공개키 생성 시간 값을 이용한 ECC 인증 프로토콜 설계

김갑열*, 박석천**
*경원대학교 소프트웨어학부
e-mail:scpark@kyungwon.ac.kr

Design of ECC Authentication Protocol using Public Key Generation Time

Kap-Yol Kim*, Seok-Cheon Park**
*Division of Software, Kyungwon University.

요 약

최근 IT 기업들은 최첨단 인프라 기술을 활용한 모바일 단말 생산에 주력하며 시장을 확대하고 있다. 이에 따라 각 선진국의 보안업체들은 모바일 단말에 특화된 보안 기술 확보에 노력하고 있으나 국내의 모바일 단말 보안 기술은 선진국에 미치지 못하고 있다. 따라서 본 논문에서는 경량화 네트워크 단말에서 활용할 수 있는 ECC 암호 알고리즘의 공개키 생성시 시간 값을 활용한 보안 프로토콜을 설계하였다.

I. 서론

최근 이동성을 제공하는 네트워크 단말인 PDA, 휴대폰, 노트북 등의 보급과 확산으로 인해 데이터 통신의 수요가 증폭되고 있다. 따라서 데이터의 안전한 송·수신의 관심이 증폭되고 있으며 특히 지극히 개인적인 정보를 담고 있는 데이터(예, 주민등록 번호)의 보호를 위한 연구가 활발히 진행 중에 있다[1].

그 결과로 지난 1970년대부터 RSA(Rivest Shamir Adleman), DES(Data Encryption Standard), AES(Advanced Encryption Standard) 등의 암호 알고리즘이 개발되어 표준화 되었으며[2] 최근에는 1985년 밀러와 코블리치에 의해 제안된 ECC(Elliptic Curve Cryptography) 암호 알고리즘[3]의 안정성과 효율성이 입증되면서 유비쿼터스 사회 실현을 위한 경량화 네트워크 단말의 데이터 보호 요구사항을 수용할 수 있게 되었다. 특히 ECC 암호 알고리즘은 현재 가장 보편화 된 암호알고리즘인 RSA 보다 H/W, S/W 구현과 기법 변형이 용의하며 또한 암·복호화 속도가 빨라 경량화 네트워크 단말의 데이터 보호를 위해 효율적으로 구현되고 있다. 따라서 본 논문에서는 경량화 네트워크 단말의 효율적 인증을 위해 ECC 암호 알고리즘에서 공개키 생성 시간 값을 활용하여 암호화키를 생성하는 기법을 제안 한다.

* 경원대학교 일반대학원 전자계산학과 석사과정
** 경원대학교 IT대학 교수(교신저자)

II. 관련 연구

2.1 ECC 암호 알고리즘

ECC 암호 알고리즘은 1985년 밀러와 코블리치에 의해 제안된 알고리즘으로 유한체 위에서 정의된 타원곡선 군에서의 이산대수 문제의 어려움에 기초한 암호 시스템이다[3].

일반적으로 160비트 키 사이즈를 가지는 ECC 알고리즘은 1024비트 키 사이즈를 가지는 RSA 알고리즘과 대등한 안전도를 가진다고 알려져 있으며 특히 160비트 키 사이즈와 193비트 키 사이즈를 가지는 ECC 알고리즘은 각각 10년에서 20년 이상 보안강도를 가진 것으로 평가되어 여타 암호 알고리즘에 비해 효율성을 인정받고 있다. 다음 표 2.1은 RSA와 ECC의 키 사이즈에 비례한 보안 강도를 보여준다[3][4].

<표 2.1>. RSA, ECC 키 사이즈 비교

RSA	ECC
1024 bit	160 bit
1048 bit	211 bit
21000 bit	600 bit

위와 같이 ECC 알고리즘은 작은 키를 이용하여 높은 보안강도를 기대할 수 있는 특징으로 제한된 메모리 공간

과 저 전력을 가지는 모바일 환경의 단말 등에 구현하기 용이하다. 또한 ECC 암호 알고리즘은 덧셈 연산을 주로 사용하기 때문에 계산에 대한 속도 역시 빠르며 H/W와 S/W 구현이 용이한 장점을 가지고 있다[5].

