通信情報保護論文誌
第9券 第4號 1999. 12

# 변형된 다항식 기저를 이용한 유한체의 연산

김 창 한*, 이 성 재**, 박 성 준**

# Arithmetic of finite fields with shifted polynomial basis

Chang Han Kim*, Sung Jae Lee**, Sung Jun Park**

## 요약

유한체(Galois fields)가 타원곡선 암호법, coding 이론 등에 응용되면서 유한체의 연산은 더 많은 관심의 대상이 되고 있다. 유한체의 연산은 표현방법에 많은 영향을 받는다. 즉, 최적 정규기저는 하드웨어 구현에 용이하고 Trinomial을 이용한 다항식 기저는 소프트웨어 구현에 효과적이다. 이 논문에서는 새로운 변형된 다항식 기저를 소개하고 AOP를 이용한 경우 하드웨어 구현에 효과적인 최적 정규기저와의 변환이 위치 변화로 이루어지고 또한 이것을 바탕으로 한 유한체의 연산이 소프트웨어적으로 효율적임을 보인다.

## ABSTRACT

More concerns are concentrated in finite fields arithmetic as finite fields being applied for Elliptic curve cryptosystem, coding theory and etc. Finite fields arithmetic is affected in represen -tation of those. Optimal normal basis is effective in hardware implementation and polynomial basis with trinomial in software. In this paper, we introduce a new shift polynomial basis of finite field which is effective in the basis conversion with optimal normal basis , and show that the arithmetic of finite field with the basis is efficient in software implementation.

Keyword : Finite fields, ECC, All one polynomials

---

* 세명대학교 컴퓨터응용수학과(chkim@venus.semyung.ac.kr)

** 한국정보보호센터

이 논문은 한국정보보호센터 연구비 지원에 의해 수행된 연구임

## I. 서 론

유한체의 연산은 타원곡선 암호법과 관련하여 연구의 관심이 집중되고 있다[2,7,9,11]. 특히 1985년에 Miller[11]와 Koblize[7]에 의하여 타원곡선 암호법이 제안되면서 타원곡선 암호법의 바탕연산이 유한체의 연산인 관계로 더욱 유한체의 연산에 관한 연구가 활발히 진행되고 있다. 유한체의 연산은 유한체의 표현방법 즉, 기저가 무엇인지, 유한체를 구성하는 기약다항식이 무엇인지에 의하여 많은 영향을 받는다. 1992년에 Harper등[3]은 합성체(composite field)를 이용한 유한체의 연산 방법을 제안하였다. 즉 $GF(2^{104})$를 $GF(2^8)$위에서의 13차 확장체로 보고 연산을 하는 방안을 제시하였다. 1995년에 Schroeppel등[12]은 Trinomial과 Almost Inverse Algorithm(AIA)을 이용한 유한체의 연산 방법을, 1996년에는 Win등[13]은 합성체와 Trinomial를 이용한 유한체의 연산 방법을 제안하였다. 또한 97년에는 Guajardo등[2]은 합성체의 연산을 Karatsuba-Ofman Algorithm(KOA)을 이용한 곱셈과 Itoh 와 Tsujii[5]의 알고리즘을 이용한 역원 계산 방법을 제안하였고 98년에는 Baily등[1]이 p가 워드 사이즈( $2^{16}, 2^{32}, 2^{64}$) 보다 조금 작은 소수인 $GF(p^n)$의 연산을 Binomial( $x^n - w$)을 이용하여 구현하는 효율적인 방법을 제시하였다. 이상의 유한체의 연산은 다항식 기저를 바탕으로 하는 소프트웨어적인 구현 방법이다.

유한체의 하드웨어구현은 표수가 2인 경우가 합리적이고, 특히 최적 정규기저 I형태를 이용한 방법을 Hasan등[4]이 제안한 이후 더 효과적인 방법은 제시되지 않고 있으며 비슷한 난이도를 갖는 방법이 Koc등[8]에 의하여 제안되었다. 이 논문은 표수가 2인 유한체의 연산에 초점을 두었다.

