PKI 기반의 암호화 통신 컴포넌트 설계 및 구현

모수종* · 조원희* · 유선영* · 임재홍*

Design and Implementation of PKI based Cryptography Communication Component

Soo-jong Mo* · Won-hi Cho* · Sun-young Yu* · Jae-hong Yim*

본 연구는 산업자원부의 지역혁신 인력양성 사업의 연구결과로 수행되었음.

요 약

인터넷을 통한 전자상거래와 전자서명기술이 대두되면서 부각된 기술이 PKI(Public Key Ifrastructure)이다. PKI는 암호화기반기술에 있어서 여러 가지 새로운 표준들을 가져왔다. 그러나 이 같은 표준화가 활발히 진행되고 있음에도 PKI를 응용한 솔루션들은 가격이 비싸고 느린 단점들이 지적되고 있다. 현재 PKI를 포함한 암호화기술의 주된 관심이 빠른 속도와 보다 향상된 보안성이라고 한다면 이 같은 문제들은 심각한 문제들이라고 할 수 있다. 본 논문은 기존의 PKI구조에서 가지는 몇 가지 문제를 개선하여 짧은 메시지 통신이나 단순 방식 암호화 통신에 단순하고 쉽게 사용될 수 있는 통신 컴포넌트를 설계하고 구현하였다.

ABSTRACT

Specially, though electronic commerce and electron signature technology through internet rose, PKI (Public Key Infrastructure) is one of technologies. PKI brought several kind of new standards in encryption base technology. In spite of several kind of standardizations consist lively, shortcoming of solutions that apply PKI is expensive and slow. If main interest of encryption technology including PKI is the fast speed and security that improve, this is very serious problem. The various kinds alternatives about these problem are presented. But, we must consider about replace expense and stability etc. still. So, I propose that use suitable encryption policy by method to solve such problem. I improved some problems of existent PKI structure. Subject of this treatise designs and embody communication component could use easily and simply short message communication or simplicity way encryption communication.

키워드

PKI, 암호화, 보안

Ⅰ. 서 론

최근에 들어서 인터넷과 네트워크 환경이 비약적으로 발전함으로서 우리의 생활 패턴이 바뀌고 다양한 형태의 요구와 그에 따른 여러 가지 기술들이 개발되고 있다. 그런 일련의 기술들 중에는 암호화 기술도 중요한 부분을 차지하고 있는데, 특히 인터넷을 통한 전자상거래 환경의 수요가 증가함에 따라 지불과 결제

* 한국해양대학교 전자통신공학과

접수일자 : 2005. 5. 25

방식에서 신뢰성과 보안성이 강력히 요구된다. 전자상거래 뿐만 아니라 부인 방지를 위한 전자 서명 분야에도 중요하게 암호화 기술이 필요하다. 이러한 요구에 따라 급격히 부상하고 있는 암호화 기술이 인증기관을 통해 인증이 보장되는 공개키 방식인 PKI(Public Key Infrastructure) 기술이다[1].

PKI는 전자상거래의 발전과 사이버 공간의 신원보증, 사업의 신뢰성을 확보하기 위한 보안 수단으로서 없어서는 안 될 중요한 부분이다. 특히 전자서명이나 암호화가 필요한 각종 응용프로그램과의 통합 솔루션으로 그 활용도가 넓어지면서 중요성이 더해지고 있다. 국내 암호화관련 산업은 통합 인증 및 권한관리, 전자세금계산서 또는 전자계약, 응용 솔루션 분야의 기술 개발과 관련 특허 연구도 활발히 이루어지고 있다[2][3].

국내 공인 인증기관에서 발행하는 코드 서명 인증서를 사용하여 코드 서명을 시행하고 사용자들은 인터넷 익스플로러를 통해 다운로드할 수 있도록 하기 위해서는 MS의 인증 정책에 의거해야 하는 문제가 있으며, 기존 PKI 기반 솔루션들의 단점은 저속도와 고비용, 그리고 높은 사양의 하드웨어 성능을 요구하는 것이었다.[4]

본 논문에서는 이러한 단점들을 개선하기 위해 PKI를 기반으로 하는 환경을 구축하고 개선된 보안 정책의 제공을 통해 보안성이 제공되는 개선된 형태의 메시지 통신이 가능한 보안통신 컴포넌트의 설계 및 구현하였다. 구현된 통신 컴포넌트 소규모의 메시지 통신에 적응성이 뛰어나고 적절하게 조정된 형태를 가지는 암호화 정책의 설계를 통해 구현하였고 범용성이 뛰어난 이중으로 보안이 제공되는 성능을 가지도록 하였다.

