모바일 단말기 상에서 안전한 인증을 위한 자바 기반의 PKI 시스템 연구

최병선*, 채철주*, 이재광*
*한남대학교 컴퓨터공학과
e-mail:{bschoi, cjchae, jklee}@netwk.hannam.ac.kr

A Study of Java-based PKI System for Secure Authentication on Mobile Devices

Byeong-Seon Choi*, Cheol-Joo, Chae*, Jae-Kwang, Lee*
*Dept of Computer Engineering, Hannam University

요 약

모바일 네트워크 환경은 언제 어디서나 네트워크를 사용하는 모바일 서비스를 편리하게 사용할 수 있도록 해준다. 그러나 언제 어디서나 서비스를 제공받을 수 있다는 것은 언제 어디서든지 정보가 누출되거나 왜곡될 위험성 또한 존재하기 마련이다. 특히, 프라이버시 문제가 해결되지 않고서는 우리 일상생활과 융합되어 편리함을 제공해주는 모바일 네트워크 환경이 오히려 모바일 네트워크 감시 체제를 구축하는 심각한 역기능을 초래하게 될 것이다. 모바일 단말기들은 크기와 모양이 다양하고 컴퓨팅 연산 능력이 적은 저성능 휴대 장치들이 많기 때문에, 컴퓨팅 연산이 많이 요구되는 공개키 암호 기술을 저성능 모바일 단말기에 적용하기는 힘든 상황이다. 이에 본 논문에서는 프라이버시 문제를 해결하면서, 컴퓨팅 연산 능력이 적은 저성능 모바일 단말기에 적용할 수 있는 자바 기반의 암호 모듈 및 PKI 기반의 사용자 인증을 제안하고자 한다. 국내 표준 암호 알고리즘(SEED)과 인증서를 기반으로 세션키와 공개키를 조합함으로서 최소한의 암복호화 연산을 통해 인증 및 전자 서명을 제공하며, 이를 대표적인 모바일 단말기인 PDA 환경에서 세션키 분배 및 사용자 인증이 안전하게 이루어짐을 확인할 수 있었다.

1. 서론

전 세계적인 인터넷의 폭발적인 발전과 보급으로 인하여 네트워크 상에서 이루어지는 전자 거래 또는 전자 결제, 전자정부 등의 사이버 생활 범위가 넓어지고 있다. 특히 무선 네트워크를 사용하는 모바일 네트워크 환경은 모바일 단말기의 눈부신 성능 향상과 더불어 무선 대역폭의 증가로 이전에는 생각하지 못했던 모바일 정보 서비스를 제공받을 수 있게 되었다. 모바일 단말기란 액정화면과 메모리 처리기를 가지고 있으며 손에 들고 다니면서 이용할 수 있는 단말기를 통칭한다. PDA, HPC, 스마트폰, 핸드폰 등 이러한 제품을 모바일 단말기라고 하고 좀더 넓게 보면 타블렛 PC나 포스트 PC를 모바일 단말기로 볼 수 있을 것이다. 하지만 현재까지 개발된 네트워크 관련 기술들은 모두 데스크 탑 PC 상에서 수 Mbps ~ 수백 Mbps의 넓은 대역폭을 가지며 일반적으로 신뢰성 있는 유선 네트워크를 기반으로 서비스 개발이 이루어져 왔다. 이에 무선 네트워크는 유선 네트워크에 비교할 때, 훨씬 제한적인 통신환경을 가지며 사용 대역과 이동성 등의 제한으로 인하여 낮은 대역폭과 낮은 연결 안정성 및 가능성, 높은 지연 특성을 가진다. 또한 무선 네트워크를 사용하는 모바일 단말기는 데스크 탑 PC에 비하여 CPU 성능의 제한과 낮은 메모리 용량, 사용전력 제한 등의 특성을 가지게 된다. 이로 인하여 유선 네트워크 환경과 비교할 때, 많은 보안 취약점을 가지고 있다. 즉, 기존의 유선 네트워크와 데스크 탑 PC 환경에서 제공하는 보안 서비스를 쉽게 적용할 수 없는 문제를 가지고 있다는 점이다[1]. 이에 본 논문에서는 모바일 단말기에 인증서를 사용하여 세션키와 공개키의 조합으로 효율적인 인증을 제공할 수