타원곡선 암호시스템은 하드웨어 와 소프트웨어로 구현이 가능하고, 각각 효과적인 기저가 다른 관계로 기저변환을 하여야 하고, 기저변환은 변환행렬에 의하여 이루어진다 [14]. 이 경우 기저변환 행렬을 저장해야 하고, 이것은 핸드폰, 스마트 카드등 제한된 조건하의 타원곡선 암호구현의 제약조건이 된다. 더욱이 컴퓨팅 파워가 증대되면서 키 사이즈가 증대될 경우 이 문제는 더 심각해 질 수 있다. 한편 98년에 Kaliski등[6]은 저장 용량을 줄이는 기저 변환 방법을 제안하였다.

이 논문에서는 AOP를 이용한 변형된 다항식 기저를 이용한 유한체의 연산이 소프트웨어적으로 효과적임을 보이고 이 기저가 최적 정규 기저로 위치변화로 기저변환이 이루어지므로 이 기저는 타원곡선 암호법의 하드웨어 구현과 소프트웨어 구현이 다 필요한 경우 하드웨어는 최적 정규기저로 소프트웨어는 제안된 기저로 구현할 경우 제한된 조건하에서 효과적으로 구현할 수 있음을 보인다.

이 논문은 5장으로 구성되었다. 2장에서는 유한체의 이론을 설명하고, 3장에서는 변형된 다항식 기저를 이용한 유한체의 연산법을, 4장에서는 구현 결과를, 5장에서는 관련 결과를 소개한다.

## II. 유한체 이론

차수가 $n$ 인 다항식 $f(x)$가 $GF(2)[x]$에서 기약이면 $2^n$개의 원소를 갖는 유한체는

$$GF(2^n) \cong GF(2)[x]/(f(x))$$

이다. 즉 $f(\alpha) = 0$ 이면

$$GF(2^n) = \{a_0 + a_1\alpha + \cdots + a_{n-1}\alpha^{n-1} | a_i \in GF(2)\}.$$

이와 같이 표현하는 것을 다항식 기저를 이용한 유한체의 표현이라 한다.

**정의 2.1** $f(x) = 1+x+x^2+x^3+\cdots+x^n$를 차수가 n인 All-One-Polynomial(AOP)이라 한다.

**정리 2.2** All-One-Polynomial $f(x) = 1+x+x^2+x^3+\cdots+x^n$ 가 $GF(2)$ 위에서 기약이기 위한 필요충분조건은 n+1 이 소수이고 2 가 유한체 $Z_{n+1}$ 의 원시원(primitive element)인 것이다[10, p.98].

$f(x) = 1+x+x^2+x^3+\cdots+x^n$ 가 $GF(2)$위에서 기약 다항식이면 n 은 짝수이고 이것을 만족하는 n 은 다음과 같은 것들이 있다. 162, 172, 178, 180, 196,… [10, p.100].

$f(x)$가 차수가 n 이고 $GF(2)$위에서 기약인 All-One-Polynomial 이면 다음 보조정리가 성립한다.

**보조정리 2.3** $f(\alpha) = 0$ 이면 $\alpha^{n+1} = 1$ 이다.

**증명.** $f(\alpha) = 1+\alpha+\alpha^2+\cdots+\alpha^n = 0$ 에 의하여

$\alpha+\alpha^2+\alpha^3+\cdots+\alpha^n+\alpha^{n+1} = 0$

이므로

$\alpha^{n+1} = \alpha+\alpha^2+\alpha^3+\cdots+\alpha^n = 1$

이다.

**보조정리2.4**

$g(x) = b_0+b_1x+\cdots+b_{m-1}x^{m-1}+x^m$를 $GF(2)[x]$에서 기약 다항식이라 하자. 그러면 $g(\alpha) = 0$이면

$GF(2^m) \cong GF(2)[x]/(g(x))$

이고 $B = \{\alpha, \alpha^2, \cdots, \alpha^m\}$는 $GF(2)$위에서 $GF(2^n)$ 의 기저이다.

**증명.** $g(x)$가 $GF(2)$위에서 기약이므로

$GF(2^n) \cong GF(2)[x]/(g(x))$.

