공개키 기반 구조에서 ElGamal 방식의 ECC를 이용한 안전한 인스턴트 메시지 시스템 설계 및 구현

박수영* · 정채영**
조선대학교 컴퓨터통계학과
csssy@nate.com
cyjung@chosun.ac.kr

Design and Implement of Secure Instant Message System Using ECC of ElGamal Method on Public Key Infrastructure

Su-young Park* · Chang-Yeoung Jung**
Dept. Computer & Statistics. Chosun University

요 약

초고속인터넷이 널리 보급되면서 최근 메신저 서비스(Messenger Service)를 이용하는 사용자가 폭발적으로 증가하고, 해킹 기술의 발달로 인하여 메신저를 통하여 전달되는 메시지들이 악의의 사용자에게 쉽게 노출될 수 있는 가증서도 커지고 있다. 본 논문에서는 인스턴트 메신저의 안전한 통신을 위해 인증서를 이용한 인스턴트 메신저 프로토콜에 대해 설계하였다. 또한 메신저 서비스에서의 메시지 보안을 구현함에 있어서 공개키 암호 알고리즘의 연산 수행시간을 단축하기 위해 ElGamal 방식의 ECC(Elliptic Curve Cryptography) 알고리즘을 사용하고, 사용자 그룹 단위의 암호화를 위해 그룹별로 타원곡선과 그 위에 있는 임의의 점을 선택하여 다른 그룹과 구별하였다.

1. 서 론

현재 우리가 널리 사용하고 있는 메신저 프로그램들은 대다수가 암호화가 없이 메시지를 전송한다. 메신저를 통하여 중요한 메시지의 교환이 이루어지고 있으나 메시지의 암호화가 이루어지지 않아 메신저들은 많은 위험에 노출 되어있다. 해킹 기술의 발달과 함께, 악의의 사용자가 네트워크 상에 돌아다니는 메시지를 스니핑(sniffing)하여 들여다볼 수 있고, 이로 인하여 비밀 정보나 사생활 정보들이 다른 사람에게 공개 될 수 있는 위험이 내포되어 있는 것을 고려한다면, 사용자가 폭발적으로 늘고 있는 메신저에 대한 효율적인 메시지 보안에 관한 연구가 절실히 요구되어진다.

본 논문에서는 제안한 시스템에서는 Diffie-Hellman 키 비밀키 교환시스템 중에서 수행시간이 상대적으로 짧게 소요되는 ElGamal 방식으로 ECC를 구현한 암호화 방법을 사용하였고, 공개키 기반 구조(이하 PKI : Public Key Infrastructure)를 사용하여 사용자들이 많은 메신저 서비스에서 반드시 요구되어지는 키 관리의 용이함을 이루었고, 인증서를 통해 사용자의 공개키가 실제로 사용자의 것임을 증명하였다. ElGamal 방식으로 ECC를 구현한 암호화 방법을 사용하여 짧은 키 길이로 메신저 메시지의 암호화 및 복호화 수행 시간을 단축 시켰다.

2. 관련 연구

2.1.1 ICQ(I Seek You)메신저

ICQ메신저는 미라빌리스사의 제품으로 인스턴트 메신저의 원조이며, 그 사용자 수도 많다. ICQ메신저의 특징으로는 사용자가 대화 모드를 선택하여, 현재 자신의 상태를 다양하게 표시할 수 있으며, 대화 모드에는 온라인, 오프라인, 방해금지, 비공개 등으로 모드에 따라서 메시지 수신 방법 등의 차이가 있다[1].

2.1.2 MSN(Microsoft Network)

MSN메신저는 마이크로소프트사의 제품으로 메일 서비스 계정으로 메신저에 접속한다. MSN메신저는 접속한 사용자에게 실시간으로 메시지 전송을 할 수 있는데, 수신자가 도착된 메시지를 읽으면, MSN 인스턴트 메시지 창이 뜨게 된다. 인스턴트 메시지 창은 일대일 대화 기능처럼 두 사용자가 주고받는 메시지를 한 화면에 보여주고, 두 사용자간에 일대일 대화를 하는 동안 다른 사람을 초대하여 대화방 기능처럼 사용할 수 있다[2].