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2006-C1090-0603-0027)

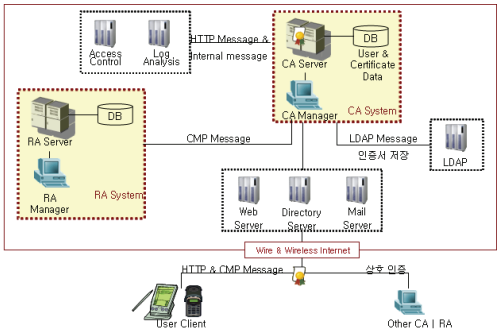
있는 PKI[2] 기반의 인증 시스템을 연구하였다. 인증서는 개인 또는 기관에서 서명 및 암·복호화에 사용되는 공개키와 이에 대응한 개인키의 소유 증명을 확인해 주는 전자문서로써 각 구성원의 신원을 증명하기 위한 중요한 수단이다[3]. 더불어 국내 환경을 고려하여 국내 표준 암호 알고리즘인 SEED와 대표적인 공개키 암호 알고리즘인 RSA, 해시 알고리즘인 SHA-1을 자바 기반의 암호 API로 구현하였다.

2. 모바일 단말기를 위한 PKI 시스템 설계

2.1 PKI 시스템

(1) 구성 및 기능

본 논문에서 연구한 모바일 단말기를 위한 PKI 시스템은 [그림 1]과 같은 구성을 가지고 있다.

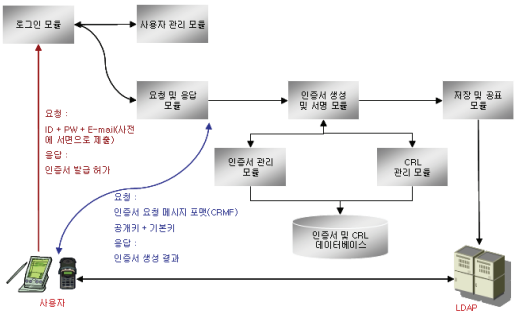


(그림 1) 모바일 단말기를 위한 PKI 시스템 구조도

- CA Server : 인증서를 발급하고 관리하는 인증 시스템의 핵심으로 다양한 공개키 알고리즘을 통하여 사용자 인증서를 발급한다.
- RA Server : 사용자 정보를 등록하고 관리하는 시스템으로 CA 서버와 연계하여 사용자 인증서에 대한 발급 업무를 보조해준다.
- LDAP : CA 서버에서 발급한 인증서를 공표하기 위한 시스템으로 공개저장소의 개념을 포함한다.
- Admin Tool : CA 서버에 대한 운영 및 관리를 위한 관리자 전용 도구로 CA 서버에 대한 전반적인 운영을 제어할 수 있다.

(2) CA 서버의 구성

본 논문에서 연구한 PKI 시스템에서 CA 서버가 가지는 역할은 절대적이라고 할 수 있다. CA 서버는 PKI 시스템의 근본을 이루는 사용자 인증서를 발행하기 위한 서버로, 이를 위해서 다양한 구성 요소를 가지게 된다. CA 서버의 역할 및 기능 구성은 [그림 2]와 같다.



