타원곡선 암호법에 관한 연구

임 종 인*, 서 광 석**, 박 상 준***
* 고려대학교 수학과
** 서남대학교 수학과(o)
*** 한국전자통신연구소 선임연구원

On the Elliptic curve cryptosystem

Jong In Lim*, Kwang Suk Suh**, Sang Jun Park***
* Korea University Mathematics
** SeoNam University Mathematics
*** Electronics and Telecommunictions Research Institute Senior Member of Technical Staff

요 약 문

본 고는 최근 키 사이즈가 적으면서도 안전하다고 알려져 있는 타원곡선 암호법에 대해서 고찰한 바, 특별히 수정된 다항식 기저를 이용하여 타원곡선의 연산을 용이하게 하는 방법을 제안한다. 한편 랜덤한 타원곡선은 공개키 암호법에 사용하기 부적당하며, 따라서 타원곡선군의 위수를 구할 필요가 있는데 이는 Schoof 알고리즘으로 구할 수 있으나 많은 시간이 소요되는 바 본 고에서는 Weil 정리를 사용하여 위수를 손쉽게 구할 수 있는 방법을 제안하며, 컴퓨터 실험 결과를 소개한다.

I. 서 론

정보화 사회로 진입하고 있는 오늘날 유무선 통신상에서 보안 관련 문제는 효율성 문제와 함께 가장 시급하게 해결해야 하며, 한편으로는 급속히 발전하고 있는 분야이기도 하다. 1976년 Diffie-Hellman [2]의 공개키 암호법(Public key cryptosystem)제안으로 전기를 맞이한 암호학 분야는 이후 눈부신 발전을 하여왔다.[12]

현재 가장 널리 사용되는 공개키 암호법은 1978년 발표된 RSA 암호법[12]과 1985년 발표된 ElGamal 암호법[3]이다. 1991년 NIST(미 표준기술국)에서 발표한 DSS(Digital Signature Standard)는 ElGamal 암호법의 변형으로 H/W 구현의 편리함과 특허 문제로 인하여 RSA는 ElGamal 암호법이 선택된 것으로 알려져 있다.[12] ElGamal 암호법의 안전도는 유한군의 이산로그 문제가 어렵다는 가정에 근거하고 있다.

한편, 유한군 중에서도 구현상의 편리함 때문에 표수가 2인 유한체 F_{2^m}의 곱셈군 $F_{2^m}^*$가 주로 사용되어 왔으나 index calculus법[1]이라는 강력한 공격법이 1984 Coppersmith에 의하여 발견되었고 최근의 급속한 컴퓨터 발전으로 현재 안전도를 보장하기 위해서는 $m \geq 800$ 이 되어야 한다.[12] 그런데 공개키 암호법을 널리 사용하기 위해서 최근에는 스마트 카드에 장착할 수 있는 방법이 많이 연구되고 있으나, m이 클 때 ElGamal 암호법을 스마트 가드의 내부 processor에 설계하는 것은 매우 어려운 일이다.

1985년 Koblitz에 의하여 제안된 타원곡선 암호법[6]이 제안되었는데 타원곡선 암호법은 $F_{2^m}^*$ 대신에 F_{2^m} 위에서의 타원곡선군을 사용하는 암호법으로서 유한체를 사용하는 경우에 비해서 키 사이즈가 1/4 이하로 줄어들어도 같은 안전도가 보장되며, 수시로 타원곡선을 교환함으로서 안전도를 크게 할 수 있다는 장점이 있다. 1993년 Vanstone등[9]은 $F_{2^{155}}$ 위에서의 타원곡선 암호법을 실용화하는데 성공하였다. 이들은 11000개의 gate를 사용하여 H/W로 구현함으로서 스마트 카드에 장착가능성을 보였지만 이들의 방법을 S/W 구현할 경우에는 속도 및 효율성 면에서 문제점을 지니고 있다.

본 논문에서는 타원곡선 암호법에 대한 소개와 타원곡선의 선택 방법의 제안 및 S/W 구현시 속도 개선에 관한 연구 결과를 소개하고자 한다.