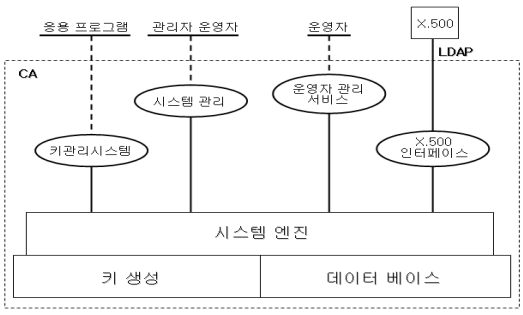
2.1.3 AOL(America Online)메신저

AOL메신저는 중앙의 BOS서버를 경유하여 한 사용자가 다른 사용자에게 HTML로 작성된 평문 메시지를 전송한다. AOL메신저의 특징으로는 직접적인 연결을 통하여 BOS서버를 경유한 AOL의 이미지 전송, 음성채팅, 게임요청, 파일공유 등이다. AOL메신저는 먼 거리에 있는 사용자에게 게임 프로그램의 실행을 요청할 수 있고, 요청 동안에는 어떠한 직접연결도 설정 될 수 없다.[3]..

3. PKI

3.1 PKI의 기본 구성도

X.500을 기반으로 한 PKI의 표준화가 IETF를 중심으로 이루어지고 있으며, 많은 RFC가 있다[4-7]. PKI의 기본 구성요소는 <그림 1>와 같으며 그 기능을 설명하면 다음과 같다.



(그림 1) PKI의 기본 시스템 구성도

3.2 시스템 관리

CA(Certification Authority)의 전체 시스템을 관리하기 위해 필요한 기능 즉 관리자의 인터페이스, 시스템 설치, 운영자 정보 관리, 데이터베이스 무결성 확인, 데이터베이스 백업 스케줄 결정, 예외적인 일의 발생시 회복 등과 같은 기능을 수행한다.

3.3 운영자 관리 서비스

운영자는 사용자 등록, 삭제, 변경 등을 담당하는 사람으로써, 운영자 관리 서비스 모듈은 이와 같은 기능을 수행한다. 일반적으로 이것을 지역등록기관이라고 한다.

3.4 키관리 서비스

응용프로그램(예: 하이브리드 메시징 시스템) 또는 다음 CA로부터의 인증서 요구, 암호화 키 생성 요청 등을 받고 수행하며 그 결과를 요청자에게 전송한다.

3.5 키 생성

CA 비밀키와 공개키, 사용자 암호화 키 등을 생성하는 역할을 담당한다. 이 부분은 하드웨어로 구성하는 것이 좋다.

3.6 데이터베이스

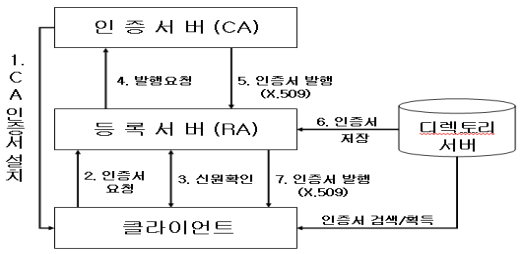
운영자와 관련된 정보, 관리자와 운영자가 수행할 수 있는 시스템 모듈과 영역을 정의한 privilege set, 키 history 등을 저장한다. CA 관리자 비밀키와 운영자 비밀키는 각각의 패스워드를 기반으로 시스템이 자동 생성한다.

3.7 시스템 엔진

시스템 관리, 운영자 관리 서비스, 키관리 서비스, 그리고 X.500 인터페이스 모듈은 시스템 엔진 위에서 시행되고, 시스템 엔진이 데이터베이스와 키 생성 모듈을 관리한다.

3.8 인증서 발급

<그림 2>는 인증기관을 이용한 인증서 발급절차를 보여주고 있다.