Ⅱ. 보안 및 인증 기술

2.1 PKI

PKI는 공개키 인증문제를 해결하여 정보의 기밀성, 접근제어, 무결성, 인증 및 부인봉쇄를 제공하는 정보보호 기반구조이다. PKI는 공개키에 대한 인증서를 발급하는 인증기관과 사용자에게 인증서 발급 요청을 등록하고 신원 확인 기능을 수행하는 등록기관과 사용자에게 인증서 발급 요청을 등록한다. 신원 확인기능을 수행하는 등록기관 그리고 인터넷 상의 다양한 사용자와 응용이 인증기관에서 발급한 인증서를 쉽게 검색할 수 있도록 인증서를 관리하는 디렉토리(directory) 서버로 구성된다. 다양한 응용에서 공개키를 이용하여 전자서명을 생성하고 검증하며 데이터에 대한 암·복호화를 수행할 수 있는 보안 툴킷(tool kit)을 제공한다. 또한 인증서 발급 정책과 관리 정책 등을 포함하고 각각의 시스템 컴포넌트 간의 통신 프로토콜을 정의한다. 그림 1은 공개키 기반의 정보보호 서비스를 제공하기 위한 PKI기반 시스템의 구성을 나타내고 있다.

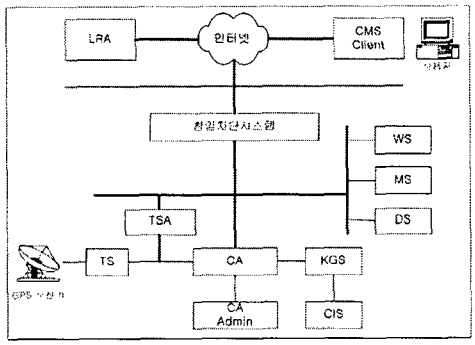


그림 1. PKI 시스템의 구성
Fig. 1 Constitution of PKI system

PKI의 하부 시스템인 CIS(Card Issuing System)는 인증서의 저장매체인 IC 카드 발급 기능을 수행하는 시스템이며 KGS(Key Generation System)는 PKI를 구성하는 CA(Certification Authority)와 운영자의 키 쌍을 생성하는 키 생성시스템이다. 인증기관은 CA와 CA Admin으로 구성된다. CA는 인증서 발급, 갱신, 정지, 폐지 등과 같은 인증서 라이프 사이클을 관리하고 인증기관, 등록기관, 사용자에 대한 정보를 관리한다. CA Admin은 CA에 대한 관리기능을 수행한다. LRA (Local Registration Authority)는 사용자의 인증서 발급 요청 접수를 대행하며 사용자의 인원 확인을 수행하는 등록관리 시스템이다.

CMS(Certificate Management System) 클라이언트는 사용자가 자신의 공개키 쌍을 생성하고 인증서 발급을 CA에 요청하고 발급된 인증서를 관리하는 기능을 제공한다. WS는 웹을 통해 CA에 접근할 수 있도록

하는 웹 서버이며 MA는 메일 관리를 수행한다. DS는 인증기관에서 발급한 인증서와 인증서 취소 목록을 게시하여 인터넷상에서 사용자와 응용프로그램들이 이를 접근할 수 있도록 한다. TSA(Time Stamping Authority)는 인증서로 보호되는 트랜잭션에 타임스탬프(time stamp)를 제공하고, TS(Time Server)는 GPS (Global Positioning System)를 통해 국제 표준시를 제공하는 시스템이다[5][6].