2. 타원곡선에 관한 일반이론

2.1 타원곡선의 정의

유한체 K가 주어졌을 때 다음의 식을 고려하자.

$$E: y^2+a_1 xy +a_3 y=x^3+a_2 x^2+a_4 x+a_6,\ a_i \in K\ (\Delta \neq 0)\ (\Delta\text{는 판별식})$$

위 식을 만족하는 $K\times K$ 상의 점들 (x, y)와 무한원점으로 이루어진 집합 $E(K)$를 타원곡선이라 한다. $K=F_{2^m}$의 경우에 타원곡선은 다음의 두 가지 경우로 나누어진다.

o Non-supersingular : $y^2 + xy = x^3 + a_2 x^2 + a_6 \quad (\Delta = a_6 \neq 0)$

o Supersingular : $y^2 + a_3 y = x^3 + a_4 x + a_6 \quad (\Delta = a_3^4 \neq 0)$

이때 $E(K)$상의 점 P, Q가 주어졌을 때 2.2 에서와 같이 연산을 정의하면 $E' = E(K)$는 무한원점 0을 단위원으로 하는 군이 된다. 대부분의 타원곡선군은 Hasse의 정리[5]에 의하여 cyclic군이 된다는 것을 보일 수 있다.

2.2 타원곡선상의 연산

타원곡선 상의 두점 $P = (x_1, y_1)$, $Q = (x_2, y_2) \neq -P$에 대해서 다음과 같이 덧셈을 정의할 수 있으며, 이 덧셈에 대해서 타원곡선은 군이 된다.

o Non-supersingular : $y^2 + xy = x^3 + a_2 x^2 + a_6 \quad (\Delta = a_6 \neq 0)$

$-P = (x_1,\ y_1 + x_1)$

$P + Q = (x_3,\ y_3)$라 하면

$$x_3 = \begin{cases} (\frac{y_1+y_2}{x_1+x_2})^2 + (\frac{y_1+y_2}{x_1+x_2}) + x_1 + x_2 + a_2, & P \neq Q \\ x_1^2 + \frac{a_6}{x_1^2}, & P = Q \end{cases}$$

$$y_3 = \begin{cases} (\frac{y_1+y_2}{x_1+x_2})(x_1+x_3) + (x_3+y_1), & P \neq Q \\ x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3, & P = Q \end{cases}$$

o Supersingular : $y^2 + a_3 y = x^3 + a_4 x + a_6 \quad (\Delta = a_3^4 \neq 0)$

$-P = (x_1,\ y_1 + a_3)$

$P+Q=(x_3,\ y_3)$라 할 때

$$x_3 = \begin{cases} (\frac{y_1+y_2}{x_1+x_2})^2 + x_1 + x_2, & P \neq Q \\ \frac{x_1^4 + a_4^2}{a_3^2}, & P = Q \end{cases}$$

$$y_3 = \begin{cases} (\frac{y_1+y_2}{x_1+x_2})(x_1+x_3) + y_1 + a_3, & P \neq Q \\ (\frac{x_1^2+a_4}{a_3})(x_1+x_3)+y_1+a_3, & P = Q \\ \qquad = (\frac{x_1^2+a_4}{a_3})(x_1+x_3) + y_1 + a_3, & P = Q \end{cases}$$

2.3 타원곡선의 형태 분석 [8]

F_2- 위에서의 타원곡선을 supersingular와 non-supersingular의 경우로 나누어 분석하자. Supersingular의 경우 m이 홀수이면 3가지 경우로 구분되며 m이 짝수이면 7가지 경우로 나눌 수 있다. 위의 10가지 경우 각각에 대한 분석은 완전히 이루어져 있으며 타원곡선군의 *order N* 도 구해져 있다. 임의의 supersingular 타원곡선이 주어졌을 때 위의 10가지 경우 중 어느 경우에 속하는지 판정할 수 있는 polynomial time 알고리즘이 있다.

