

ECC 키분배에서 공유키 존재에 관한 연구

A Study on a Shared Key Existence of ECC Based Key Distribution System

이 준* 박종범*
Jun Lee Jongbum Park

Abstract

As a result from Hasse's theorem it is not always possible to share a common key between any two ECC public keys. Even though ECC algorithm is more efficient than any other Encryption's with respect to the encryption strength per bit, ECC ElGamal algorithm can not be used to distribute a common key to ECC PKI owners. Approaching mathematical ways in a practical situation, we suggest possible conditions to share a common key with ECC PKI's. Using computer experiments, we also show that these suggestions are right. In the conditions, we can distribute a common key to proper peoples with ECC ElGamal algorithm.

Keywords : Key Distribution, ECC ElGamal

1. 서론

키 공유의 개념은 KDC를 통해서 3인 이상의 다수에게 공통된 비밀키를 분배하는 것이다. 비밀키를 분배하는 대표적인 공개키로는 RSA와 ECC가 있다. 비슷한 수준의 안전도에서 RSA 1024비트에 대해 ECC는 160비트로 비교된다^[1]. 단위 비트 당 암호화 강도는 ECC가 훨씬 높아 시간당 암호화 및 복호화에 소요되는 계산량이 적고 속도가 빠르다. 하지만 RSA는 수의 범위에서 모든 비밀키를 메시지로 안전하게 분배할 수 있으나 ECC는 소수체 위에서 정의된 타원방정식 위의 임의의 x 에 대해서 y 가 항상 존재함을 보

장할 수 없으므로 한 쌍의 좌표로 구성된 메시지를 이용하여 참여자 모두에게 공통의 비밀키를 전달할 수 없다.

본 연구는 ECC에서 실용적 공유키 전달 가능 조건을 검토한다. 현재 전산기 계산 수준에서 대칭키 블록 암호는 112비트 이상이면 안전함을 주장한다^[2]. 160비트 ECC의 공개키가 블록암호에서 필요한 비밀키 길이 112 비트를 전달하는 경우와 같이 공유키보다 큰 ECC 공개키 체계에서 메시지의 x 좌표로 공유키를 전달하는 특성을 수학적으로 접근하여 ECC에서 공유키 존재를 검토한다. 그 결과를 컴퓨터로 실험하여 이론과 실재를 비교하고, ECC에서 실용적 공유키가 존재할 수 있음을 보인다. 또한 업무용으로 보급된 컴퓨터를 공유키 분배 서버로 사용하여 키 분배에 필요한 시간을 측정함으로써 ECC ElGamal 알고리즘^[3]을 이용한 공유키 분배가 실용 가능함을 보인다. 이는 국방망에

† 2009년 4월 20일 접수~2009년 6월 26일 게재승인
* 공군사관학교 전자전산학과(Korea Air Force Academy)
책임저자 : 이 준(jlee@afa.ac.kr)

서 기밀성을 유지하면서 효율적으로 자료를 공유하는 NCW의 비밀키 분배에 관한 핵심기술이 될 것이다.

2. 관련연구

가. Hasse의 정리

q가 소수(prime number) 인 소수체(prime field)에서 정의된 타원 곡선 $E(F_q)$ 의 방정식을 만족하는 점들의 수 N은 다음과 같은 구간에 놓여있다^[4].

$$q+1-2\sqrt{q} \leq N \leq q+1+2\sqrt{q}$$

이 수 N은 항등원 0을 포함한 수이므로 메시지로 사용할 수 있는 점의 수 N_q 는 다음과 같다.

$$q-2\sqrt{q} \leq N_q \leq q+2\sqrt{q}$$

소수체 위에서 정의된 $E(F_q)$ 의 점은 (x, y) 와 $(x, -y)$ 로 짝을 이루고 있으므로 메시지로 사용할 수 있는 x의 수는 $N_q/2$ 이다. $E(F_q)$ 에서 x의 범위는 0부터 q-1까지 이다. 따라서 $E(F_q)$ 에서 메시지 x가 존재할 확률은 다음과 같다.

$$(q-2\sqrt{q})/2q \leq N_q/2q \leq (q+2\sqrt{q})/2q$$

위의 부등식은 다음과 같이 정리된다.

$$1/2-1/\sqrt{q} \leq N_q/2q \leq 1/2+1/\sqrt{q}$$

또한 소수 q를 n 비트의 수라고 할 때 다음과 같이 q의 범위로부터 \sqrt{q} 는 대략 n/2비트 자리의 수가 된다.

