(x), k_0>0$ 이라 놓으면 $A_1(x)$의 상수항은 1이다. 따라서 Almost Inverse 알고리즘을 이용하여

$$A_1(x)B_1(x)\equiv x^{k_1} \bmod f \tag{8}$$

를 만족하는 $B_1(x)$와 k_1을 구한다. 그러면 $A(x)$의 역원은 $B_1(x)$를 $x^{k_0+k_1}$로 나눈 다음, 상수항을 소거하여 구할 수 있다.

3.2. 근점기저 곱셈기의 구성

식 (1)과 같이 근점기저로 표현된 유한체의 두 원소 **a, b**의 곱을 행렬을 사용하여 표현하면 다음과 같다.

$$a \times b=(C+D)[b]_B=E, \quad (9)$$

단,

$$C=\begin{pmatrix} 0 & a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 \\ a_0 & 0 & a_{n-1} & a_{n-2} & \cdots & a_2 \\ a_1 & a_0 & 0 & a_{n-1} & \cdots & a_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & a_{n-4} & a_{n-5} & \cdots & 0 \end{pmatrix}, \quad (10)$$

$$D=\begin{pmatrix} a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_0 \\ a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_0 \\ a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_0 \end{pmatrix}, \quad (11)$$

그리고 $[b]_B = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix}$ 이다.

따라서 만일 $C[b]_B = (c_0,c_1,c_2,\cdots,c_{n-1})$, $D[b]_B = (d,d,d,\cdots,d)$, $E = (e_0,e_1,e_2,\cdots,e_{n-1})$로 놓으면, 근점기저를 사용한 곱셈기의 기본 구조는 그림 1과 같으며, 기본구조를 바탕으로 구축한 근점기저 곱셈기의 구조는 그림 2와 같다.

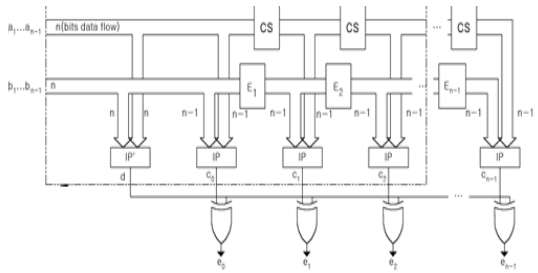

그림 1. 근점기저 곱셈기의 기본 구조
Fig. 1 Fundamental Structure of Anomalous Basis Multiplier

곱셈기의 기본구조는 내적기(IP), 순환좌표이동기(CS) 와 교환기(E_i)로 구성되어 있다.

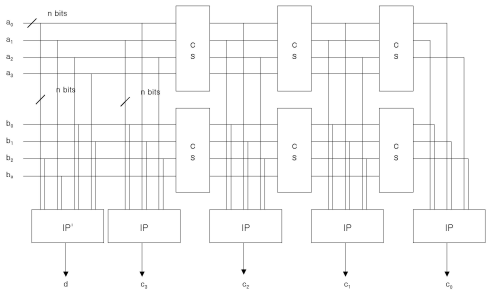

그림 2. 근점기저 곱셈기의 구조
Fig. 2 The Structure of Anomalous Basis Multiplier

특히, 제곱연산은 식(3)과 (4)에 의하여 게이트와 시간 연기(Delay)가 없는 단순교환(rewiring)인 것을 쉽게 확인할 수 있다.

3.3. 복잡도

제안하는 곱셈기는 내적(Inner Product : IP), 순환이동(Cyclic Shift : CS), 덧셈기(Adder)로 구성되어있다. IP는 $GF(2^n)$ 위에서 m 개의 벡터의 내적으로 구성되어있다. CS 는 좌표의 단순교환으로 구성되어 있으며, 덧셈기는 $GF(2^n)$ 의 덧셈(XOR)으로 구성되어 있다. 부분체 $GF(2^n)$ 에서 곱셈에 필요한 연산은 n^2 AND gates, n^2-1 XOR gates와 $D_A+(1+\lceil \log_2 n \rceil)D_X$ Delays가 필요하다. 그리고 식 (4)에서 알 수 있듯이 제곱연산은 단순한 좌표의 순환이동이므로 소요시간이 적다.

근점기저를 이용한 우리의 곱셈기와 Win등[11]이 제시한 삼항식으로 구축한 유한체위에서 다항식 기저를 사용한 곱셈기의 복잡도를 표 1에 제시하였으며, 복잡도는 우리의 곱셈기가 Win등의 곱셈기보다 25% 정도 개선됨을 보였다.