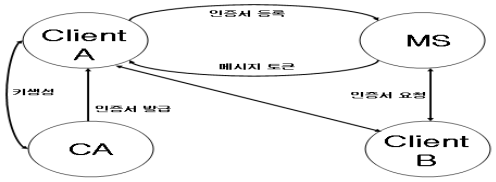
(그림 2) 인증서 발급절차

1. 인증서를 신청하는 곳에서는 우선 인증서버의 인증서를 설치해야 한다.
2. 클라이언트는 공개키와 인증서 발급요청서를 등록기관에 보낸다.
3. 접수한 인증 신청을 심사한다.
4. 신원확인에 문제가 없다면, 등록서버는 인증서버에 발행요청을 한다.
5. 인증서버는 공개키와 사용자정보를 이용하여 X.509 인증서를 만들어, 해당 인증서를 등록서버에 전달한다.
6. 등록서버는 모든 신뢰당사자가 이용할 수 있도록 인증기관의 저장소 또는 디렉토리 서버에 저장한다.
7. 등록서버는 클라이언트에게 인증서를 발급한다.
8. 발급된 인증서는 인증기관의 정책에 따라 관리된다.

4. 안전한 인스턴트 메신저

4.1 PKI를 이용한 키 생성 및 인증서 발급

본 논문에서 제안하는 인증 프로토콜의 전반적인 흐름은 <그림 3>과 같다.



(그림) 프로토콜 전체 개략도

※ 인증 프로토콜에 사용되는 기호

CA : 인증기관
MS : 메신저 서버
Client A : 사용자 A
Client B : 사용자 B
M : 메시지
ER : 공개키 암호알고리즘 암호
DR : 공개키 복호알고리즘 복호
Z : 압축 알고리즘
KUa : A의 공개키
KUb : B의 공개키
KUs : 서버의 공개키
KRa : A의 개인키
KRb : B의 개인키
KRs : 서버의 개인키
Ks : 세션키
H : 해쉬 알고리즘
Certificate A : A의 인증서
Certificate B : B의 인증서
PSE : 사용자의 정보를 저장하고 있는 국부 메모리

4.2 인증서 신청

CA와 Client A는 세션키를 안전한 방법(out of band)을 통하여 공유하고 있어야한다. Client A는 PSE를 생성하여 세션키로 암호화하여 인증기관에 전송한다.

ERKUKs { PSE } (식 4.1)

4.3 인증서 발행

CA는 세션키로 검증 후 인증서를 생성하여 Client A의 공개키로 암호화하여 전송한다.

ERKUa { Certificate A } (식 4.2)

4.4 신규등록

신규등록 기능은 사용자로부터 ID와 패스워드 등의 사용자 정보를 입력받아 신규 사용자 등록 작업을 수행한다.

ERKRa { H(PSE) } (식 4.3)

그리고 차후에 발생될 전송여부와 변조여부의 시비를 확인할 수 있도록 Hash된 문서(H(PSE))를 보관한다.

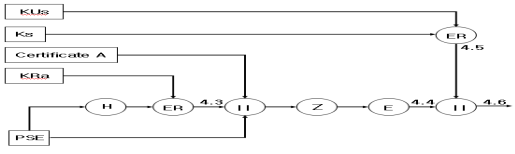
EKs{Z{M||ERKRa{H(PSE)||Certifacate A}} (식 4.4)

세션키(Ks)는 MS의 공개키(KUs)로 암호화 한 후

ERKUs(Ks) (식 4.5)

세션키로 암호화한 문서와 같이 MS에게 전송한다.

EKs{Z{M||ERKRa{H(PSE)||Certifacate A}} || ERKUs (Ks) (식 4.6)



(그림 4) 등록 전송전의 암호화 방법

4.5 등록 접수

그림 5는 MS가 사용자로부터 암호화된 정보를 받아 개인키를 이용하여 해독하는 부분이며 수행과정은 다음과 같다.

MS는 Client A에게서 받은 문서 중 ERKUs(Ks)를 자신의 개인키(KRs)를 사용하여 세션키(Ks)를 구한다.

DRKRs{ ERKUs(Ks)} = Ks (식 4.7)

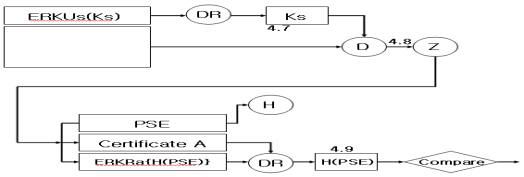
구한 세션키(Ks)를 사용하여 관용키 암호화된 부분을 복호화 한다.

DKs{EKs{Z{PSE||ERKRa{PSE}||Certificate A}}(식 4.8)

Client a의 개인키(KRa)로 암호화 된 PSE를 Certificate A에 포함되어 있는 A의 공개키를 사용하여 복호화 한다.