2.2 인증 기술

인증서는 사용자의 신분과 공개키를 연결해 주는 문서로 인증기관의 비밀키로 전자 서명을 수행하여 생성된다. 다시 말해 이것은 사용자의 공개키가 실제로 사용자의 것임을 증명한다. PKI에서 인증서의 발행 대상은 인증기관과 사용자, 서버 등으로 인증기관에게는 상위 인증기관이 인증기관의 적법성을 증명하기 위해 발행하고 사용자와 서버에게는 사용자의 신원, 서버 등의 적법성을 증명하기 위해 인증기관에서 발행한다. 인증서의 형식은 1988년에 ITU-T가 X.509 초기 버전을 공표하고 1993년에 버전 2를 공표했으며 1995년 이후로는 ISO/IEC 9594-8의 문서와 동일시되어 공동 개발되어 왔다. 현재는 X.509 v3까지 공표되었고 인증서의 확장 영역에 대한 개정이 진행되고 있다. X.509 v3의 인증서 형식은 그림 2와 같다. X.509v3가 갖는 항목들을 자세히 설명하면 아래와 같다.

- Version : X.509의 버전으로 0은 버전1, 1은 버전2, 2는 버전3을 의미한다.
- Serial Number : 발행자가 생성한 각각의 확인서에 대한 유일 식별자
- CA Signature Algorithm : 발행자가 확인서를 서명에 사용한 알고리즘
- Issuer Name : 확인서를 생성한 발행자의 식별자로 X.500 명명 방식을 따른다.
- Validity Period : 확인서가 사용될 수 있는 시작 시간과 끝 시간을 기입하는 것으로 시간과 날짜(UTCT 형식)로 표현된다.
- Subject Name : 확인서를 받는 공개키 소유주의 식별자로 X.500 명명 방식을 따른다.
- Subject Public Key Information : 사용자의 공개키와 공개키에 대한 정보(알고리즘과 파라미터)를 기입한다.
- Issuer Unique Identifier(optional) : 버전2 이상에서 사용되는 것으로 발행자의 부가적인 정보를 포함(선택)한다.
- Subject Unique Identifier(optional) : 버전2 이상에서 사용되는 것으로 사용자의 부가적인 정보를 포함(선택)한다.
- Extension Field(optional) : 인증 정책 등 여러 가지 사항을 포함(선택)한다.
- Issuer's Signature : 앞의 목록들에 대한 서명값이다.

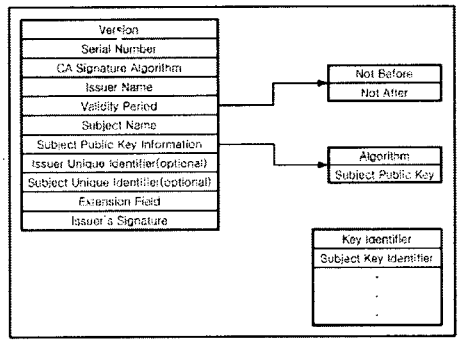


그림 2. X.509 v3 인증서 형식
Fig. 2 Form of X.509 certificate

Ⅲ. 암호화 통신 컴포넌트 설계

컴포넌트는 기본적으로 서버와 클라이언트 그리고 인증을 책임지는 인증기관 역할을 수행하는 CA로 이루어져 있다. 우선 클라이언트측은 전송 또는 수신할 데이터를 암·복호화 하기 위한 암호화 모듈과 인증서에 관련한 전체적인 작업을 관리함으로서 키 관리를 수행하는 키 관리 모듈과 서버 측과의 통신을 위한 통신모듈로 구성된다. 그리고 서버측은 클라이언트와 같은 데이터의 암·복호화를 위한 암호화 모듈과 데이터 전송을 위한 통신모듈이 포함된다. 클라이언트와는 다소 다른 키 배포와 관리를 맡은 키 관리모듈과 디렉토리 서버라고 불리는 인증서 저장 및 CRL 유지 같은 업무를 수행할 디렉토리 서버 모듈과 클라이언트의 작업을 위해 접근 권한 등을 직접적으로 적용할 DB 관련모듈을 포함한다.

그림 3은 전체 컴포넌트의 구성을 나타낸다. 전체

컴포넌트의 주된 기능은 MS사에서 API 형태로 제공되는 Cryptographic library를 이용하여 구현하였다[7][8].