$a_1\alpha+a_2\alpha^2+\cdots+a_n\alpha^m = 0$ 라 하면

$\alpha^m = b_0+b_1\alpha+\cdots+b_{m-1}\alpha^{m-1}$

이므로

$$\begin{aligned} & a_1\alpha+a_2\alpha^2+\cdots+a_m\alpha^m \\ = \; & a_mb_0+(a_1+a_mb_1)\alpha+(a_2+a_mb_2)\alpha^2 \\ & +\cdots+(a_{m-1}+a_mb_{m-1})\alpha^{n-1} \\ = \; & 0 \end{aligned}$$

이다. 또한 $\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$는 기저이고 $b_0 = 1$ 이므로

$a_1 = a_2 = \cdots = a_m = 0$

이다.

$f(x)$가 기약인 n차의 AOP이고 $f(\alpha) = 0$이면 $\{\alpha, \alpha^2, \cdots, \alpha^n\}$는 GF(2) 위에서 $GF(2^n)$의 기저이다.

**정의 2.5.** $B = \{\alpha, \alpha^2, \cdots, \alpha^n\}$ 가 GF(2) 위에서 $GF(2^n)$의 기저일 때 B를 변형된 다항식 기저라 한다.

최적 정규기저를 사용하는 유한체의 연산은 하드웨어 구현에 매우 효율적이고[4] AOP 다항식은 I타입의 최적 정규기저를 형성한다[10].

**보조정리 2.6** $f(x) = 1+x+\cdots+x^n$가 $GF(2)[x]$에서 기약 다항식이고 $f(\alpha) = 0$ 이면 집합적으로

$\{\alpha, \alpha^2, \cdots, \alpha^n\} = \{\alpha, \alpha^2, \cdots, \alpha^{2^{n-1}}\}$

이다 [10, p.97].

이 보조정리는 정리2.2에 의하여 얻어진다.

**예제 2.7** n=10이면 f(x)는 기약다항식이다. 2는 $Z_{11}$의 원시근이다. 즉

$2^0=1, 2^1=2, 2^3=8, 2^4=5, 2^5=10, 2^6=9,$

$2^7=7, 2^8=3, 2^9=6$ 이다. 따라서

$\{\alpha, \alpha^2, \alpha^{2^2}, \alpha^{2^3}, \alpha^2, \alpha^{2^4}, \alpha^{2^5}, \alpha^{2^6}, \alpha^{2^7}, \alpha^{2^8}, \alpha^{2^9}\}$

$= \{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^5, \alpha^{10}, \alpha^9, \alpha^7, \alpha^3, \alpha^6\}$.

**참고 2.8** 보조정리 2.6에 의하여 정규기저와 변형된 다항식 기저의 기저변환은 위치변화만으로 이루어지는 것을 알 수 있다.

## III. 변형된 다항식 기저를 이용한 유한체 $GF(2^n)$의 연산

$f(x) = 1+x+\cdots+x^n$가 $GF(2)[x]$에서 기약 다항식이고 $f(\alpha) = 0$이면 $B = \{\alpha, \alpha^2, \cdots, \alpha^n\}$는 $GF(2)$위에서 $GF(2^n)$의 변형된 다항식 기저이다. 그러면 $GF(2^n)$ 의 원소 $a, b$ 는 다음과 같이 표현된다.

$$\begin{aligned} a &= a_0\alpha + a_1\alpha^2 + \cdots + a_{n-1}\alpha^n \\ &\equiv (a_0, a_1, \cdots, a_{n-1}), \quad a_i \in GF(2) \end{aligned}$$

$$\begin{aligned} b &= b_0\alpha + b_1\alpha^2 + \cdots + b_{n-1}\alpha^n \\ &\equiv (b_0, b_1, \cdots, b_{n-1}), \quad b_j \in GF(2). \end{aligned}$$

따라서

$$a+b = (a_0+b_0, a_1+b_1, \cdots, a_{n-1}+b_{n-1}).$$

다음은 $a, b$ 의 곱 $a \cdot b$ 를 계산하자.