(그림 2) CA 시스템의 모듈 구조도

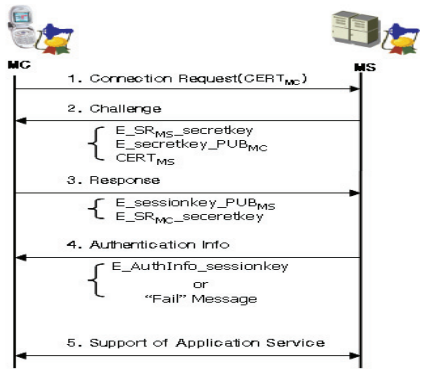
2.2 제안하는 모바일 단말기를 위한 인증 시스템

본 논문에서 제안하는 인증시스템은 자체 구축한 CA에서 발급받은 인증서를 PDA와 같은 휴대단말기에 저장하고, 이를 세션키와 조합하여 안전한 로그인 기능을 제공한다. 인증서에서 사용하는 공개키 암호알고리즘은 RSA(512 비트의 키)를 사용하며, 세션키는 SEED(128 비트의 키) 암호 알고리즘을 사용하여 생성한다. 다음 <표 1>은 본 논문에서 제안한 인증과정을 설계하는데 사용한 표기들이다.

<표 1> 사용자 인증 프로토콜 표기

표기	의미
E	암호화(Encryption)
D	복호화(Decryption)
CA	인증기관(Certificate Authority)
MC	모바일 클라이언트, 휴대단말기 사용자
MS	모바일 서버, 인증서버 및 서비스제공자
$CERT_{MC}$	모바일 클라이언트 인증서
$CERT_{MS}$	모바일 서버 인증서
SR_{MC}	MC가 생성한 난수(Secure Random 값)
SR_{MS}	MS가 생성한 난수(Secure Random 값)
PRI_{MC}, PUB_{MC}	모바일 클라이언트의 개인키와 공개키
PRI_{MS}, PUB_{MS}	모바일 서버의의 개인키와 공개키
secretkey	비밀키(Secret Key)
sessionkey	세션키(Session Key)
AutoInfo	인증이 성공했음을 포함하는 인증 메시지
SEED	국내 표준 대칭 암호 알고리즘(SEED)
RSA	비대칭 암호 알고리즘(RSA)
SHA-1	해시 알고리즘(Secure Hash Algorithm)
‖	연접(concatenate) 연산자

모바일 네트워크에서 안전하게 정보를 전송하기 위해서는 먼저 모바일 클라이언트(MC)와 모바일 서버(MS) 또는 기기간 상호인증 및 키 동기화(세션키) 과정이 필요하다. 그림 3은 이러한 과정을 보여주고 있다.



(그림 3) MC와 MS의 인증절차

(1) Connection Request와 Challenge

Step 1. [MC -> MS] : 모바일 클라이언트(Mobile Client)가 모바일 서버(Mobile Server)로 접속을 요청한다. 이때 클라이언트는 자신의 인증서($CERT_{MC}$)를 서버로 전송한다.

Setp 2. [MC <- MS] : 서버가 $E_SR_{MS}_secretkey$와 $E\_secret\_PUB_{MC}$, $CERT_{MS}$를 클라이언트로 전송한다.

① 클라이언트의 연결요청과 인증서를 수신한 서버는 CA에게 인증서의 유효성 유무를 요청한다.

② 클라이언트의 인증서가 유효한 경우, 서버는 SecureRandom 함수를 이용하여 난수(SR_{MS})를 생성하고, 이를 SEED 기반의 비밀키(secretkey)를 사용하여 암호화 한다($E\_SRMS\_secretkey$).

③ secretkey를 안전하게 전송하기 위해, $CERT_{MC}$로부터 획득한 RSA 기반의 공개키(PUB_{MC})를 사용하여 암호화 한다($E\_secretkey\_PUB_{MC}$).

④ 서버는 자신의 인증서($CERT_{MS}$)를 클라이언트로 함께 전송한다.

(2) Response

Step 3. [MC -> MS] : 클라이언트가 $E_sessionkey_PUB_{MS}$와 $E_SR_{MC}_secretkey$를 서버로 전송한다.

① 서버로부터 인증서를 수신한 클라이언트는 CA에게 인증서의 유효성 유무를 요청한다.