2.2 ECC 암호 알고리즘 정의

p개의 원소를 갖는 유한체 GF(p) 상의 ECC 알고리즘은 체의 표수 p가 3 초과인 경우 암호 알고리즘으로서 실효성을 가지며 이때 타원곡선을 적당히 이동하면 다음 식 (1)과 같은 방정식에 만족하게 된다.

$$E: x^3 + ax + b - y^2 \equiv 0 \pmod{p} \quad \left(where,\ p > 3\ AND\ prime,\ a, b \in GF(p)\right) \tag{1}$$

타원곡선의 점들은 x 축을 대칭으로 y 점이 두 개가 존재 하는데 이때 서로 대칭한 두 점은 서로 역원이며 x 축으로 대칭한 y 점이 없을 경우 역원을 구할 수 없으므로 중근으로 정의하여 암호 알고리즘에서는 사용할 수 없다. 위와 같은 식 (1)을 만족하는 타원곡선 E는 GF(p) 상의 모든 점 (x, y)와 무한원점 O로 구성이 되며 암호 알고리즘에서는 이 점들의 덧셈과 뺄셈 연산을 통해 공개키와 메시지를 공유한다. 이때 타원곡선상의 점 P, Q에 대한 연산은 다음 식 (2), (3)으로 연산할 수 있다.

$x_1 \neq x_2$일때

$$k = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$
$$x_3 = k^2 - x_1 - x_2 \pmod{p}$$
$$y_3 = k(x_1 - x_3) - y_1 \pmod{p}$$

$x_1 = x_2\ AND\ y_1 = y_2 \neq 0$ 일때

$$k = \frac{3{x_1}^2 + 2y_1}{2y_1} \tag{3}$$
$$x_3 = k^2 - x_1 - x_2 \pmod{p}$$
$$y_3 = k(x_1 - x_3) - y_1 \pmod{p}$$

식 (2)는 타원곡선상의 점 P=(x1, y1)와 Q=(x2, y2)의 덧셈 연산을 나타내며 P+Q=R(x3, y3) 로 정의한다. 식 (3)은 타원 곡선상의 한 점 P에 대한 덧셈 연산 P+P가 되며 2P=R(x3, y3)로 나타낼 수 있다. 이와 같이 식 (2), (3)을 이용하여 P+P+P=3P를 구하면 P+P=2P, 2P=Q, P+Q=R=3P(x3, y3)로 정의 할 수 있고, 다시 2P를 구하기 위해선 R+(-P)=2P로 연산하면 된다. 같은 방법으로 P+P+P+P... =kP를 구할 수 있고 ECC 암호 알고리즘은 난수 k를 알아내기 힘들다는 문제를 이용하며 이때 -P는 다음 식 (4), (5)로 정의 한다.

$x > 0\ AND\ y > \frac{p}{2}$ 일때

$$x_2 = x \tag{4}$$
$$y_2 = \frac{p}{2} - (y - \frac{p}{2})$$

$x > 0\ AND\ y < \frac{p}{2}$ 일때

$$x_2 = x \tag{5}$$
$$y_2 = \frac{p}{2} + (\frac{p}{2} - y)$$

III. 공개키 생성 시간 값을 이용한 ECC 인증 프로토콜의 설계

3.1 제안 프로토콜 정의

본 논문에서 제안하는 인증 프로토콜은 모바일 컴퓨팅 환경에서 가장 효율적으로 알려진 ECC 암호 알고리즘을 이용하였으며 제안 프로토콜의 파라메터 정의는 다음 표 3.1과 같다.