표 1. 유한체의 연산에 대한 공간과 시간 복잡도
Table 1. Complexity of Our Multiplier

	Mult.	Sq.
Anom.	n^2 AND, n^2-1 XOR gates, $D_A+(1+\lceil \log_2(n-1) \rceil)D_X$ Delays	Rewiring
Poly.	n^2 AND, n^2-1 XOR gates, $D_A+(2+\lceil \log_2(n-1) \rceil)D_X$ Delays	n-1 XOR gates, $1D_X$ Delay

3.4. H/W 구현의 예

이 절에서는 $GF(2^{180})$ 위에서 근점기저를 사용하여 Pentium 166 MHZ CPU에서 C-언어 프로그램을 사용한 H/W 구현 결과와 Win등[11]이 $GF(2^{177})$ 에서 삼항식과 다항식 기저를 이용하여 구현한 결과를 실행시간을 마이크로초(μsecond)를 단위로 계산하여 표 2에 정리하였으며, 근점기저 곱셈기가 다항식 기저 곱셈기보다 계산 속도가 25% 개선되었다.

표 2. H/W 구현 결과 비교표
Table 2. Comparison of Complexities of Multipliers. (unit : μsec)

	Anomalous Basis	Poly. Basis using Trinomial
Mult.	51.5	71.8
Sq.	1.5	2.7
Inv.	161.4	225

V. 결 론

본 논문에서는, 유한체위에서 ECC를 사용하는 비밀통신과 전자상거래에서 송수신자가 서로 다른 기저를 사용할 때 연산속도가 지연되는 원인을 제거하기 위해서, 메시지 송수신 과정에서 기저변환이 필요 없는 근점기저 곱셈기를 제안하였으며 H/W 구현한 결과를 다항식 기저를 사용한 곱셈기와 비교하여 제시하였다. 그 결과로, 효율적인 이진수열 생성방법[12]을 적용하여, 근점기저를 사용한 우리의 곱셈기가 다항식 기저를 사용한 곱셈기보다 복잡도가 25% 개선됨을 보였다.

감사의 글

본 논문은 광주교육대학교 2015년도 학술연구비 지원에 의한 것임

References

[1] H. Wu and M.A. Hasan, “Low Complexity bit-parallel multipliers for a class of finite fields,” *IEEE Trans. Computers,* vol. 47, no. 8, 1998, pp. 883-887.

[2] A. Reyhani-Maslleh and M. H. Hasan, “Efficient Digit Serial Normal Basis Multiplier over Binary Extension Fields,” *ACM Trans. Embedded Systems and Security,* vol. 3, 2004, pp. 575-592.

[3] B. Sunar and C. K. Koc, “An efficient optimal normal basis type II multiplier,” *IEEE Trans. Computers,* vol. 50, no. 1, 2001, pp. 83-88.

[4] C. C. Wang, T. K. Truong, H. M. Shao, L. J. Deutsch, J. K. Omura, and L. S. Reed, “VLSI architecture for computing multiplications and inverses in $GF(2^m)$,” *IEEE Trans. Computers,* vol. 34, no. 8, 1985, pp. 709-716.

[5] C. Kim, S. Oh, and J. Lim, “A new hardware architecture for operations in $GF(2^m)$,” *IEEE Trans. Computers,* vol. 51, no. 1, 2002, pp. 90-92.

[6] S. Cho, J. Kim, U. Choi, and S. 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 Conf. June,* vol. 5 no. 1, Pusan, Korea 2011, pp. 155-158.

[7] H. Kim, S. Cho, M. Kwon, and H. An, “A study on the cross sequences,” *J. of the Korea Institute of Electronic Communication Sciences,* vol. 7, no. 1, 2012, pp. 61-67.

[8] R. Schroeppel, H. Orman, S. O’Malley, and O. Spatscheck, "Fast key exchange with elliptic curve systems,” *Crypto ‘95,* Santa Barbara, CA, *LNCS 963, Springer-Verlag*, Aug. 1995, pp. 43-56.

[9] B. S. Kaliski Jr. and Y. L. Yin, "Storage-Efficient Finite Field Basis Conversion,” *SAC’ 98, ACM Symp. on Applied Computing,* Atlanta, GA, Aug. 1998.

[10] A. J. Menezes, *Applications of finite fields.* Kluwer Academic Publishers, Massachusetts, 993.

[11] E. De Win, A. Bosselaers, S. Vandenberghe, P.

De Gersem, and J. Vandewalle, "A fast software implementation for arithmetic operations in $GF(2^t)$," *Asiacrypt'96,* Kyong Ju, South Korea, *LNCS 1163,* Springer-Verlag, Nov. 1996, pp 65-76.

[12] U. Choi and S. Cho, "Design of Binary Sequence with optimal Cross-correlation Values," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 6, no. 4, 2011, pp. 539-544.

저자 소개

김용태(Yong-Tae Kim)

1976년 2월 공주사범대학 수학교육과(이학사)

1986년 2월 고려대학교 대학원 수학과(이학석사)

1991년 2월 고려대학교 대학원 수학과(이학박사)

2000년 8월 서울대학교 대학원 수학교육과(교육학석사)

2008년 2월 서울대학교 대학원 수학교육과(박사과정수료)

1992년 3월~현재 광주교육대학교 수학교육과 교수

※ 관심분야 : ECC, 정수론적 암호학, 공개키암호학