Non-supersingular의 경우에는 $2(q-1)$개의 경우가 있으며 흔히 쓰는 형태는 $y^2+xy=x^3+a_6$ 이다. $(q=2^m,\ a_6\neq 0)$ 이 경우 order N을 구하는 것은 Schoof 알고리즘에 의하여 구할 수 있으며, 실행 시간은 $O(\log_q 8)$이다. 1993년 Vanstone등은 m = 155의 경우 Schoof 알고리즘을 성공적으로 실행하였으며 실행시간은 Sun workstation하에서 60여 시간이었다.[9]

3. 타원곡선 ELGamal 암호법 소개 및 새로운 제안

3.1 이산로그 문제

Order N 의 cyclic 유한군 $G=\langle g\rangle$가 주어졌다고 하자. G의 임의의 원소 y는 $y=g^x$ ($0\le x\le N-1$)의 형태로 표시되며, 이 때 x를 원시원 g를 base로하는 y의 이산로그(discrete logarithm of y with respect to primitive element g)라하며 $x=\log_g y$ 로 나타낸다. G와 원시원 g가 공개되어 있어도 주어진 원소의 이산로그를 구하는 것은 대단히 어려운(intractable)문제로 알려져 있다. 그러나 $G=F_{2^n}^*$의 경우에는 index calculus 법[1]이라는 polynomial time 알고리즘이 있다.

$F_{2^n}^*$ 의 경우와 달리 F_{2^n} 상의 타원곡선을 이용할 경우 이산로그 공격법은 현재로서는 square root 법과 Pohlig-Hellman법 밖에 없다.[12] 이들 공격법은 N 이 10^{40}이상이 되고 큰 소인수를 가지면 적용할 수 없다. $m\ge 130$로 하면 N은 Hasse의 정리에 위하여 10^{40} 이상이 되고 대개의 경우 큰 소인수를 가지게 된다.

(주의) supersingular의 경우 최근 F_{2^n} 위에서의 터원곡선에 대한 이산로그 문제가 $F_{2^{kn}}$ 에서의 이산로그 문제로 귀착될 수 있다는 것이 보여졌다.[8]

따라서 supersingular 타원곡선을 사용할 때는 $m\ge 200$이 되어야 할 것이다. 예를 들어 $y^2+y=x^3+x+1$, $m=239$로 하면 $N=P_{72}$가 되고 이산로그 문제는 7.2×10^{20}의 실행시간이 걸린다.

Non-supersingular의 경우에는 위와 같은 공격법이 없으므로 $m\ge 130$ 으로 하여 적당한 $y^2+xy=x^3+a_6$를 택하면 된다. m을 크게 하면 안전도는 증가하지만 N을 구하는 Schoof 알고리즘[11] 실행 시간이 커진다.

3.2 타원곡선 ElGamal 암호법

타원곡선 ElGamal 암호법은 다음과 같다.

(i) $K=F_{2^n}$ 위에서의 적당한 타원곡선 E와 E = <P> 가 되는 P를 구한다.

(ii) 각 user는 비밀키 l과 공개키 $l\cdot P$를 구한다.

(iii) 메시지 $M=(M_1, M_2)\in K\times K$를 전송하기 위하여 송신자 A 는 난수 k를 택하고 $k\cdot P$를 계산한다.

(iv) 수신자 B의 공개키 $d \cdot P$로 부터 $k(b \cdot P) = (\bar{x}, \bar{y})$를 계산한다. 이때, $\bar{x} \neq 0$, $\bar{y} \neq 0$라 가정하자. 0이 되는 경우의 확률은 매우 작으므로 이렇게 가정하여도 무방하다. 그리고, 유한체 상의 곱 $M_1 \times \bar{x}$, $M_2 \times \bar{y}$를 계산한다.

(v) A는 B에게 $k \cdot P$와 $M_1 \times \bar{x}$, $M_2 \times \bar{y}$ 를 전송한다.

(vi) B는 $b(k \cdot P) = (\bar{x}, \bar{y})$를 계산하고 이것을 이용하여 $M = (M_1, M_2)$ 를 구한다.

ElGamal암호법은 많은 장점을 가지고 있지만, message expansion이라는 약점을 지니고 있으며 이를 극복하기 위한 여러가지 방법이 제안되어 있다.[12]