$$2^n \leq q < 2^{n+1}$$

$$2^{-\frac{n+1}{2}} < \frac{1}{\sqrt{q}} \leq 2^{-\frac{n}{2}}$$

이를 위의 부등식에 대입하면 다음과 같다.

$$1/2-(1/2)^{n/2} \leq N_q/2q \leq 1/2+(1/2)^{n/2}$$

n이 크면 클수록 $E(F_q)$ 에 메시지가 존재할 확률은 1/2에 가까워지는 것과 이 점들은 범위 양 끝점 가까이에서 밀도가 줄어드는 것을 제외하고 골고루 분포하는^[5~7] 사실에서 모든 공개키에 대하여 공통 메시지로 보낼 수 있는 공유키가 항상 존재하지 않음을 알 수 있다. 또한 이는 ECC ElGamal 알고리즘으로 공통 메시지를 전달할 수 없음을 의미한다.

나. 메시지 존재 판단

임의로 생산된 값 x에 대하여 타원방정식을 만족하는 y 존재는 다음과 같은 Euler - Fermat 판정^[8]으로 간단히 할 수 있다. ECC 공개키로 구성된 3차 방정식을 다음과 같이 f(x)로 정의 할 때

$$f(x) = (x^3+ax+b) \bmod q$$

x에 대한 y 존재 필요충분조건은 다음과 같다.

$$g(x) = f(x)^{\frac{q-1}{2}} \bmod q = 1$$

q보다 작은 임의 정수의 집합 $N = \{x_1, x_2, x_3, \dots, x_k\}$ 에 대해서 메시지로 사용가능성을 검토할 때 $g(x_i) = 1$ 을 만족하는 경우만 한 쌍의 좌표 (x_i, y_i) 를 메시지로 만들 수 있다. 위와 같은 메시지 존재 판단은 ECC 공개키 중 q, a, b만으로 할 수 있다.

3. 공유키 존재

가. 용어정의

n비트 공개키 암호에서 메시지의 크기가 m비트라고 하자. 여기서 수의 크기는 $n > m > 0$ 이다. 이 때 x좌표는 두 부분으로 구성된다. Table 1에서 가장 오른쪽부터 m비트 공간을 메시지 공간이라 하고 그 나머지 n-m비트를 여유 공간이라 하자. 예를 들면 160비트 ECC에서 공유키로 사용 될 메시지의 크기가 128비트이면 우측에서 128비트는 메시지 공간이고, 그 후부터 좌측 끝까지 32비트는 여유 공간이다. 메시지 공간은 공유키 M_x 로 고정한다. 좌측 여유 공간에 임의의 수를 할당하여 공개키의 타원방정식을 만족하는 메시지가 존재할 때 변형된 메시지 (M'_x, M'_y) 를 확장된 메시지라 하며, 이 절차를 메시지 확장이라 하자.

Table 1. x 좌표 구성

	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
$n - m$ 비트(여유 공간)	m 비트(메시지 공간)

다. 메시지 확장

k 개의 공개키로 하나의 고정된 공통 메시지 전달 가능한 조건을 검토한다.

1) 메시지의 조건

ECC 공개키 집합 E 에 대해 공통으로 전달할 수 있는 메시지 M 의 존재를 살펴보자.

$$E = \{ P_i \mid i = 1, k \}$$

$$\text{where } P_i = \{ q_i, a_i, b_i, Q_i, k_i Q_i \}$$

E 에서 소수체(prime field)를 이루는 q_i 중에서 가장 작은 수를 q_{\min} 라 하자. 공통의 메시지 M 은 $M < q_{\min}$ 관계를 만족해야 한다.

2) 메시지 확장 특성

임의 공개키 P_i 로 이루어지는 3차 방정식을 다음과 같이 정의하자.

$$f_i(x) = (x^3 + a_i x + b_i) \bmod q_i$$

공개키 P_i 에 대해서 메시지가 공개키로 이루어진 타원함수의 원소를 이를 확률은 Hasse의 정리에 의해서 $1/2$ 이다. 이는 메시지 M 에 대해서 E 의 원소 중 약 반 정도가 타원방정식을 만족하는 $(M, \sqrt{f_i(M)})$ 이 존재함을 의미한다. 이 집합을 다음과 같이 E_0 이라 하자. 그리고 그 차집합을 E'_0 이라 하자. E_0 의 원소의 수를 $|E_0|$ 로 표시하자.