② 서버의 인증서가 유효한 경우, 클라이언트는 자신의 개인키(PRI_{MC})를 사용하여 서버로부터 수신한 secretkey를 복호화한다($D\_secretkey\_PRI_{MC}$).

③ 복호화한 secretkey를 사용하여 SR_{MS}를 복호화한다($D\_SRMS\_secretkey$).

④ 클라이언트는 자신의 난수(SR_{MC})를 생성하고, 이를 secretkey로 암호화한다($E\_SR_{MC}\_secretkey$).

⑤ 클라이언트는 자신이 생성한 SR_{MC}와 복호화한 SR_{MS}를 연접($SR_{MS} \| SR_{MC}$)하여 16바이트(SEED는 128비트의 키를 사용)의 세션키(sessionkey)를 생성하고 이를 $CERT_{MS}$로부터 획득한 공개키(PUB_{MS})를 사용하여 암호화한다($E\_sessionkey\_PUB_{MS}$).

(3) Authentication Info

Step 4. [MC <- MS] : 서버가 인증 유무를 알려주는 E_AuthInfo_sessionkey 또는 Fail 메시지를 클라이언트로 전송한다.

① secretkey를 사용하여 SR_{MC}를 복호화한다. ($D\_SR_{MC}\_secretkey$).

② 복호화한 SR_{MC}와 서버 자신의 SR_{MS}를 연접하여, 16바이트 sessionkey를 생성한다($SR_{MS} \| SR_{MC}$).

③ 서버의 개인키(PRI_{MS})를 사용하여 클라이언트로부터 수신한 sessionkey를 복호화한다($D\_sessionkey\_PRI_{MS}$).

④ 서버는 Step 4-②에서 생성한 sessionkey와 Step 4-③에서 복호화한 sessionkey를 비교한다. 만약, 두 키가 일치한다면 sessionkey를 동기화하고 클라이언트에게 인증이 성공했음을 알리는 메시지를 동기화한 sessionkey를 사용하여 암호화한다(E_AuthInfo_sessionkey). 그러나 두 키가 일치하지 않는다면, 인증이 실패했음을 알리는 메시지를 생성한다(Fail).

(4) Support of Application Service

Step 5. [MC <-> MS] : 인증이 성공한 경우 서버에게 응용 서비스를 제공받을 수 있다.

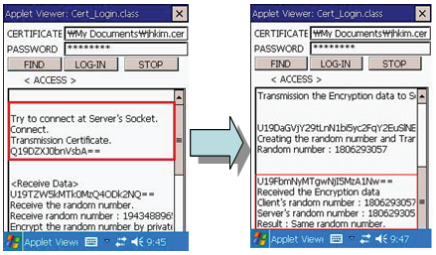
3. 구현 및 성능 평가

본 논문에서 개발한 CA 및 보안 서비스, LDAP 등과 같은 전체 시스템은 MS사의 Windows2003 Server와 RedHat사의 리눅스9을 이용하여 시스템을 구축하였다. 또한, JDK1.4[4]를 통하여 각종 API 및 인터페이스 등을 구현하였으며, 보안서비스와 관련된 모듈은 Bouncycastle[5]의 핵심 API를 활용하여 구축하였으며, SEED를 추가하였다. 다만 PDA의 자바환경은 Pesonal Java 1.1을 사용하였다. 가장 중요한 특징 중의 하나는 현재 개발한 PKI 시스템은 PDA 등과 무선 단말기 환경을 고려하여 개발하였으나, 일반 유선 데스크탑 환경으로 확장이 용이하다는 점이다.