DRKUa{ ERKRa{ H(PSE) }} = H(PSE) (식 4.9)

압축을 풀어낸 문서에 포함된 PSE를 Hash하고 식 4.9에서 나온 H(PSE)과 비교하여 다르면 재전송을 요구하고 같으면 전송도중에 변조되지 않은 것으로 인정한다.



(그림 5) 등록 수신후의 복호화 방법

4.5 ElGamal 방식의 ECC를 이용한 암호화된 대화 기능

사용자 A가 B에게 채팅을 요청하여 상호간의 메시지 송·수신 중에 암호화된 메시지를 전송하는 기능을 수행한다. 인증서에는 각자의 공개키가 들어 있으므로 인증서를 서버로부터 획득하여 서로의 공개키를 확인한 다음 이 공개키를 이용해서 필요한 데이터를 암호화해서 보낸다. 이때 제약사항으로는 채팅하는 사용자가 인증서가 등록된 그룹멤버여야 한다.

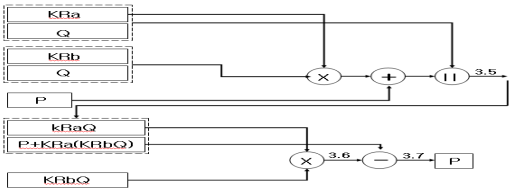
Client A와 Client B는 사용할 타원곡선 E와 타원곡선위의 임의의 점Q를 결정한다. Client B는 자신이 암호문을 전달받기 위하여 자신의 개인키를 이용하여 KRbQ를 계산하여 공개한다. 그리고 Client A는 자신의 개인키를 이용하여 KRaQ를 계산하고, 평문 P의 암호문으로 순서쌍 (KRaQ, P+KRa(KRbQ))를 Client B에게 보낸다.

{ KRaQ || P+KRa(KRbQ) } (식 3.5)

또한, Client B는 KRaQ에 자신의 개인키 KRb를 곱하여 KRa(kRbQ)를 구하고 이를 이용하여 P + KRa(KRbQ) - KRb(KRaQ)를 구하여 평문 P를 얻는다.

{ KRaQ x KRb } = { KRa(KRbQ) } (식 3.6)

{ P + KRa(KRbQ) - KRb(KRaQ) } = P (식 3.7)



(그림 6) 암호화된 대화 기능

ElGamal 방식을 이용하여 암호화 및 복호화를 할 때에는 덧셈연산 및 뺄셈 연산만을 수행하기 때문에 메시지가 증가함에 따라 수행시간이 비례적으로 증가하게 된다.

5. 결 론

본 논문에서는 Sever와 Client 환경에서 PKI를 이용한 사용자 인증을 소개하고, 서버로부터 획득한 인증서에 공개키를 사용하여 ElGamal 방식의 타원곡선암호(ECC)알고리즘을 이용한 메신저 메시지 보안시스템을 소개하였다.

인스턴트 메신저에서 사용자는 메신저에 등록하기 위하여 개인 신상에 관한 정보 및 인증서를 서버에 전송한다. 사용자는 서버에 접속하여 상대방의 인증서를 요청하고 유효한 인증서인지 확인한다. 유효한 인증서이면 타원곡선 E와 타원곡선 위의 임의의 점Q를 선택하여 암호화 통신을 시작한다. 타원곡선 암호 알고리즘을 이용함으로써 공개키 암호 방식의 단점인 연산처리 속도를 향상시킬 수 있다.

참고문헌

[1] ICQ, http://www.icq.com
[2] MSN, http://www.dreamseurity.com/products
[3] AOL, http://www.aim.com
[4] R. Housley, W. Ford, W. Polk., D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC2495, January 1999
[5] R. Housley, W. Polk, Representation of Key Exchange Algorithm(KEA) Keys in Internet X.509 Public Infrastructure Certificates, RFC 2528, March 1999
[6] S. Boeyen, T. Howes, P. Richard, Internet X.509 Public Key Infrastructure Operational Protocols-LDAPv2, RFC 2559, April 1999
[7] C. Adams S. Farrell, Internet X.509 Public Key Infrastructure Certificate Management Protocols, RFC 2510, March 1999