$$E_0 = \{ P_j \mid \exists y_j \in Z_{q_j}, y_j^2 = f_j(M) \}$$

$$E'_0 = E - E_0$$

$$|E_0| \cong k/2$$

$$|E'_0| \cong k - k/2 = k/2$$

메시지 M 을 만족하지 않는 공개키로 구성된 E'_0 에 대해서 메시지 여유 공간을 변형하여 새로운 확장된 메시지 M_1 을 만들자. 새로운 메시지를 E'_0 에 적용했

을 때 타원 방정식을 만족하는 공개키로 구성된 집합을 E_1 이라 하자. E'_0 의 원소가 $k/2$ 이므로 Hasse의 정리에 의해서 역시 반 정도인 $k/2^2$ 가 새로운 메시지에 대해서 타원방정식을 만족한다. 새로운 메시지에 대해서 타원방정식을 만족하지 못하는 원소의 집합을 E'_1 이라 하자.

$$E_1 = \{ P_j \in E'_0 \mid \exists y_j \in Z_{q_j}, y_j^2 = f_j(M_1) \}$$

$$E'_1 = E'_0 - E_1$$

$$|E_1| \cong k/2^2$$

$$|E'_1| \cong k - k/2 - k/2^2 = k/2^2$$

위와 같이 메시지 공간은 변형하지 않고 여유 공간을 변형하는 메시지 확장 절차를 반복하여 새로운 메시지 M_i 를 만들었을 타원 방정식을 만족하는 공개키로 구성된 집합을 E_i 이라 하고, 확장된 새로운 메시지 M_i 에 대해서 타원방정식을 만족하지 못하는 원소의 집합을 E'_i 이라 하자. 귀납적 방법과 Hasse의 정리에 의해서 다음과 같이 정리된다.

$$E_i = \{ P_j \in E'_{i-1} \mid \exists y_j \in Z_{q_j}, y_j^2 = f_j(M_i) \}$$

$$E'_i = E'_{i-1} - E_i$$

$$|E_i| \cong k/2^{i+1}$$

$$|E'_i| \cong k - k/2 - k/2^2 - \dots - k/2^{i+1} = k/2^{i+1}$$

공개키의 수와 여유 공간을 변형한 메시지 확장 결과는 다음 Table 2와 같이 정리할 수 있다.

Table 2. 메시지 확장 회수와 통계적 계산량(다수 공개키/단일 메시지)

메시지 확장 회수 i	$ E_i $	$ E'_i $	단계별 서버 판정계산 총량
0	$k/2$	$k/2$	$1 \cdot k/2$
1	$k/2^2$	$k/2^2$	$2 \cdot k/2^2$
2	$k/2^3$	$k/2^3$	$3 \cdot k/2^3$
...
$i - 1$	$k/2^i$	$k/2^i$	$i \cdot k/2^i$
i	$k/2^{i+1}$	$k/2^{i+1}$	$(i + 1) \cdot k/2^{i+1}$

가) 메시지 확장 최대 수

메시지 확장 여부는 $|E'_i|$ 에 의존한다. 확장된 메시지에 대해서 만족하지 않는 ECC 공개키의 수가 최소 1 이하가 되는 $i-1$ 단계에서 한 번 더 확장할 경우 통계적으로 i 단계에서 모든 공개키가 메시지 M 을 전달할 수 있다.

$$|E'_{i-1}| \cong k - k/2 - k/2^2 - \dots - k/2^i = k/2^i \leq 1$$

$$\therefore i \geq \log_2 k$$

따라서 서버가 k 개의 공개키로 동일한 메시지 N 을 전달하는 최악의 경우 특정한 공개키 한 개는 통계적으로 적어도 $\log_2 k$ 번 이상의 메시지 확장으로 메시지를 전달할 수 있다.

나) 메시지 존재 판정 최대 횟수

서버는 k 개의 공개키로 메시지를 전달하기 위해, 위에서 계산한 i 번의 메시지 확장이 필요하고, 이에 따라서 메시지 확장여부를 판정하는 총 횟수 $2k$ 는 다음과 같다. 이 때 $|E'_i|$ 는 1보다 작은 정수이므로 0이 된다. ($r=2^{-1}$ 로 치환하고 식을 정리한다.)