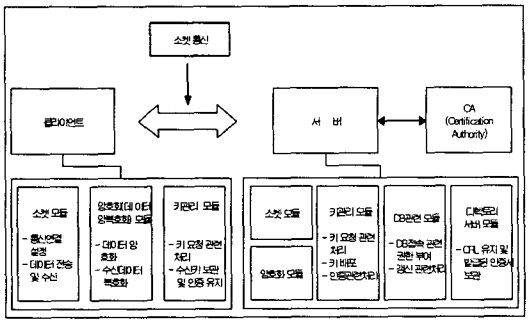


그림 3. 전체 컴포넌트의 구성
Fig. 3 Constitution of component

3.1 서버 컴포넌트의 설계

그림 4는 서버 컴포넌트 구성을 보여준다. 전체의 컴포넌트는 서버와 클라이언트 두 부분으로 구성되어 동작하도록 할 것이다. 이중 서버의 역할은 인증 요청시 CA에 이를 의뢰하여 인증을 발부한다. 인증 요청자에 대한 개인키를 X.509 인증서 형식에 포함시켜서 보내게 된다. 서버는 이 같은 일련의 과정을 키 관리 모듈을 두어 처리하도록 하며 인증에 대한 내역을 디렉토리 서버에 보관함으로서 인증에 대한 내역을 보관하도록 하여 인증에 대한 상태를 유지하도록 한다.

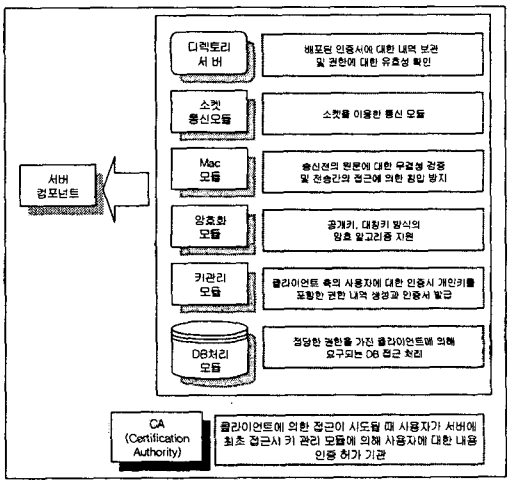


그림 4. 서버 컴포넌트의 구성
Fig. 4 Structure of server component

서버와 클라이언트 컴포넌트간의 통신에는 키 관리 모듈에서 생성된 세션키를 이용한 소켓통신을 사용하며 MAC 모듈에는 해쉬 함수를 사용하여 메시지 간의 무결성을 보장하도록 할 것이다. MAC 모듈은 통신 설정될 때마다 메시지에 대한 압축 검증코드를 생성하게 되어 메시지 전송 전에 송신하게 된다.

그림 5는 서버에서 키 관리 모듈의 동작 형태를 보여주고 있다. (a)는 서버 측에 최초 접속시 클라이언트 측의 사용자에 대한 신원을 CA에 의뢰해 확인하고 이에 대한 개인키와 공개키를 생성하며 개인키와 일정 시간동안 사용 가능한 세션키, 유효기간 등의 정보를 포함하는 인증서를 발급하여 클라이언트 측에 전달하게 된다. 이후의 통신 요청시 (b)와 같이 세션키의 확인과 그에 따르는 공개키를 매칭하는 역할도 수행하게 된다. 또한 키 관리 모듈은 CRL에 대한 정보 갱신에도 관련하여 동작을 하도록 하게 된다.

