ECSET 설계를 위한 타원곡선 알고리즘

양승해* , 이병관**
*관동대학교 전자계산공학과
**관동대학교 컴퓨터공학과
e-mail:yang7177@mail.kwandong.ac.kr
bklee@mail.kwandong.ac.kr

An Elliptic Curve Algorithm for designing ECSET

Seung-Hae Yang* , Byung-Kwan Lee**
*Dept of Computer Science, Kwandong University
**Dept of Computer Engineering, Kwandong University

요약

SET보안 프로토콜은 DES, RSA, Hash 알고리즘으로 구성되어있는데, 본 연구는 기존의SET에서 사용되는 RSA알고리즘대신에 ECC알고리즘을 연구 개발하였고, 이것을 RSA와 속도면에서 성능을 비교 분석하였다.

1. 서론

웹 보안 프로토콜인 SSL(Secure Socket Layer)에서 객체들간의 인증 기능을 강화한 SET(Secure Electronic Transaction) 메커니즘은 통신 응용간의 기밀성(confidentiality), 메시지 무결성(data Integrity) 부인방지(non-repudiation), 인증(authentication)서비스를 만족시켜준다.[1]

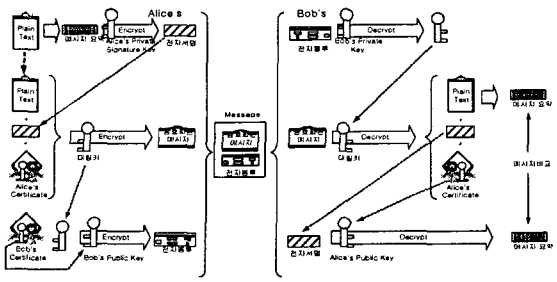
기존의 SET메커니즘은 공개키 기반으로 곱셈연산을 이용하여 암호화를 수행하는 RSA인데 반해서 본 연구의 ECSET(Elliptic Curve Secure Electronic Transaction)에서는 성능이 보다 향상된 덧셈연산을 주체로 하는 ECC(Elliptic Curve Cryptography)를 구현함으로써 기존의 SET보다 훨씬 나은 암호화의 속도를 향상시키는 알고리즘을 설명한다. 2장에서는 SET프로토콜을 구성하고 있는 객체들과, 객체들 간에 처리되는 암호의 종류와 과정을 보인다. 3장에서는 SET프로토콜을 구성하는 암호화 방법 중 공개키 암, 복호화 알고리즘 즉, RSA 알고리즘과 ECC 알고리즘에 대하여 보이고, 4장에서는 시뮬레이션 및 결과를 보인다. 그리고 5장에서는 결론을 설명한다.

---

* 준 회 원 : 관동대학교 전자계산공학과
** 종신회원 : 관동대학교 컴퓨터공학과 교수

2 ECSET 암호 메커니즘

SET은 공개키 암호, 대칭키 암호, 서명, 해쉬 등을 사용하여 안전성을 보장해 주게 된다. 공개키암호방식은 키를 두 개로 나누어 하나는 암호화 키로 또 하나는 복호화 키로 사용한다. 대칭키 암호화 방식은 관용 암호 방식이라고도 하며 암호화 키와 복호화 키가 일치한다. 즉, 송신자 A가 수신자 B에게 평문을 암호화하여 메시지를 보낼 때 쓰여진 키와 수신자 B가 암호문을 평문으로 바꾸는 데 쓰는 키가 동일하다는 것이다. 디지털 서명은 서명이나 인장의 효과를 전자적 매체내에 저장하여 전송하는 전자문서의 효과적인 서명 방식을 말한다. 그리고 해쉬 알고리즘은 서명문 압축과 디지털 서명을 효율적으로 계산할 수 있는 방법을 제공하고 있다

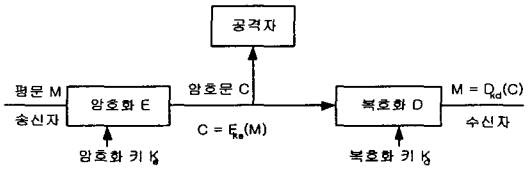


[그림 2-1] ECSET의 암호 처리 과정

3. 공개키 암/복호화 알고리즘

3.1 암호방식

정보를 도청자(공격자)로부터 보호하기 위한 방식이 암호방식인데 구성은 [그림 3-1]와 같다.