3.3 구현 효율성 제고를 위한 새로운 제안

타원곡선 암호법을 구현하기 위해서는 타원곡선의 위수 N를 계산하여 N의 약수 중 큰 소인수가 있는지를 확인해야 한다. 그러므로 타원곡선의 위수를 계산할 필요성이 있는데 non-supersigular의 경우에는 supersingular의 경우와는 달리 정확한 위수를 구할 수 있는 빠른 알고리즘이 존재하지 않는다. 사실 타원곡선의 위수를 구하는 polynomial time 알고리즘인 Schoof 알고리즘이 있으나 계산복잡도가 $O(\log_q 8)$인 알고리즘으로서[11] m이 커지면 실행 시간이 엄청나 수행 불가능한 알고리즘이다. 1993년 Vanstone 등[9]은 Schoof 알고리즘에 baby step - giant step method를 도입하여 수행 시간을 대폭 줄였으며, m = 155인 경우 구현에 성공하였으며 실행시간은 Sun Workstation하에서 60여 시간이었다. 실제 PC 환경하에서 m = 105의 경우를 적용한 결과 100여 시간이 소요되었다. 이러한 결과는 타원곡선 암호법을 구축하는데 매우 많은 시간이 소요되어 타원곡선군의 변경이 어렵게 된다.

본 논문에서는 Weil의 정리[5]를 이용하여 타원곡선군의 위수를 쉽게 계산할 수 있는 방법을 제안하고자 한다.

Weil 정리. 유한체 F_q 위에서의 타원곡선 $E = E(F_q)$의 위수를 $N = q+1-t$ 라 하자. 그러면 E를 F_{q^k} 위에서의 타원곡선으로 고려한 경우 타원곡선군 $E_k = E(F_{q^k})$의 경우의 위수 N_k는 다음과 같다.

$N_k = q^k + 1 - \alpha^k - \beta^k$ (단, α, β는 $1-tX+qX^2$의 근임)

Weil 정리[5]를 이용하면 F_q 위에서의 타원곡선군의 위수를 계산하면 아주 쉽게 F_{q^k} 위에서의 타원곡선군의 위수를 계산할 수 있으므로 Schoof 알고리즘을 적용하지 않아도 안전한 타원곡선 암호법을 구축할 수 있다. 예를 들면 $155 = 5 \times 31$이므로 먼저 $\#E(F_{2^5})$을 구한다. 이것은 Schoof 알고리즘을 적용하지 않고도 시행착오법으로 구할 수 있다. $\#E(F_{2^5})$은 모두 12가지 경우가 나타나며 각각에 대하여 Weil정리를 이용하여 $\#E(F_{2^{155}})$를 구한다.

실험결과 $\#E(F_{2^5}) = 36$ 또는 42의 경우가 $\#E(F_{2^{155}})$ 이 10^{40} 이상이 되고 큰 소인수를 가져야 하는 조건을 만족하는 좋은 타원곡선임을 확인할 수 있었다. 그런데 Weil 정리를 이용하여 Schoof 알 고리즘 적용과정을 생략하면 가용타원곡선이 제한되는 불리함이 있지만 타원곡선은 이 경우에도 충분히 있고 타원곡선 암호법 구축 시간이 훨씬 빨라지는 장점이 있다. 실제로 M = 155에 대한 위의 결과는 PC 환경하에서 10여시간밖에 소요되지 않았다. 이와 같은 제안은 몇몇학자들에 의하여[9] 이미 제안되었지만 우리는 실험을 통하여 효율성을 확인하였고 다음의 제안을 통하여 특히 S/W 구현시 효율성을 제고하려 한다.

두번째 제안은 수정된 다항식 기저(modified ploynomial basis)[13]를 사용하는 것이다.

보통 F_{2^m}의 기저로는 최적정규기저를 사용한다.[10] 타원곡선 암호법의 구현시 F_{2^m} 위에서의 많은 연산이 이루어지므로 최적정규기저를 사용하는 것은 다항식 기저를 사용하는 것에 비하여 연산속도를 빠르게 함으로써 구현시간을 단축한다. 특히 최적정규기저는 H/W로 구현하는데 적합하다. 그러나 유한체 위의 곱셈 연산의 S/W 구현시에는 수정된 다항식 기저가 최적 정규기저보다 더욱 효율적인 것으로 평가되고 있다.[4] 이 방법을 첫번째 제안과 결합하면 다음과 같이 된다.

$m = st$라 하자.($s \ll t$) F_{2^s}위에서의 타원곡선 E를 택하여 $\#E(F_{2^s})$를 구한다. $[F_{2^m} : F_{2^s}] = t$ 이므로 F_{2^m}을 생성하려면 F_{2^s}에서 기약인 t차 다항식을 찾아야 한다.

정리 1. $m = st$이고, $\gcd(s, t) = 1$이라 하자. 그러면 F_2 위에서 기약인 t차 다항식은 F_{2^s}위에서도 기약이다.