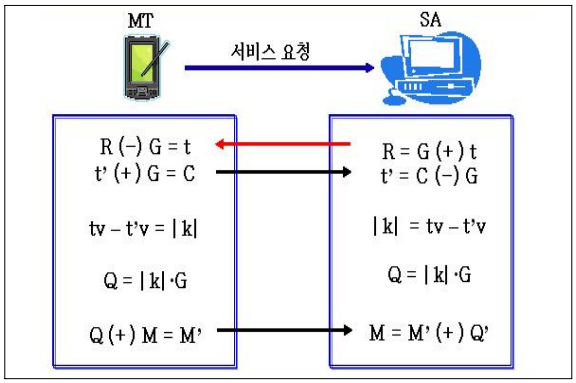
<표 3.1> 제안 프로토콜 파라메터

파라메터	설 명
MT, SA	모바일 단말, 서버
E(GF(p))	유한체 GF(p) 위에서 a, b ∈ GF(p)에 의해 정의된 타원곡선
p	타원곡선이 정의되는 유한체의 크기
a, b	타원곡선을 결정하는 방정식의 상수이며 GF(p)에 속한 원소
G, (x, y)	타원곡선 상의 임의 점, 타원곡선 한 점의 좌표
|k|	시간 값의 차에 의해 생성된 난수의 절대 값
t	x좌표 값을 0, y좌표를 호출되는 시간 값으로 가지는 점
t′	x좌표 값를 t의 y좌표 값, y좌표를 호출되는 시간 값으로 가지는 점
tv, t′v	t의 y 값, t′의 y 값
C, R	MT의 공개키, SA의 공개키
M, Q, Q′	서명될 메시지, 암호키, 복호키(Q의 역원)
(+), (−)	타원곡선상의 덧셈, 타원곡선상의 뺄셈

제안 프로토콜은 160비트 이상의 키를 가지는 ECC 암호 알고리즘을 이용하기 때문에 체수 p를 160 비트 이상의 소수를 사용하고 모바일 단말과 서버는 미리 서로 공유한 형태로 가정한다. 또한 타원곡선 상의 임의의 점 G

와 상수 a, b 값 역시 서로 알고 있다고 가정하고 프로그램 설치 시 서로 동기 된다.

제안 프로토콜 주요 키워드는 공개키 생성 시 시간 값을 호출하여 타원곡선상의 점과 연산하는 과정을 거치기 때문에 호출되는 시간 값을 x, y 값을 가지는 좌표 형태로 정의하여야 한다. 따라서 t를 구하기 위해서는 현재 호출되는 시간 값, 예를 들어 2008년 11월 21일 12시 31분 22.999초라고 가정하면 t의 x좌표 값은 0, y좌표 값은 20081121123122999을 가지게 된다. 다음 그림 3.1은 본 논문에서 제안하는 ECC 암호 알고리즘을 이용한 인증 프로토콜의 단계별 처리 절차를 나타낸다.



(그림 3.1) 제안 ECC 프로토콜 단계별 처리 절차

그림 3.1에서 최초 (MT)가 서비스를 요청하면 (SA)는 현재의 시간 값을 y좌표 값으로 가지는 t를 생성해 낸다. 이후 타원곡선의 임의의 한 점 G와 (+) 연산을 하여 공개키 R을 생성하고 (MT)에 송신한다. (SA)의 공개키 R을 수신한 (MT)는 G 값과 (-) 연산하여 t를 복원해 낸다. t의 y좌표 값은 tv 변수에 등록하고 다시 tv를 t′의 x좌표 값으로 설정한 후 현재 시간 값을 y 좌표 값으로 t′를 생성해 낸다. 생성한 t′는 G와 (+) 연산하여 공개키 C를 생성한 후 (SA)에 송신하고 tv-t′v 정수 연산하여 난수 k를 구한다. (MT)의 공개키 C를 수신한 SA 역시 C (-) G를 연산하여 t′를 구하고 tv-t′v 연산하여 k를 구한다. 이후 (MT)와 SA는 각각 G를 k만큼 스칼라 곱하여 Q를 구하고 SA는 복호화 키 Q′까지 구한 후 (MT)가 Q와 M을 (+) 연산하여 송신한 M′를 Q′와 (+) 연산하여 M을 복원해 낸다.

IV. 공개키 생성 시간 값을 이용한 ECC 인증 프로토콜의 구현

3.2 제안 프로토콜 구현

본 논문에서 제안하는 인증 프로토콜은 노트북과 데스크 탑에 구현하였으며 개발 언어는 C와 C++를 사용하였다. 상세한 구현 환경은 다음 표 4.1과 같다.