[그림 3-1] 암호 방식의 구성

평문(plaintext)은 송신자가 수신자에게 전달하려는 정보 내용으로 누구나 그 의미를 알 수 있는 정보이다. 송신자는 평문 M을 암호화키 K_e와 암호(Encryption)알고리즘을 적용시켜 암호문 (ciphertext) C를 생성하여 수신자에게 전달한다.

$$C = E_{ke}(M)$$

수신자는 송신자가 전송한 암호문 C를 수신하여 복호화키와 복호화(decryption)알고리즘을 적용시켜 송신자가 전송하려는 평문 M을 복원한다.

$$M = D_{kd}(C)$$

3.2 RSA와 ECC 알고리즘

(1) RSA 알고리즘

RSA는 합성수의 소인수 분해의 어려움을 이용한 암호화 기법이다. 가입자는 백 자리 크기 이상의 두 개의 소수 p, q를 선택하여 $n = p \cdot q$를 계산한다. 이때 p와 q를 알고 있는 사람은 n을 계산하기 쉽지만 n만 알고 있는 사람은 n으로부터 p와 q를 찾는 소인수 분해는 어렵다.

p와 q를 선택하여 n을 계산한 다음 Euler 함수 값 $\Phi(n) = (p-1)(q-1)$와 서로소인 K_e를 선택한다. 다시 유클리드 알고리즘을 이용하여 다음 식을 만족하는 K_d를 계산한다.

$$K_e \cdot K_d \equiv 1 \bmod \Phi(n)$$

K_e와 n은 공개 목록에 등록하여 공개하고 K_d는 비밀리에 보관한다. 즉, K_e는 공개 암호화 키가 되고 K_d는 비밀 복호화 키가 된다. RSA 암호 방식의 구성 절차와 암호와, 복호화 과정은 [그림 3-2]와 같다.

가입자 B에게 평문 M을 비밀리에 전달하려는 가입자 A는 공개목록에서 가입자 B의 공개 암호화 키 K_{eB}를 찾아, 암호문 $C \equiv M^{k_{eB}} \bmod n_B$를 계산하여 가입자 B에게 전송한다. 가입자 B는 가입자 A로부터 수신한 암호문 C를 자신이 비밀리에 보관하고 있는 복호화 키 K_{dB}로 평문 $M \equiv C^{K_{dB}} \bmod n_B$를 복원한다.

물론 가입자 B가 가입자 A에게 평문을 비밀리에 전송하려면 가입자 A의 공개 암호화 키 K_{eA}를 공개 목록에서 찾아 암호문 $C \equiv M^{K_{eA}} \bmod n_A$를 계산하여 가입자 A에게 전송한다. 가입자 A는 가입자 B로부터 수신한 암호문 C를 자신이 비밀리에 보관하고 있던 비밀 복호화 키 K_{dA}로 평문 $M \equiv C^{K_{dA}} \bmod n_A$을 복원한다.

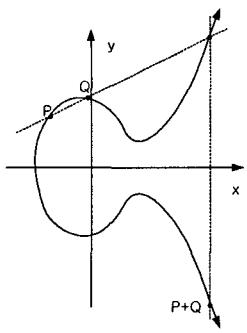
가입자 A		가입자 B
p_A, q_A $n_A = p_A \cdot q_A$ $\Phi(n_A) = (p_A-1)(q_A-1)$ $\gcd(K_{eA}, \Phi(n_A)) = 1$ $K_{eA} \cdot K_{dA} \equiv 1 \bmod \Phi(n_A)$		p_B, q_B $n_B = p_B \cdot q_B$ $\Phi(n_B) = (p_B-1)(q_B-1)$ $\gcd(K_{eB}, \Phi(n_B)) = 1$ $K_{eB} \cdot K_{dB} \equiv 1 \bmod \Phi(n_B)$
$C \equiv M^{K_{eB}} \bmod n_B$	C →	$M \equiv C^{K_{dB}} \bmod n_B$

[그림 3-2] RSA 공개키 암호 방식

(2) ECC 알고리즘

ECC(Elliptic Curve Cryptographic)는 이미 알려진 공용키 시스템보다 비트 당 더 높은 보안을 제공하기 때문에 최근에 많은 관심의 초점이 되는 분야이다. RSA에 비해 좀 더 작은 키 크기를 갖고있으므로 응용분야에 적용하였을 때 수행속도에서 효과가 있다.