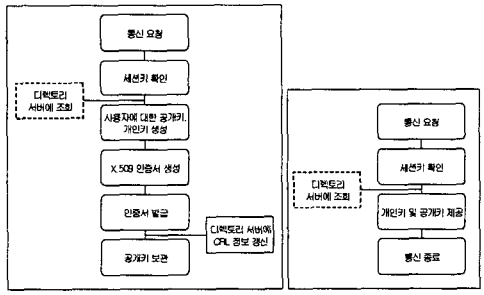


그림 5. 키 관리 모듈의 동작
Fig. 5 Execution of key management module

3.2 클라이언트 컴포넌트의 설계

클라이언트는 서버와의 통신에 접속하기 위해 필요한 기능을 구현할 수 있도록 설계하였다. 상호통신에 대한 보안성과 이에 대한 무결성 검증에 필요한 모듈들을 내장하도록 하였다. 클라이언트 컴포넌트는 서버로부터 발급받은 인증서를 보관하고 이에 대한 인증서 소유주가 접근시 패스워드를 사용한 클라이언트 차원의 인증과정을 제공하며 메시지 암호화에 사용되는 인증서에 포함된 개인키 및 세션키를 제공하는 키 관리 모듈과 개인키와 세션키를 사용해 메시지를 암·복호화 하는 암호화 모듈을 가지고 있다. 또한 서버에서와 같이 메시지에 대한 해쉬 함수를 이용한 코드를 생성하고 이에 대한 상호 해석을 통한 전송간의 메

시지에 대한 무결성을 제공하는 MAC 모듈과 상호 통신을 위한 소켓통신을 통해 통신 기능을 제공하는 소켓통신 모듈이 포함된다. 그림 6은 클라이언트 컴포넌트 구성을 나타낸다.

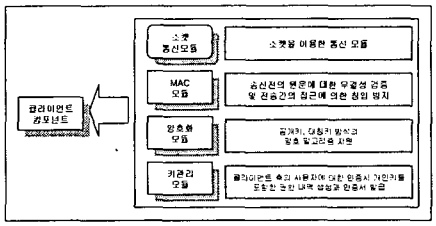


그림 6. 클라이언트 컴포넌트의 구성
Fig. 6 Structure of client component

클라이언트 측의 키 관리 모듈은 서버 측의 키 관리 모듈과 달리 공개키 및 개인키 생성기능을 필요로 하지 않으며 메모리공간에 예약되는 저장소를 이용한 인증서에 대한 보관과 통신 요청시 세션키의 생성, 그리고 통신 간에 메시지 암·복호화에 사용되는 공개키와 개인키 쌍에 대한 정보를 제공하는데 역할을 한다. 이 같은 키 관리 동작은 Cryptographic library내에 인증 체인이라고 불리는 동작으로서 수행된다. 인증서를 보관하는 디렉토리 서버나 보관소 내의 해당 인증서와의 내용을 비교하게 됨으로서 가능하게 되는 것인데 여기에 확장 필드들을 사용함으로서 일반적인 확장 기능을 수행할 수 있게 된다.

Ⅳ. 구현 및 고찰

본 논문의 실험 결과를 위해 암호화 통신 컴포넌트를 이용한 서버-클라이언트 형태의 간단한 암호화 통신 프로그램을 구현하였다. 암호화 기능을 제공하는 여러 프로그램 라이브러리들 중에 가장 쓰기 편리하고 많은 기능을 제공하는 MS Cryptographic library를 사용하여 구현을 하였다. 구현 환경은 Visual C++, ODBC(Open Database Connectivity)를 사용하였고 암호화에 사용된 알고리즘 방식들은 아래와 같고, 그림 7은 프로그램 구현 화면이다. Cryptographic library와 구현한 컴포넌트를 사용하기 위해 라이브러리를 추가하여 프로그램을 구성하였다.

- 키 교환 및 서명 알고리즘 : RSA(128 비트 키)
- 세션키 암호화 : RC4(40 비트 키)
- 암호화 알고리즘 : DES(56 비트 키)
- 해쉬 알고리즘 : MD5

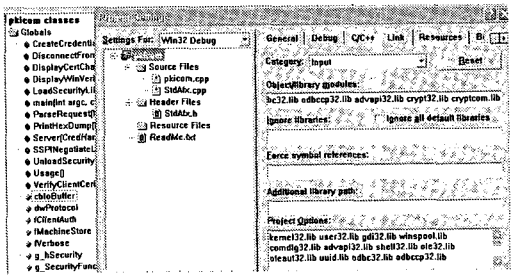


그림 7. 라이브러리 구성
Fig. 7 Structure of library

그림 8은 서버가 최초 소켓을 이용하여 클라이언트의 요청을 감지하고 클라이언트의 요청이 있을 경우 핸드쉐이크 메시지 교환을 통해 클라이언트 측의 사용자를 인증하고 이에 대한 클라이언트의 동작에 대해 권한 여부를 검사하고 동작을 수행해준 후 클라이언트의 종료 요청을 들어준 후 요청 감지 상태로 돌아가는 동작을 수행하는 과정을 나타내고 있다.