<표 4.1> 제안 프로토콜 구현 환경

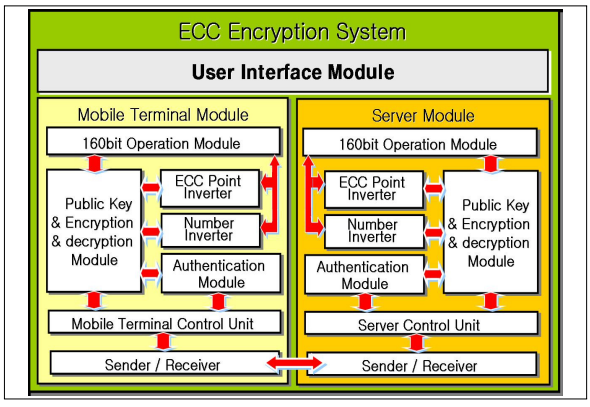
구 분	구성 요소		사 양
H/W	SA	CPU	Intel Duo Core 2.66Ghz
		RAM	2GB
		이더넷 카드	Realtek RTL-8168
		그래픽 카드	GeForce 7300
	MT	CPU	Intel Core2 2.00Ghz
		RAM	512MB
		이더넷 카드	Intel PRO 3945ABG
		그래픽 카드	Mobile Intel 945GM Chip
S/W	운영체제		Windows XP pro
	개발 플랫폼		Visual Studio 6.0

본 논문에서 제안하는 프로토콜의 구현은 향후 PDA나 휴대폰과 같이 경량화 된 단말에 동작이 가능하도록 시간 값을 호출하는 라이버러리 외에 다른 라이버러리를 사용하지 않았으며 기본 변수 타입 이상의 연산을 위해 160비트 정수의 4칙 연산이 가능한 함수를 구현하였다. 또한 비슷한 기능을 하는 키 생성, 암호화, 복호화 등의 연산에서는 인자 값의 구분에 의해 다른 기능을 수행하도록 소스의 중복을 최대한 줄였고 모듈별 기능을 철저하게 나누었다.

<표 4.2> 고정 파라메터(fix parameter)

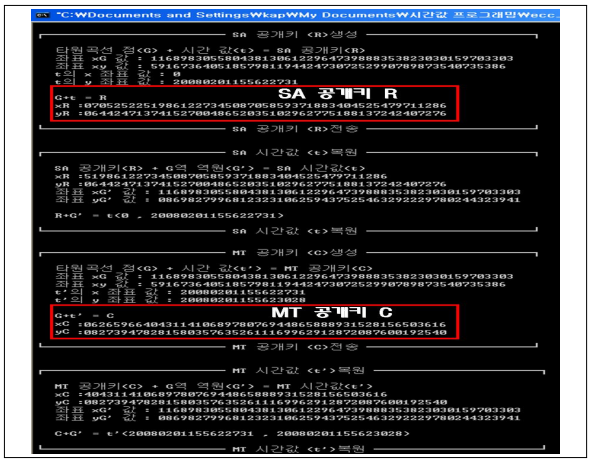
구분	값
p	1461501637330902918203684832716283019653785059327
a	1461501637330902918203684832716283019653785059324
b	618161358937170673988121756987436237099350920727
x_G	1168983055804381306122964739888353823030159703303
y_G	591673640518579811944247307252990789873540735386

본 논문에서 제안하는 파라메터 값은 표 4.2의 값과 같이 고정하였으며 이 값은 ECC 표준에서 권장하는 160비트 이상의 사이즈로 1024비트 키 사이즈를 가지는 RSA 암호 알고리즘과 같은 보안강도를 가지게 된다. 다음 그림 4.1은 제안 프로토콜 구현을 위한 모듈도를 나타낸다.