$\alpha^{n+1} = 1 = \alpha+\alpha^2+\cdots+\alpha^n$ 을 이용하면

$$\begin{aligned} a \cdot b = & \left(\sum_{i=1}^{n} a_{i-1}\,\alpha^i\right) \cdot \left(\sum_{j=1}^{n} b_{j-1}\,\alpha^j\right) \\ = & \quad (0\alpha + a_0b_0\alpha^2 + a_0b_1\alpha^3 + \\ & \cdots + a_0b_{n-2}\alpha^n + a_0b_{n-1}\alpha^{n+1}) \\ & + (a_1b_{n-1}\alpha + 0\alpha^2 + a_1b_0\alpha^3 + \\ & \cdots + a_1b_{n-3}\alpha^n + a_1b_{n-2}\alpha^{n+1}) \\ & + \cdots \\ & + (a_{n-1}b_1\alpha + a_{n-1}b_2\alpha^2 + \cdots + \\ & a_{n-1}b_{n-1}\alpha^{n-1} + 0\alpha^n + a_{n-1}b_0\alpha^{n+1}). \end{aligned} \quad (*)$$

그러므로

$$a \cdot b = c_0\alpha + c_1\alpha^2 + \cdots + c_{n-1}\alpha^n + c_n\alpha^{n+1}$$

라 하면

$$\begin{aligned} a \cdot b = & (c_0+c_n)\alpha + (c_1+c_n)\alpha^2 \\ & + \cdots + (c_{n-1}+c_n)\alpha^n \end{aligned} \quad (**)$$

이다. 위의 결론을 종합하면 다음과 같은 정리를 얻을 수 있다.

**정리 3.1.** 다항식 $f(x) = 1+x+\cdots+x^n$ 가 $GF(2)$위에서 기약 다항식이면 $GF(2^n)$의 원소 $a$, $b$ 의 곱 $a \cdot b$ 를 구하는데 $GF(2)$ 에서 $n^2$번의 곱과 $n^2-1$번의 덧셈이 필요하다.

이것을 알고리즘으로 표현하면 다음과 같다.

**알고리즘 3.2** (Multiplication algorithm)

Input : $a=(a_0, a_1, a_2, \cdots, a_{n-1})$,
$b=(b_0, b_1, b_2, \cdots, b_{n-1})$.

Output : $c=a \cdot b=(c_0, c_1, c_2, \cdots, c_{n-1})$

1. $b \leftarrow (0, b_0, b_1, \cdots, b_{n-1})$, $c \leftarrow a_0 \wedge b$ ;
2. for i = 1 to n-1
   2.1 $b \leftarrow shift(b)$;
   2.2 $c \leftarrow c \oplus (a_i \wedge b)$ ;
3 reduction(c) (** 과정 계산)
4 return(c)

여기서 $a_i \wedge b$ 는 스칼라 $a_i \in GF(2)$ 와 벡터 $b \in GF(2)^{n+1}$의 스칼라 곱 연산이다. 그리고 shift(b)는 벡터 b를 오른쪽으로 한번 순환 쉬프트하는 것이다. 여기서 shift 과정을 다음과 같이 쉬프트 처리하면 소프트웨어적으로 더 효과적이다. 즉, 곱은

$$
\begin{aligned}
a \cdot b &= \Big(\sum_{i=1}^{n} a_{i-1}\,\alpha^{i}\Big)\cdot\Big(\sum_{j=1}^{n} b_{j-1}\,\alpha^{j}\Big) \\
&= a_0\times(0\alpha+b_0\alpha^2+b_1\alpha^3+\cdots \\
&\quad +b_{n-2}\alpha^{n-1}+b_{n-1}\alpha^{n})+ \\
&\quad a_1\times(0\alpha+0\alpha^2+b_0\alpha^3+\cdots \\
&\quad +b_{n-2}\alpha^{n}+b_{n-1}\alpha^{n+1})+ \\
&\quad \cdots+a_{n-1}\times(0\alpha+0\alpha^2+0\alpha^3 \\
&\quad +\cdots+b_{n-2}\alpha^{2n-1}+b_{n-1}\alpha^{2n})
\end{aligned}
$$

(***)

이므로

$$a \cdot b = c_0\alpha+c_1\alpha^2+\cdots+c_{2n-2}\alpha^{2n-1}+c_{2n-1}\alpha^{2n}$$

라 하면

$$
\begin{aligned}
a \cdot b = {} & (c_0+c_n+c_{n+1})\alpha+ \\
& (c_1+c_{n+}c_{n+2})\alpha^2+ \\
& \cdots+(c_{n-2}+c_n+c_{2n-1})\alpha^{n-1} \\
& +(c_{n-1}+c_n)\alpha^n
\end{aligned}
\qquad (****)
$$

이다. 그리고 선계산(pre-computation) 과정을 도입하면 더 효과적이다.