정리 1.을 이용하면 F_{2^s}에서 t차 기약다항식 $f(x)$을 쉽게 찾을 수 있다. $f(x)$을 이용하여 F_{2^m}이 표현되었을 때 수정된 다항식 기저로 표현되였다고 정의하자. 수정된 다항식으로 표현된 유한체 F_{2^m}의 원소는 t 보다 작은 F_{2^s} 위의 다항식일 것이고 두 원소의 곱셈은 모듈라 곱일 것이다. 그리고 한 원소의 역원 계산은 유클리드 알고리즘을 사용하면 매우 쉽게 계산된다. 수정된 다항식 기저를 사용하였을 때의 두 원소의 곱셈은 F_{2^s} 위의 곱셈 연산이 필요한데 이 연산은 $F^*_{2^s} = \langle g \rangle$를 표시하여 다음과 같이 계산하면 된다.

$$g^i \times g^j = g^{i+j \bmod 2^m-1}$$

한편 계수가 0 또는 1인 기약다항식으로 유한체 F_{2^m}를 표시하였으므로 모듈라 곱셈도 효율적으로 수행된다. 그러므로 수정된 다항식 기저를 사용하면 최적정규기저보다 S/W 구현시 연산속도면에서 효율적이다. 현재까지는 수정된 다항식 기저를 사용할 수 있으려면 $\gcd(s, t) = 1$일 조건이 필요하다.

정의 1. 유한체 F_{2^t} 위의 다항식 $f(x) = a_0+a_1x+\cdots+a_tx^t$ $(a_i \in F_{2^t})$ 중 상수항을 제외한 계수가 0 또는 1인 경우, 즉 $1 \le i \le t$, $a_i = 0$ 또는 1인 다항식을 단순 다항식이라 하자.

Conjecture. 유한체 F_{2^t} 위의 $t > 2$이면 t차인 단순 기약다항식(원시다항식)이 적어도 하나 존재한다.

실제 컴퓨터 실험한 결과 F_{2^8}인 경우에는 3차에서 32차 까지의 단순 원시다항식이 존재한다. 단순 다항식은 유한체 F_{2^t}의 표현 방법에 무관하다. 즉, F_{2^t}의 기저를 바꾸어도 단순 다항식은 단순 다항식이 된다. 단순 기약다항식을 사용하면 $\gcd(s,t) \neq 1$인 경우에도 $\gcd(s,t) = 1$인 경우와 마찬가지 빠른 속도로 유한체 F_{2^m} 위의 연산을 수행할 수 있다.

따라서 위의 2가지 제안을 함께 사용하면 S/W 구현시 연산 속도가 빨라지고 Schoof 알고리즘 과정이 생략되기 때문에 타원곡선 암호법의 실행 시간은 대폭 단축될 것이다.

3.4 실험 결과

$a_6(\neq 0)$과 trace가 1인 γ들이 $K=F_q$의 원소일 때 타원곡선 $y^2+xy = x^3+a_6$의 위수와 $y^2+xy = x^3+\gamma x^2+a_6$의 위수 합은 2q+2이다. 그러므로 $y^2+xy = x^3+a_6$에 대한 위수만 구하면 $y^2+xy = x^3+\gamma x^2+a_6$의 위수는 자동적으로 구해진다. 이때 $y^2+xy = x^3+\gamma x^2+a_6$를 $y^2+xy = x^3+a_6$의 대응곡선이라 한다.

$g(z) = 1+z^2+z^5$은 F_2 위의 원시다항식이다. $K = F_2[z]/(f(z))$라하면 $y^2+xy = x^3+a_6$ 는 $a_6 = z^i$ 에 대한 위수는 다음과 같다.

i	위수	대응곡선의 위수
0	44	22
1,2,4,8,16	32	34
3,6,11,12,13, 17,21,22,24,26	36	30
5,9,10,18,20	28	38
7,14,19,25,28	24	42
15,23,27,29,30	40	26

Weil 정리를 이용하여 위에서 구한 F_{2^5} 위에서의 타원곡선을 $F_{2^{155}}$ 위의 타원곡선 고려하였을 때의 위수를 구하면 다음과 같다.