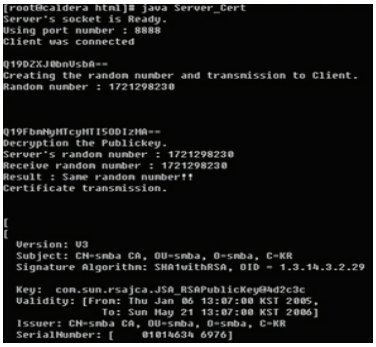
3.1 인증서 기반의 인증 테스트

본 논문에서 인증서 기반의 인증 구조는 [그림

4]와 [그림 5]에서 보는 바와 같이 비밀키와 공개키를 조합한 인증 시스템으로 공격자가 중간에 메시지를 가로챌 수 없으며, 이 과정을 통하여 안전하게 메시지를 송수신할 수 있는 세션키를 확립한다. 세션키는 인증 이후, 서버가 클라이언트에게 서비스를 제공하는 경우, 민감한 데이터(예를 들어, 뱅킹 또는 전자거래 정보)에 대하여 암호화를 수행할 수 있다.



(그림 4) 인증 과정(클라이언트)



(그림 5) 인증 과정(서버)

또한 다음 [그림 6]에서 보는 바와 같이 서버가 클라이언트를 인증하게 되면, 클라이언트에게 인증 완료 메시지를 전송하고, 클라이언트의 응용 서비스 요청을 기다린다.

(그림 6) 클라이언트 인증 성공

3.2 성능 평가 및 분석

암호 알고리즘의 성능을 평가하기 위해, 다음 [표 2]와 같이 동일한 암호 알고리즘에서 중요한 연산들이 속도를 PDA와 데스트 탑 PC 환경에서 테스트 하였다. 수행 시간은 동일한 연산을 1000회 반복한 시간이며, 단위는 ms(1/1000초)이다. 처리 가능한 최대 자료형이 JDK1.4 (PC)에서는 8 바이트 길이의 Long형이고, Personal Java 1.1(PDA)에서는 2바이트 길이의 short형이다. 이러한 제한 때문에 본 논문에서 구현한 연산 클래스가 JDK1.4보다 소요 시간이 오래 걸린다.

<표 2> 수행시간(단위 : ms)

Method	PDA	PC
최대 공약수 x.gcd(y)	221	20
모듈러 지수 연산(x^y mod n) x.modPow(y, n)	130	71
승산 역원 (x^{-1} mod n) x.modInverse(n)	140	50

4. 결론 및 향후연구

본 논문에서는 모바일 단말기가 가지는 하드웨어적 제약 사항을 극복하고자, 모바일 단말기에 적합한 인증서 관리 서비스와 자바 기반의 암호화 알고리즘을 설계 및 구현하였다. 또한 비밀키와 공개키를 조합하여 보다 안전한 인증을 제공할 수 있도록 하였으며, 세션키와 공개키의 조합을 통하여 안전한 인증 기능을 제공할 수 있었다. 또한 국내 표준 암호 알고리즘인 SEED를 모바일 단말기에 적용하여 테스트하여 보았다. 또한, 암호 알고리즘과 PKI 시스템 구축을 위한 핵심 클래스의 결합을 통해서 인증서를 발행 및 관리를 자유롭게 할 수 있는 PKI 체계를 구축하였다. 본 논문에서 개발된 PKI 시스템은 기존 사용 제품들에 대항하여 독자적인 기술/상업적인 가치를 지니고 있으며, 향후 각종 보안 서비스의 제공에 있어서 선도적인 위치를 차지할 수 있을 것이다. 또한, 자바 기반의 PKI 시스템은 이식성이 매우 높으며, 개별 서비스에 대한 모듈 형식으로 구성되어 있어, 그 활용의 범위가 고정되지 않고, 다양한 시스템 및 서비스에 적용할 수 있는 장점을 가지고 있다. 향후 연구로 이러한 보안 모듈을 WIPI 기반의 환경에 적용시켜 보고자 한다.

참고문헌

[1] 윤종호, “무선 LAN 보안 프로토콜”, (주)교학사, 2005년

[2] 전문석 외 6명, “PKI”, 도서출판 미래컴, 2003년

[3] RSA Data Security, Inc., "Public Key Cryptography Standards #1-12", June 3, 1991

[4] http://java.sun.com

[5] Jess Garms, "Professional Java Security", 정보문화사, 2001