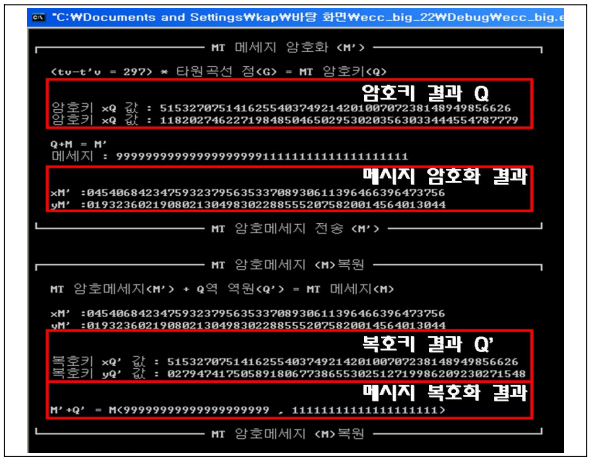
(그림 4.1) 제안 ECC 프로토콜 모듈도

그림 4.1 모듈도는 각 Control Unit에 의해 키 생성, 암·복호화 모듈과 인증 모듈을 구분하여 연동한다. 키 생성, 암·복호화 모듈은 ECC Point 연산을 하기위해 160비트 4칙 연산을 하는 모듈과 연동하며 복호화 수행에는 ECC Point Inverter 모듈을 이용해 복호키를 생성한다. Number Inverter는 ECC Point 연산을 효율적으로 하기 위한 유클리드 알고리즘을 이용한 정수의 역원을 구하는 모듈이고 Authentication 모듈은 복호된 메시지를 인증하는 모듈이다. 다음 그림 4.2는 제안 프로토콜의 공개키 생성 결과로 단계적 과정을 보여 주기위해 printf() 함수를 이용하여 콘솔화면에 출력하였다.



(그림 4.2) 제안 ECC 프로토콜 공개키 생성 결과

실제 Windows 프로그래밍에선 현 시스템의 시간 값이 초 단위 밖에 호출되지 않아 공개키 생성 모듈을 호출하는 순간의 타임 클럭을 1/1000초 시간 값으로 활용하였다. 이는 k가 절대 값을 취하는 난수이므로 tv-t'v 연산 결과가 일정한 범위를 넘지 않으면 그 의미를 잃지 않고 문제가 되지 않기 때문에 가능한 대처이다. 다음 그림 4.3은 암호키를 생성하고 메시지를 암·복호하는 결과를 보여 준다.



(그림 4.3) 제안 ECC 프로토콜 메시지 암·복호 결과

V. 결론

본 논문에서 제안하는 프로토콜은 효율성과 안정성이 입증되고 빠른 연산속도로 구현이 가능한 ECC 암호 알고리즘을 활용하였으며 한 번의 키 교환으로 일정기간 같은 패턴의 키를 교환하는 일반 암호 시스템과 달리 호출되는 시간 값에 따라 공개키를 갱신하여 암·복호키를 생성하는 난수를 쉽게 공유할 수 있도록 구성하였다. 만약 공격자에 의해 공개키의 도청이 있어도 일정 시간 값의 범위를 넘게 되면 공격을 감지할 수 있도록 하였고 실제 각 파라메터 값과 ECC 연산 방법, 프로토콜의 패턴을 알지 못할 경우 공격자는 인증 메시지를 알 수 없다. 또한 공격자가 데이터를 도청하여 다시 (SA)에 응답을 요청하여도 중복된 메시지는 사용하지 않으므로 응답을 받을 수 없고 (MT)의 데이터를 드롭시켜 정당한 사용자로 위장을 하더라도 원천적으로 160비트의 ECC 키를 해석할 수 없다.

따라서 본 논문에서 제안하는 프로토콜은 이동성을 제공하는 네트워크 단말인 PDA, 휴대폰, 노트북 등 보안에 취약성을 가지는 단말에 적용이 적합하며 공인인증서나 개인정보를 송·수신해야하는 시스템에 구성하여 공격자에 대한 대비를 철저히 할 수 있을 것으로 기대한다.

참고문헌

[1] 민병관, “암호화 기술의 최근 동향”, 전자부품 연구원, 2004. 02.

[2] 유영준, “디지털 암호화 기술 현황”, 전자부품 연구원, 2006. 11.

[3] SEC1, “Elliptic Curve Cryptography,” v.1.0, pp.62, Sept. 2000.

[4] 포항공대 “부가형 전자서명 방식 표준(안) - 제 3부: 타원곡선을 이용한 인증서 기반 전자서명 알고리즘”, TTA.KO, Dec. 2000.

[5] C. G. Pollman, “XML Pool Encryption,” XMLSEC02, USA, pp.1-9, 22, Nov. 2002.