서버와 클라이언트가 같은 호스트에서 실행된바 클라이언트가 서버측이 위치한 호스트의 주소로 연결을 한 후 서버에 접속하기 위한 ID와 패스워드를 이용해 접속한 후 핸드쉐이킹 과정을 통해 인증서를 받은 뒤 파일을 요청해 이를 수신 메시지 복호화 및 무결성 검증을 거쳐 종료 요청을 수행하는 동작 과정을 그림 9가 보여주고 있다.

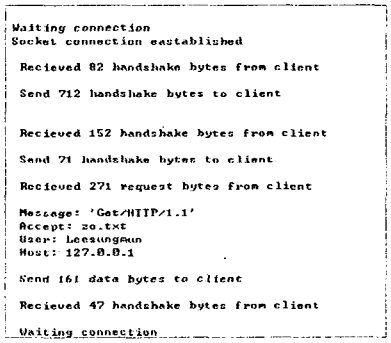


그림 8. 서버 프로그램 실행
Fig. 8 Execution of server program

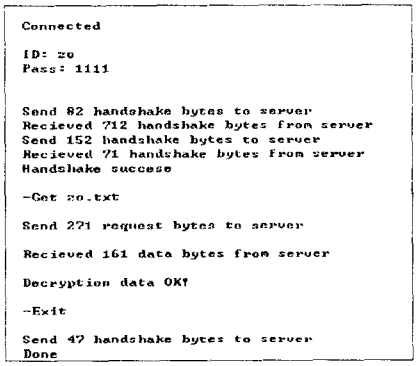


그림 9. 클라이언트 프로그램 실행
Fig. 9 Execution of client program

프로그램에 사용되는 인증서는 서버 측의 CA로부터 사전에 클라이언트에 발급된 간단한 형태를 취하였다.

그림 10은 실험에 사용된 인증서와 동일한 인증서로서 패스워드를 이용해 클라이언트의 보관소에 접근한 인증서 내용이다. 여기에는 사용자의 이름, ID, 유효기간과 표시되지 않는 공개키 및 시리얼넘버가 포함되어 있다.

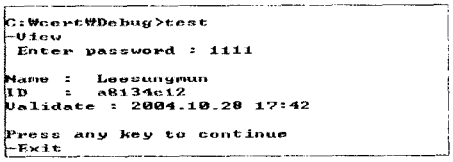
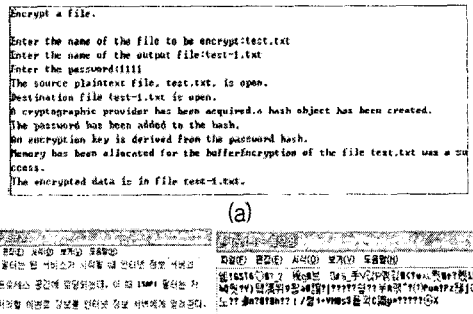


그림 10. 인증서 실행 테스트
Fig. 10 Certification execution Test



(a)

(b) (c)

그림 11. 파일 암호화
Fig. 11 File Encryption

실험상에서 암호화와 복호화가 두 번씩 실행되나 암호화의 보안 특성상 이에 대한 접근이 어려워 파일을 블록 암호화하는 실행 테스트를 하였다. 위 실험에서 사용된 40비트 키를 사용하는 RC4 방식의 알고리즘을 동일하게 사용하였으며 마찬가지로 위 실험에서 전송한 것과 동일한 내용의 한 개의 간단한 텍스트 파일을 이용하였다.

그림 11은 이에 실행 화면을 보여주고 있다. (a)가 프로그램의 실행 화면이며 이 과정은 첫 실험에서 사용한 세션키를 이용한 평문 암호화 과정과 동일하다. (b)가 실행 전의 평문 형태이며 (c)가 암호화된 문장을 보여주고 있다.