**알고리즘 3.3** (Multiplication algorithm)

Input : $a=(a_0, a_1, a_2, \cdots, a_{n-1})$,
$b=(b_0, b_1, b_2, \cdots, b_{n-1})$

Output : $c=a\cdot b=(c_0, c_1, c_2, \cdots, c_{n-1})$

1. $c \leftarrow 0$
2. 선계산 :
   $p_1 \leftarrow a \gg 1$
   $p_2 \leftarrow a \gg 2$
   $p_3 \leftarrow p_1 \oplus p_2$
3. for i = 0 to $\left\lfloor \frac{n-1}{2} \right\rfloor$ (*** 과정)
   3.1 $s \leftarrow (b_{2i}b_{2i+1})_2$
   3.1 if $s \neq 0$ then
   $c \leftarrow c \oplus (p_s \gg 2i)$
4. reduction(c) (**** 과정 계산)
5. return(c)

위에서 $a \gg i$는 $a$를 오른쪽으로 $i$만큼 쉬프트 하는 것을 의미한다. 이 알고리즘은 쉬프트 연산과 XOR연산의 수를 줄이기 위해 a의 쉬프트 연산을 미리 계산을 해놓고 각 b의 비트의 값에 의해 계산하는 방법을 택하였다. 2단계는 선 계산을 하는 부분으로 $p_1$은 $(b_{i+1}b_i)_2$의 값이 1인 경우 $p_2$는 2인 경우 $p_3$는 3인 경우의 값을 계산한 것이다. 3단계는 선계산한 것을 직접 적용하는 부분으로 $b$를 2진 표현으로 봤을 때 2bit씩 잘라서 그 값에 따라 $p_1, p_2, p_3$를 적용한다. 예를 들어 $11_2$는 3를 의미하므로 $p_3$를 적용한다. 이 방법은 선계산을 하지 않는 경우에 비해 계산양을 반으로 줄일 수 있는 장점이 있다.

다음은 $GF(2^n)$의 원소 $a$에 대한 제곱을 계산하여 보자. 만약 $f(x) = 1+x+\cdots+x^n$ 가 $GF(2)[x]$에서 기약다항식이면 $n$은 짝수이다. 따라서 $n = 2m$라 하자. 그러면

$$
\begin{aligned}
a^2 &= \Big(\sum_{i=1}^{2m} a_{i-1}\,\alpha^{i}\Big)^2 \\
&= \sum_{i=1}^{m} a_{i-1}\alpha^{2i} + \sum_{i=m+1}^{2m} a_{i-1}\alpha^{2i} \\
&= \sum_{i=1}^{m} a_{i-1}\alpha^{2i} + \sum_{i=m+1}^{2m} a_{i-1}\alpha^{2i-2m-1} \\
&= \sum_{i=1}^{m} a_{i-1}\alpha^{2i} + \sum_{j=1}^{m} a_{m+j-1}\alpha^{2j-1}.
\end{aligned}
$$

위의 결과에 의하여 다음의 결론을 얻을 수 있다.

**정리 3.4** $f(x) = 1+x+\cdots+x^n$ 가 $GF(2)[x]$에서 기약 다항식이라 하자. 그러면 $a=(a_1, a_2, \cdots, a_n) \in GF(2^n)$의 제곱은 좌표의 위치를 바꾸면 구할 수 있다. 즉 n=2m이면
$a^2=(a_m, a_0, a_{m+1}, a_1, \cdots, a_{2m-1}, a_{m-1})$
이다.