이러한 타원곡선은 유한체 상에 정의된 타원곡선에 대하여 타원곡선군은 3차 방정식을 만족하는 순서쌍들과 무한점을 포함한 집합을 말한다.



[그림 3-3] ECC 그래프

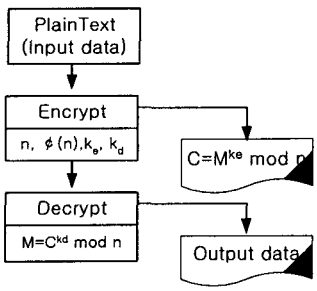
ECC 암호화 방법은 유한체 k위에서 정의된 타원곡선 E위의 점들이 가환군의 형태를 이며 다음과 같은 장점을 가지고 있다.[2],[3],[4],[5]

첫째, 이 군에서의 이산대수 문제는 매우 어렵고, 특히 같은 크기인 k 유한 체에서의 이산대수 문제보다 더 어렵다. 다시 말하면, ECC는 작은 키 길이를 가지고 현존하는 공개키 암호화 방법의 안전도를 보장받을 수 있다. 둘째, 비록 모든 사용자들이 같은 기초에 k를 사용하더라도 각 사용자가 다른 곡선 E를 선택할 수 있다. 이는 주어진 군에서 다양한 타원곡선을 사용할 수 있음을 의미한다.

3.3 RSA와 ECC 구현

(1) RSA 구현모드

[그림 3-4]는 RSA를 구현하기 위한 기본 모드로서 암호화시 평문 내용을 포함하고 있는 Input data라는 특정파일을 읽어 임의의 두 개의 키 p와 q를 선택하여 이를 이용하여 공개키와 개인키를 생성하여 암호화 과정을 수행한 결과로는 암호화된 data를 출력하였다.



[그림 3-4] RSA 구현모드

복호화는 암호화를 역순으로 수행하여 출력 data파일을 입력받아 복호화 루프를 수행한 후 평문을 확인할 수 있다.

(2) ECC 구현모드

본 장에서는 송신자(sender)가 암호화 메시지를 수신자(receiver)에게 보내는 것을 보여준다. 간단하게 메시지는 소수 P보다 작은 정수의 쌍이다. 랜덤한 정수 r과 메시지를 송신코드에 의해 수신자에게 보낸다. 수신자의 복호는 수신자의 비밀키 k로 메시지를 복호한다.

타원곡선 암, 복호화 과정은 다음과 같다.

Step 1] Receiver : Choose p

```
int Select_p()
{
 int is_prime, i; int sqrn; unsigned int p;
 while(1)
 { is_prime = 1;
   p = rand();
   if(p <= 50)
   { sqrn = (int)sqrt(p);
     for(i=2; i<sqrn; I++)
       if(p%i == 0); }
   else
     continue;
   if(is_prime == 1)
     return p' } }
```

Step 2] Receiver : Choose an elliptic curve

$$y^2 = x^3 + bx + c \text{ (Choose b and c)}$$

Step 3] Receiver : Find a point P on the curve

```
Addition (P≠Q)
  PointEC(x1, y1, b, c) + PointEC(x2, y2, b, c)
          := Module(L, IdentityECQ, x3, y3),
           if (IdentityECQ = False)
             {
             x3 = L2 -x1 -x2;
             y3 = L(x1 - x3) -y1;
             L = (y2 - y1)/(x2 - x1);
             Return PointEC(x3, y3, b, c)
             }
```

```
Multiply (P=Q)
  n_Integer * PointEC(x, y, b, c) := Module(dgt =
   IntegerDigits(n, 2), w = PointEC(x, y, b, c),
                        sm = IdentityEC, i},
          i = Length(dgt);
           while( i >= 1)
            { if(dgt(i) == 1)
                 {  sm += w;
                  }
                 w = w+w;
            i--
            } Return(sm);
```