$$\sum_{j=1}^i (kj/2^j) = \frac{k}{2} \sum_{j=1}^i \frac{j}{2^{j-1}} = \frac{k}{2} \sum_{j=1}^i jr^{j-1}$$

$$= \frac{k}{2} \sum_{j=1}^i \frac{d}{dr} r^j = \frac{k}{2} \frac{d}{dr} \sum_{j=1}^i r^j$$

$$= 2k - 2(i+1)kr^i + ikr^{i+1} \approx 2k$$

($\because r=2^{-1}, kr^i \approx 1, ikr^{i+1} \approx 0, k \gg i$)

다) 메시지 전달 조건

타원 소수체를 정의하는 q_{\min} 은 n 비트로 구성되므로 $q_{\min} > 2^{n-1}$ 이다. 따라서 실제로 메시지를 확장할 수 있는 여유 공간 $(n-1-m)$ 비트에 할당되는 경우의 수는 메시지 판정 최대 횟수보다 커야 하므로 공개키 수 k 와 관계는 다음과 같으며 이 조건이 만족할 때 k 개의 공개키로 공통 메시지 전달이 가능하다.

$$2^{(n-1-m)} \geq 2k$$

$$\therefore 2^{(n-m-2)} \geq k$$

라. 암호화 시간 구성

ECC는 앞에서 검토한 메시지 확장 조건과 절차가 추가되어야 메시지를 보낼 수 있다. 공개키 k 개 각각에 대해 서버가 공통의 메시지를 보낼 때 효율성에 주된 영향을 주는 사항을 검토한다.

160비트 ECC 공개키에서 메시지 존재를 판정하는 평균시간을 t_{ECC0} , 메시지가 존재할 때 메시지 좌표 (x,y) 를 계산하는 평균시간을 t_{ECC_m} 이라 하자. 또한 암호화에 소요되는 평균시간을 t_{ECC1} 복호화에 소요되는 평균시간을 t_{ECC2} 이라 하자.

공개키 k 개에 대해서 서버가 암호화 가능여부를 판정하는 시간은 $2k \cdot t_{ECC0}$ 이고, 메시지 찾는 데 소요되는 시간은 $k \cdot t_{ECC_m}$ 이며 암호화에 필요한 시간은 $k \cdot t_{ECC1}$ 이다.

암호화에 소요되는 시간은 다음과 같다.

$$k \cdot (2t_{ECC0} + t_{ECC_m} + t_{ECC1})$$

4. 실험

실용적인 공개키와 메시지를 선택하여 업무용 컴퓨터를 KDC 서버로 실험함으로 이론과 실재를 확인한다. 실험은 서버가 다수의 160비트 ECC를 대상으로 공통의 메시지만 128비트 블록암호의 비밀키를 전달하는 과정에서 공유키 존재와 메시지 확장 최대 수, 존재 판정 최대 수를 측정한다.

또한 서버가 메시지를 160비트 ECC 암호화로 공유키를 전달할 때 소요되는 시간을 측정하여 실용성도 검토한다.

가. 실험 장비

공유키 분배 서버로 실험에 사용된 장비는 노트북 (Intel Celeron M 1.40 GHz, 496MB, window XP) 이다.

나. 메시지 생산

메시지는 visual C++의 함수 rand를 이용해서 생산했다. 함수 rand가 생산하는 값의 범위는 0부터 32767이다. 이는 2^{16} 진법으로 표기할 때 32768부터 65535를 표기할 수 없으므로 rand()에 숫자 13을 곱하여 65536을 나눈 나머지를 난수로 고루 분포되게 메시지를 생산하였다.

다. 공개키 생산

공개키는 160비트 유사소수(pseudo prime number)를 Miller 수수 판정법 알고리즘^{[9],[10]}으로 10,000개 생산하여 파일로 저장한 후, 파일에서 유사소수 q 를 하나씩 선택하여, a 와 b 는 메시지 생산과 같은 방식으로 16 비트 난수를 10개 만들어 160비트 임의의 수로 할당한 후 증근 조건을 판정하여 적합하면 공개키로 선택하였다.

라. 공유키 전달

1) 실험 내용

128비트 공유키는 메시지 생산과 같은 방식으로 167개 생산하여 공유키 존재 가능성을 실험하였다. 실험은 128비트 공유키 하나에 대해서 10,000개의 160비트 ECC 공개키를 적용하여 Euler - Fermat 판정하였다.

위와 같은 과정을 167개의 공유키를 대상으로 1,670,000개의 공개키로 공유키를 전달하는 실험을 통해 공유키 존재, 공유키 확장 최대수, 공유키 존재판정 최대 횟수를 측정하여 이론과 실제를 비교하였다^[11].