위수 = 44인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865369586664589280504350596

위수 = 32인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865316669375756005815532512

위수 = 36인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193864769109245788687953062028 =

$2^2 \times 3^2 \times 126866461573863100538513247525682746355425 1723$

위수 = 28인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865577831603432792572676884 =

$2^2 \times 7 \times 4217 \times 1108313 \times 34899987243800759898721397577177 1043$

위수 = 24인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865049333247828076001209256

위수 = 40인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193864737975717109037727376280

위수 = 22인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193864932458103099447991433342 =

$2 \times 11 \times 26041 \times 41231 \times 193350420720153567979425606968917329 1$

위수 = 34인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193864985375391932722680251426

위수 = 30인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865532935521900040542721910

위수 = 38인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193864724213164255935923107054 =

$2 \times 19 \times 4093 \times 475169 \times 617982087590864164165926670909680049$

위수 = 42인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865252711519860652494574682

위수 = 26인 타원곡선의 확대체 $F_{2^{155}}$ 위에서의 위수 :

45671926166590716193865564069050579690768407658

4. 결 론

본 고에서는 타원곡선 암호법의 효율적인 구축을 위하여 Schoof 알고리즘 대신에 Weil 정리를 사용할 것을 제안하였으며, 타원곡선 위의 연산은 유한체 위의 연산을 기본으로 하여 수행되는 바 유한체 위의 연산을 효과적으로 수행할 수 있는 수정된 다항식 기저를 이용할 것을 제안하였다.

수정된 다항식 기저는 이미 타 논문에서 발표된 바 있으나, 본 논문에서는 이들 논문의 한계인 $\gcd(s,t) = 1$를 극복할 수 있는 단순 기약다항식을 사용할 것을 제안하였다. 컴퓨터 실험결과 3차 이상에서는 단순 기약다항식이 항상 존재함을 확인하였으며, 존재성에 대한 증명은 지최적정규기저는 임의의 확대체에 존재하는 것이 아니고 특별한 경우에만 존재하므로 단순 기약다항식을 이용한 타원곡선 암호법의 S/W 구현은 매우 효용성이 크다고 할 것이다. 추후 수정된 다항식기저를 사용하여 타원곡선 암호법을 실제 구현하여 이에 대한 효율성을 입증할 예정이다.

한편 본 연구에서는 F_{2^5} 위의 타원곡선에 대한 위수를 이용하여 확대체의 위수를 구하였으며, 대부분의 경우 타원곡선군을 안전하게 사용할 수 있는 위수가 큰 소인수를 갖는 타원곡선을 발견할 수 있었다.

참 고 문 헌

1. D. Coppersmith, Fast evaluation of logarithms in finite fields of characteristic two, *IEEE Transactions on Information Theory*, 30(1984), 587-594.

2. W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, 22(1976), 644-654.

3. T. ElGamal, A subexponential-time algorithm for computing discrete lograithms over $GF(p^2)$, *IEEE Transactions on Information Theory*, 31(1985), 473-481.

4. G. Harper, A. Menezes, S. Vanstone, Public key cryposystem with very small key lengths, *Advance in Cryptology EUROCRYPT'92*, Lecture Note in Computer Science, 658(1993), Springer-Verlag, 163-173.

5. N. Koblitz, *Introdution to Elliptic Curves and Modular Forms*, Springer-Verlag, 1987.

6. N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, 48(1987), 203-209.

7. N. Koblitz, Constructing elliptic curve cryptosystems in characteristic 2, *Advance in Cryptology EUROCRYPT'90*, Lecture Note in Computer Science, 537(1991), Springer-Verlag, 156-167.

8. A. Menezes, T.Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory*.

9. A. Menezes and S. Vanstone, Elliptic curve cryptosystems and their implementation, *Journal of Cryptology*.

10. R. Mullin, I. Onyszchuk, S. Vanstone and R. Wilson, Optimal normal bases in $GF(p^n)$, *Discrete Applied Mathematics*, 22(1988), 149-161.

11. R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, *Mathematics of Computation*. 44(1985), 483-494.

12. G. Simmons(editor), *Contemporary Cryptology, The Science of Information Integrity*, IEEE Press, New York, 1991.

13. Y. J. Choi and H. Hwoang, On the cryptosystem using elliptic curve, Jw-Isc'93 Proceedings, Seoul, Korea,1993.