**예제 3.5** $f(x) = 1+x+x^2+x^3+x^4$를 $GF(2)$위에서 기약다항식이라 하고 $1+\alpha+\alpha^2+\alpha^3+\alpha^4 = 0$ 라 하자. 그러면 $B = \{\alpha, \alpha^2, \alpha^3, \alpha^4\}$ 는 $GF(2)$위에서

$GF(2^4)$에 대한 변형된 다항식 기저이다. $a = a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4$를 $GF(2^4)$의 원소라 하면

$$a^2 = (a_0, a_1, a_2, a_3)^2 = (a_2, a_0, a_3, a_1).$$

마지막으로 Schroeppel[12]이 제안한 Almost Inverse Algorithm (AIA )은 다항식 기저에서 $a \cdot b = a^k$ 를 만족하는 b, k를 먼저 구하여 a의 역원을 구하는 알고리즘으로 변형된 다항식 기저의 경우에 적용할 수 있으며 알고리즘은 다음과 같다.

**알고리즘 3.6** ( Inversion using AIA)

Input : $a=(a_0, a_1, a_2, \cdots, a_{n-1})$,

Output : $a^{-1}=(b_0, b_1, b_2, \cdots, b_{n-1})$

1. AIA 알고리즘을 이용하여 $a \cdot b = a^k$를 만족하는

   $a^k$, $b=(b_0, b_1, b_2, \cdots, b_{n-1})$을 계산한다.

2. $k \leftarrow k-(n+1)$

3. if ( $k \neq 0$) then

   3.1 $d_0=(b_0, \cdots, b_{k-1})$,

   $d_1=(b_k, \cdots, b_{n-1}, 0, \cdots, 0)$

   3.2 $b \leftarrow (d_0 \gg n+1-k) \oplus d_1$

   3.3 $a^{-1} \leftarrow$ reduction(b)

   else $a^{-1} \leftarrow b$

위에서 $\overline{b}$는 $b$의 complement를 나타낸다. 그리고 1번 단계는 $a \cdot b = a^k$ 를 구하는 단계이고 구하여 이 경우 $k$의 값이 $n+\frac{n}{2}$ 에서 $2n$ 사이의 값을 갖기 때문에[12] 2단계에서 $a^{n+1}=1$을 이용하여 $k$ ( $0 \leq k < n$)의 값을 줄인다. 3단계에서 3.1, 3.2는 $a^{n+1-k}$을 b에 곱하여 $b=b_0\alpha + \cdots + b_{n-1}\alpha^n + b_n\alpha^{n+1}$를 구하는 과정이고 3.3은 (**)의 reduction 과정이다.

## Ⅳ. 구현한 결과

| | 변형된다항식 기저 | 기존의 결과[13] | |
|---|---|---|---|
| 유한체 | $GF(2^{180})$ | $GF(2^{176})$ (Composite field) | $GF(2^{177})$ (Trinomial) |
| 컴퓨터 기종 | Pentium 166 | Pentium 133 | Pentium 133 |
| 곱셈 | 32.6 | 62.7 | 71.8 |
| 제곱 | 1.1 | 5.9 | 2.7 |
| 역원 | 181.4 | 160 | 225 |

위 표는 Pentium 166에서 GNU C 를 이용하여 변형된 다항식 기저로 구현한 결과와 기존의 결과를 비교한 것이다. 변형된 다항식 기저의 곱은 알고리즘3.3을 이용한 것이고 $GF(2^{180})$은 AOP $x^{180}+x^{179}+\cdots+x+1$로 구성한 유한체이다. 또한 타원곡선의 한번의 덧셈 연산은 유한체의 두 번의 곱셈과 한번의 제곱, 한번의 역원 연산으로 이루어지므로 타원곡선 연산의 효율성을 추측할 수 있다.