2) 결과

메시지 확장 과정을 포함하여 하나의 128비트 공유키를 10,000개의 각기 다른 160비트 ECC 공개키 P_i 로 전달 할 때, 공개키로 구성된 타원 E_i 에서 메시지 혹은 확장된 메시지가 존재하지 않아 전송이 불가능한 공유키는 없었다. 이론적으로 공개키의 수 k 가 $2^{160-128-2}$ 보다 작으면 항상 전달 가능한 이론적 예측 결과가 타당함을 보인다.

이는 160비트 ECC로 실용적 공유키 전달이 가능함을 의미한다. Table 3은 공유키 최대 확장과 적합성 판정횟수를 1,670,000개 공유키 실험결과를 평균값으로 정리한 내용이다. 가장 왼쪽 k 와 숫자는 고정된 128비트 공유키에 대해 처음 100, 1,000, 10,000개의 공개키로 분배함을 의미한다.

가장 윗줄의 수 i 는 고정된 공유키에 대하여 여유 공간에 임의수를 할당한 메시지 확장 횟수를 의미하며, 윗줄의 MAX는 여유 공간 확장 최대 횟수, total은 공개키 k 개에 대해서 Euler-Fermat 판정하는데 소요된 총 횟수를 의미한다. Table 3에서 제시된 수는 k 개에 대한 반복실험 167회 실험 결과의 평균이며 소수점의 수는 반올림 하였다. 실험결과 이론과의 비교는 다음과 같이 요약된다.

- ① 메시지 확장 없이 공유키를 보낼 수 있는 공개키는 k 개 중에서 반 정도인 $k/2$ 개가 메시지 전달 가능함을 보여 예측된 이론 결과와 근사하다.
- ② 여유 공간에 난수를 할당하는 i 단계에서 공개키로 확장된 공유키를 보낼 수 있는 공개키 수는 대략 $k/2^{i+1}$ 이며 이론적 결과에 근사한 값을 보였다.
- ③ 메시지 확장을 포함하여 서버가 모든 공개키에게 공유키를 보낼 경우 Euler - Fermat를 판정한 총 횟수(total)는 이론적 결과에 근사한 $2k$ 이다.
- ④ 공유키를 분배하기 위한 메시지 최대 확장 횟수 MAX는 $k=100$ 에서 평균값 $i=6$ 이며, 167회 실험에서 나온 최대치는 7이다. 이보다 큰 값 $k=1,000$ 과 $k=10,000$ 에서는 평균값이 $i=10$ 과 $i=13$ 이며, 167회 실험에서 나온 최대치는 10과 14이다. 이는 $\log_2 k$ 의 정수 값 근사치와 유사하거나 약간 큰 정수임을 보이면서 이론적 결과와 일치한다.

바. 시간측정

클라이언트의 160비트 ECC 공개키를 서버가 받아 메모리에 저장한 후 공유키 분배하는 환경을 가정하여 필요한 평균시간을 측정하였다.

1) 실험 내용

KDC에서 160비트 ECC 공개키로 동일한 공유키를 분배할 경우 서버가 키 분배에 소요되는 시간은 다음과 같이 3 단계로 구성된다.

Table 3. 평균값

$k \backslash i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	MAX	total
100	51	25	12	6	3	1	1								7	198
1000	503	250	124	61	31	15	8	4	2	1	1				10	1991
10000	5003	2498	1253	623	311	155	79	39	19	10	5	2	1	1	14	19987

- ① 한 개의 공개키에 대해서 Euler - Fermat 판정으로 메시지 존재확인애 소요되는 시간 t_{ECC0}
- ② 메시지가 존재할 경우 공유키 x 에 대한 y 값을 찾음으로 메시지 구하는데 소요되는 시간 t_{ECC_m}
- ③ 메시지를 ECC ElGamal알고리즘으로 암호화에 소요되는 시간 t_{ECC1}

통계적으로 ①단계는 공개키 k 개에 대해서 서버는 약 $2k$ 개의 Euler - Fermat 판정을 하며 ②, ③단계는 공개키 k 개에 대해서 서버는 k 개의 계산이 소요된다.

가) 공유키 발견시간

공유키 실험에서 메시지 확장 검사 ①단계와 메시지 x 에 대한 y 값 발견 ②단계 시간소요를 함께 측정하였다. 메시지에 대해 10,000개 ECC공개키로 전달하는 실험을 167회 실시하여 공개키 당 공유키 발견 평균 시간을 측정하였다.