V. 결 론

본 논문에서는 우리 실생활 전반에 걸쳐 폭넓게 자리잡아가고 있는 PKI를 기반으로 하는 암호화 통신 컴포넌트를 구현하였다. 서버와 클라이언트 형태로 이루어져 있는 통신 형태에서 서버는 클라이언트의 사용자에 대한 정보를 포함하고 있는 인증기관으로부터 인증내역과 사용자에 대응하는 공개키를 분배받는다. 클라이언트 측의 사용자는 키 관리 모듈에 의해 관리되는 인증서를 통해 서버에 접근하여 정해진 권한 내에서 동작을 수행할 수 있도록 하였다. 또한 서버는 인증기관의 역할을 하는 사용자의 DB를 포함하여 사용자 개인 인증과 더불어 권한 등을 설정하도록 하며 역시 서버 측의 키 관리 모듈에 의해 배포된 인증서와 공개키에 대한 유효함을 관리한다.

인증서를 통한 인증과 이를 통한 암호화 통신 방식은 기존의 암호화 환경에 비해 훌륭한 성능의 환경을 제공하고 있다. 본 논문에서와 같이 인증서를 통한 확장한 형태의 기능인 권한부여 같은 동작의 수행은 아직도 여러 방법들로 상용화시키려는 노력이 계속되어지고 있다. 향후 ODBC(Role Based Access Control)에 연계된 형태의 개발과 인증기관에 대한 신뢰성 향상에 관한 부분에 대한 연구가 필요할 것이다.

참고문헌

[1] 홍기향, 김정덕, "ISO에서의 정보보호관리 국제 표준화 동향", 정보보호학회지 14권 2호, pp.6-12, 2004. 4

[2] 정보통신연구원, "정보통신 표준화백서", pp.37 - 387, 2003.

[3] 염홍열, "IETF 공개키 기반구조 및 PKI기반 응용 표준화동향", 정보보호학회지, pp.24-37, 2004. 4.

[4] 이래, 이동훈 "코드 서명 기술의 국내 PKI 적용 방안 비교 연구", 정보보호학회지, 14권 3호, 2004. 6

[5] 한국정보보호센터, "PKI 기술 규격안", pp. 17-41, 52-75, 2001. 5.

[6] 김덕기, "PKI 기반 보안 웹 서비스 제공 방안에 관한 연구", 한국정보처리학회 논문집 10권 2호, pp.1897-1900, 2003. 11.

[7] Michael Welschenbach, "Cryptography in C and C++", Apress, 2003.1.

[8] 조은애, "SSL 컴포넌트의 설계 및 구현", 한국정보과학회, pp.808-810, 2003.10

저자소개

모수종(Soo-Jong, Mo)

1996년 경상대학교 물리학과 졸업(이학사)
1999년 한국해양대학교 대학원 전자통신공학과(공학석사)
2004년～현재 한국해양대학교 대학원 전자통신공학과 박사과정
※ 관심분야 : 분산 네트워크 시스템, 통신프로토콜

조원희(Won-Hi, Jo)

2003년 한국해양대학교 대학원 전자통신공학과 졸업(공학사)
2005년 한국해양대학교 대학원 전자통신공학과(공학석사)
※ 관심분야 : 분산 네트워크 시스템, 통신프로토콜

유선영(Sun-Young, Yu)

1998년 한국해양대학교 전자통신공학과(공학사)
2002년 한국해양대학교 대학원 전자통신공학과(공학석사)
2005년 한국해양대학교 대학원 전자통신공학과(공학박사)
※ 관심분야 : XML, Semantic Web

임재홍(Jae-Hong, Yim)

1986년 서강대학교 전자공학과 졸업(공학사)
1988년 한양대학교 대학원 전자공학과(공학석사)
1995년 한양대학교 대학원 전자공학과(공학박사)
1995년 3월～현재 한국해양대학교 전파·정보통신공학부 부교수
※ 관심분야 : 분산 네트워크 시스템, 임베디드 시스템