Step 4] Receiver : Choose an integer k

Step 5] Receiver : Calculate kP

Step 6] Receiver : Tell Sender the values p, b, c, P, kP

Step 7] Sender : Choose a random integer r

Step 8] Sender : Calculate rP and $r(kP)$

Step 9] Sender : Let $rP=(i,j)$, $r(kP)=(m,n)$, $d=mx, e=ny$ Tell i,j,d,e to Receiver

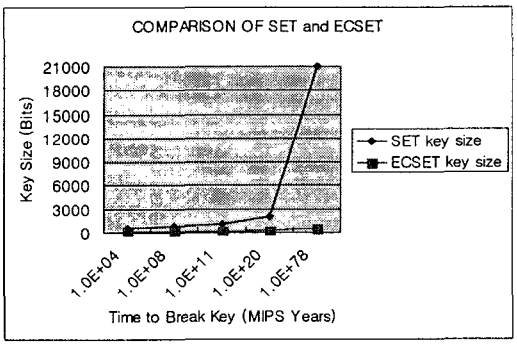
Step 10] Receiver : Let $R=(i,j)$. Calculate $kR=(m,n)$

Step 11] Receiver : Solve the equations $mx=d(\mathrm{mod}\,p)$ and $ny=e(\mathrm{mod}\,p)$

이로서 타원곡선 알고리즘의 암호화 과정과 복호화 과정을 보았다.

4. 시뮬레이션 및 결과

RSA 암호방식은 합성수의 소인수 분해 계산의 어려움을 이용하여 백 자리 이상의 두 개의 소수 p, q를 선택하여 공개키와 비밀키를 구성한 후 평문을 암, 복호화 하는 형식을 갖는다. 반면에 ECC 암호화 방식은 이산대수에서 사용하는 유한체의 곱셈군을 타원곡선군으로 대치한 암호시스템이다. 다음 그림 [4-1] 은 RSA와 ECC의 성능비교 이다. 그림은 같은 크기의 bit를 이용하였을 때 암, 복호화 하는 시간이 ECC가 10^4 일때에는 5:1, 10^8 일때에는 6:1, 10^{11}일때에는 7:1, 10^{20} 일때에는 10:1, 10^{78}일때에는 35:1의 비율을 갖는다.



[그림 4-1] SET와 ECSET의 성능비교

5. 결론

지금까지의 SET프로토콜은 RSA방식을 사용하여 합성수의 소인수분해에 관한 곱셈의 연산으로 암, 복호화의 시간이 길다. 이러한 대책으로 타원곡선을 이용한 ECSET를 개발함으로서 RSA보다 짧은 구현시간을 얻을 수 있고, ECSET은 짧은 키 크기를 사용함으로 타원곡선 암호시스템은 스마트카드나 무선환경에서처럼 상대적으로 작은 키 크기와 제한적 대역폭과 메모리가 요구되는 분야에서 각광받고있는 차세대 암호시스템이다.

참고문헌

[1] Dabid Pointchebal and Jacques Stern "Security Proofs for Signature Schemes" Advances in Cryptology-Proceedings of EUROCRYPT' 96 page387-398

[2] Mugino Saeki "Elliptic Curve Cryptosystems" McGill University, Montreal

[3] K.H.Leung, K.W.Ma, W.K.Wong and P.L.W.Leong "FPGA Implementation of a Microcoded Elliptic Curve Cryptographic Processor" The Chinese University of Hong Kong Shatin, N.T.Hong Kong

[4] Toshio HASEGAWA, Junko NAKAJIMA, Mitsuru MATSUI "A Small and Fast Software Implementation of Elliptic Curve Cryptosystems over GF(p) on a 16-Bit Microcomputer" IEICE TRANS.FUNDAMENTALS,VOL.E82-A, NO.1 JANUARY 1999

[5] Toshio HASEGAWA, Junko NAKAJIMA and Mitsurn MATSUI "A Practical Implementation of Elliptic Curve Cryptosystems over GF(p) on a 16-bit Microcomputer" Information Technology R&D Center Mitsubishi Electric Corporation

[6] Michael J. Wiener, Robert J. Zuccherato "Faster Attacks on Elliptic Curve Cryptosystems" Canada K1V 1A7 April 8, 1998

[7] Gadiel Seroussi "Compact representation of elliptic curve points over F_2^n" Hewlett-Packard Laboratories April 6,1998