나) 암호화 시간

KDC가 클라이언트에게 공유키를 보내기 위한 암호화 시간을 측정하였다. 측정은 암호화 시간과 복호화 시간을 별도로 측정하였다. 실험 방법은 유사 소수와 난수로부터 생성된 공개키를 선택하고, 메시지를 찾은 후 암호화 시간과 복호화 시간을 측정하였다. 정확한 측정을 위하여 메시지 100,000개에 대한 암호화 시간과 복호화 시간을 종합하여 평균치를 계산하였다.

2) 결과

167개의 128비트 메시지를 보내는 실험에서 1개의 공유키가 ①단계와 ②단계를 완성하는데 소요되는 시간은 공개키 당 평균 0.078483sec 이다.

암호화에 소요된 시간은 평균 0.40378sec 이며 복호화에 소요된 시간은 평균 0.22978sec 이다. ElGamal 알고리즘은 암호화에 두 번의 곱셈과 한 번의 덧셈 절차가 필요하고, 복호화에 한 번의 곱셈과 한 번의 덧셈이 소요되는 계산에서 암호화/복호화 시간 비율은 타당하다.

따라서 서버가 클라이언트 당 공유키 발송에 소요되는 평균시간은 $0.078483+0.40378 = 0.48226sec$ 이다.

암호화 시간의 20% 정도 추가비용으로 160비트 ECC ElGamal 알고리즘에서 128비트 실용적 규모의 공유키 전달이 가능하며 복호화에는 별도의 비용이 소요되지 않는다.

5. 결론

본 연구는 서버가 다수의 ECC 공개키로 공유키를 전달할 수 없는 문제점을 지적하고, 실용적으로 활용할 수 있는 조건과 방법을 수학적으로 제시하였다.

이론적으로 제시된 결과는 실용적 수준의 공유키 128비트 메시지를 160비트 ECC로 전달 가능함을 실험으로 증명하였다. 128비트 메시지 당 1만 개의 160비트 ECC 공개키로 공유키를 분배하는 실험을 167회 실시하여 이론과 실제가 일치됨을 통계적으로 보였다. 시간적으로 서버가 공유키를 전달할 때 암호화 시간의 약 20%정도를 추가비용으로 부담함으로 ECC ElGamal 알고리즘을 키 공유에 실용적으로 사용할 수 있음도 보였다.

또한 업무용으로 보급된 컴퓨터를 KDC 서버로 실험함으로 국방망에서 추가비용 없이 실용적 사용가능함을 보였다. 연구된 결과는 NCW와 같이 기밀성을 요하는 특정집단의 임무수행에서 자료공유 및 실시간 보안회의 등에 필요한 블록암호 비밀키를 다자간 공유키로 신속히 분배하는 데 활용할 수 있다.

후 기

본 연구 논문은 08년도 공군사관학교 국고연구비 (KAFA 08-14) 예산지원으로 수행된 결과입니다.

References

- [1] 강주성 외 6명, 「현대암호학」 (경문사), p. 178, 2000.
- [2] 홍성룡, 조정호, “SDR System 적용을 위한 한국형 암호 알고리즘(SEED) 구현 및 성능분석”, 한국정보과학회, 2002.
- [3] 이민섭, 「현대암호학」 (교우사), p. 375, 2002.
- [4] 김창한, 서광석, 「암호학과 대수학」 (북스힐), p. 356, 1999.
- [5] H. W. Lenstra, Jr. Factoring Integers with Elliptic Curves, *Annals Math.* 126, 649673, 1987.
- [6] W. Waterhouse, Abelian Varieties Over Finite Fields, *Ann. Sci. E'cole Norm. Sup.* 2, pp. 521~560, 1969.
- [7] 서광석, 김창한, 「초보자를 위한 암호와 타원곡선」 (경문사), p. 155, 1998.

- [8] 박승안, 「대수학과 암호학」 (경문사), p. 21, 1999.
- [9] 이 준, RSA 공개키 암호의 실용적 구현에 관한 연구(연구보고서 KAFA 02-1-3-9), 공군사관학교, 2002.
- [10] 이 준, ECC 공개키 암호의 실용적 구현에 관한 연구(연구보고서 KAFA 05-23), 공군사관학교, 2005.
- [11] 이 준, ECC를 통한 키분배에서 공유 대칭키 존재에 관한 연구(연구보고서 KAFA 08-14), 공군사관학교, 2008.