## Ⅴ. 결론

타원곡선 암호시스템을 하드웨어 또는 소프트웨어로 구현할 수 있다. 이 경우 최적 정규기저는 하드웨어 구현에 Trinomial을 이용한 다항식 기저는 소프트웨어 구현에 용이하다. 이 두 암호 시스템에서 교신을 할 경우 기저변환이 필요하다. 기저변환의 경우 기저변환 행렬을 저장해야 하는 문제는 제한된 조건하에서(스마트카드, 핸드폰 등) 타원곡선 암호법 구현의 제약 조건이 된다. AOP는 최적 정규기저를 형성하는 다항식으로 이 기저를 사용하면 하드웨어 구현이 효과적이다. 또한 이 논문에서 AOP

를 이용한 변형된 다항식 기저를 사용하면 소프트웨어적으로 효과적일 뿐 아니라 최적 정규기저와의 기저 변환을 위치변환으로 가능한 것을 보였다. 그러므로 AOP를 사용하여 최적정규기저로 하드웨어 구현을 하고 변형된 다항식 기저로 소프트웨어 구현을 할 경우 제한된 조건하에서도 효과적인 구현이 될 것이다.

## 참고 문헌


[1] D.V. Baily and C. Paar, "Optimal extension field for fast arithmetic in public key algorithm, Crypto'98, Springer-Verlag, pp. 472 -485, 1998.

[2] J. Guajardo and C. Paar, "Efficient algorithms for elliptic curve cryptosystems'', Crypto'97, Springer-Verlag, pp. 342-356, 1997.

[3] G. Harper, A. Menezes and S.A. Vanstone, "Public-Key Cryptosystems with very small Key length'', Eurocrypt'92, Springer-Verlag, pp. 163-173, 1992.

[4] M.A. Hasan, M.Z. Wang and V.K. Bhargava, "A modified Massey-Omura parallel multiplier for a class of finite fields'', IEEE Transactions on Computers, 42(10), pp.1278-1280, October 1993.

[5] T. Itoh, O. Teechai and S. Tsuji, "A fast algorithm for computing multiplicative inverses in $GF(2^l)$ using normal bases"(in Japanese), *J. Society for Electronic Communications (Japan)*, 44 pp. 31-36, 1986.

[6] B.S. Kaliski Jr and Y.L. Yin, "Storage-Efficient Finite Field Basis Conversion'', SAC'98, 1998.

[7] N. Koblitz, "Elliptic curve crypto -systems'', *Mathematics of Computation*, 48, pp. 203-209, 1987.

[8] C.K. Koc and B. Sunar, "Low complexity bit-parallel canonical and normal basis multiplier for a class of finite fields", IEEE Transactions on Computers, 47(3), pp. 353-356, March, 1998.

[9] R. Lidl and H., Niederreiter, Introduction to finite fields and their applications, Revised edition, Cambridge University press, Cambridge, 1994.

[10] A.J. Menezes, Applications of finite fields, Kluwer academic publishers, Boston, 1993.

[11] V.S. Miller, "Use of elliptic curve in crypto -graphy'', Crypto'85, Springer-Verlag, pp.417-426, 1986.

[12] R. Schroeppel, H. Orman, S. O'Malley and O. Spatscheck, "Fast key exchange with elliptic curve systems'', Crypto 95, Springer-Verlag, pp. 43-56, 1995.

[13] E. De Win, A. Bosselaers, S. Vandenberghe, P. De Gersem, and J. Vandewalle. "A fast sofrware implemen -tation for arithmetic operations in $GF(2^l)$", Asiacrypt 96, Springer-Verlag, pp65-76, 1996.

[14] IEEE P1363, Standard specification for public key cryptography, Annex A, 1999.

著者紹介 ----------

**김 창 한** : 85년 고려대학교 수학과 졸업

92년 고려대학교 대학원졸업(이학박사)
92년 - 현재 세명대학교 조교수
관심분야 : 암호론, 응용대수학, 정보이론

**이 성 재** : 97년 고려대학교 수학과 졸업
99년 고려대학교 대학원졸업(이학석사)
99년 - 정보보호센터
관심분야 : 암호론, 응용대수학

**박 성 준(Sung Joon Park) 정회원**

1983년 2월 :한양대학교 수학과 졸업
1996년 2월 : 성균관대학교 박사
1985년 1월~1994년 3월 : 한국전자통신연구원
1996년 4월~현재 : 한국정보보호센터 기반기술팀장
〈관심분야〉 암호학, 정보이론