웹기반 인증서 및 키관리 시스템의 설계 및 구현

박 윤 주, 문 창 주, 박 대 하, 백 두 권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실

Design and Implemention of A Certificate and Key Management System based on the WEB

Yoon-joo Park, Chang-joo Moon, Dae-Ha Park, Doo-kwon Baik
Software System Lab., Dept. of Computer Science & Engineering, Korea University

요 약

전자상거래에 대한 요구가 급증하고있는 오늘날, 상대방의 신뢰성을 보증해 주는 인증서 사용이 빈번해지고 있다. 기존에는 인증기관이 발급한 인증서를 사용자가 보관하다가, 전자상거래를 할 경우 이를 전자서명과 함께 상대방에게 제시하여 서로의 신뢰성을 확인하였지만, 이 방법은 사용자가 인증서와 비밀키를 자신의 하드웨어 디스크에 보관하거나, 스마트 카드 또는 플로피디스켓에 휴대하고 다녀야 하는 번거로움이 있었다. 사용자의 이동이 많아지고, 사용자가 자신의 위치나 하드웨어 플랫폼에 상관없이 전자상거래를 하고자 하는 요구가 증대됨에 따라서, 인증서와 비밀키를 휴대해야 하는 불편함은 커다란 제약점이라고 할 수 있다.

본 논문에서는 이러한 문제를 극복하고, 사용자가 어느 곳에서든지 웹브라우져를 사용하여 쉽게 인증서와 비밀키를 사용할 수 있도록 하는 웹기반 인증서 및 키관리 시스템의 설계와 구현방법을 제안한다. 제안된 시스템은 사용자가 복잡한 패스워드를 기억하지 못한다는 점과 패스워드가 쉽게 노출 될 수 있다는 점을 고려하여 SPEKE 에서 제시한 방법을 활용하여 로그인 하였고, 시스템에 외부인이 침입할 경우에 대비하여 데이터베이스 안의 중요한 정보들은 암호화하여 저장하도록 하였으며, SSL 이 설정되지 않았을 경우에도 안전하게 인증서를 위탁할 수 있도록 사용자와 인증서 관리 시스템은 정보를 암호화하여 통신하도록 한다.

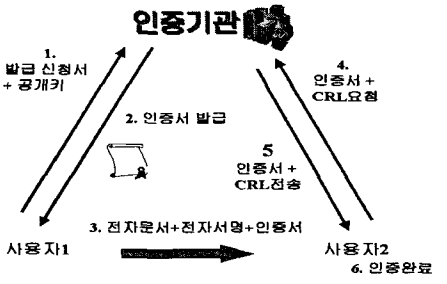
1. 서론

인증서는 전자상거래 시 인증기관(CA)이 전자서명을 통하여 전자서명 공개키와 이를 소유하는 자연인 또는 법인과의 귀속관계를 확인, 증명하는 전자적 정보를 말하는 것으로, 전자상거래 시에 상대방에게 이를 제시하여 상호 신뢰성을 확인하는데 사용된다.[1]

인증서는 인증기관에 의해 발급되는데, 사용자가 인증서 발급을 요청하면, 인증기관이 사용자의 신뢰성을 판단한 후, 믿을 수 있는 사람 또는 기관 이라고 판단될 경우에 발급한다. 인증서를 발급 받은 사용자는 이를 보관하다가 전자상거래를 하고자 할 경우, 전자서명과 함께 상대방에게 제시하고, 서로 인증이 확인되면 전자상거래를 한다. 다음은 인증서 발급과 사용과정을 단계별로 나타내고 있다.

1. 사용자가 공개키/비밀키 쌍을 생성 ·
   "사용자의 공개키"+"발급신청서" 전송
2. 인증기관이 사용자 신원 확인 후 인증서 발급
3. 사용자 1 이 전자상거래를 원하는 사용자 2 에게
   "전자문서 + 전자서명+ 인증서" 전송
4. 사용자 2 는 인증기관에게 사용자 1 에 대한 인증서와 인증서 취소 리스트를 요청
5. 인증기관은 사용자 2 의 신원 확인 후, 사용자 2 에게 사용자 1 의 인증서와 인증서 취소리스트 전송
6. 사용자 2 는 사용자 1 의 인증서와 디지털 서명을 인증기관에서 받은 사용자 1 의 인증서와 비교. 검토하여 사용자 1 의 인증을 확인

위 단계를 거쳐서 상호 인증을 확인한 후, 사용자 1 과 사용자 2 는 전자상거래를 할 수 있게 된다. [그림 1]은 인증서를 발급 받아서 사용하는 모습을 나타내고 있다.



[그림 1]인증서의 발급 및 사용

기존에는 "단계 2".에서 인증서를 발급 받은 사용자가 자신의 하드웨어 디스크, 또는 플로피 디스켓에 인증서와 비밀키를 보관하다가 전자상거래 시에 "단계 3"에서와 같이 상대방에게 전자문서, 비밀키를 이용해 생성한 전자서명, 인증서를 함께 전송하였다. 하지만, 사용자가 이동 할 때마다 플로피 디스켓을 휴대하는 방법은 번거로울 뿐만 아니라, 이를 분실할 경우, 인증서와 비밀키가 타인에게 타인에게 노출될 수도 있다는 점에서도 보안상의 한계를 가지게 된다.

본 논문은 이러한 문제점을 극복하는 방안으로, 사용자가 인증서와 비밀키를 필요로 할 경우, 웹 브라우져를 사용하여 로그인 하면, 어느 곳에서든지 자신의 인증서와 비밀키를 받아서 사용할 수 있도록 하는 "웹 기반 인증서 관리 시스템"을 제안한다.

2. 웹 기반 인증서 및 키관리 시스템

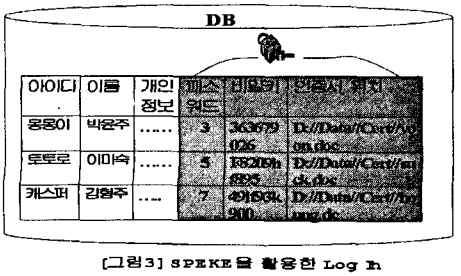
웹 기반 인증서 관리 시스템은 사용자가 웹브라우져를 사용하여 시스템에 로그인 하면, 사용자의 신원을 확인한 후, 시스템이 관리하고 있던 인증서와 사용자의 비밀키를 안전하게 사용자에게 전송해 주는 시스템이다. 이 시스템은 보안을 위하여 비대칭 알고리즘인 RSA를 사용하여, 중요한 정보들을 암호화하여 저장하고, 관리한다.

2.1 인증서 관리 시스템의 구조

인증서 관리 시스템은 크게 두 부분으로 구성되어 있다. 즉, 인증서와 키를 관리하는데 필요한 정보들을 가지고 있는 "데이타베이스부분"과, 실제로 인증서 파일들을 저장하고 있는 하드디스크의 "디렉토리"부분 이다. 데이타베이스에는 사용자에 대한 정보와, 사용자의 비밀키, 그리고, 인증서가 저장되어있는 디렉토리 위치만을 저장하도록 하고, 실제로 인증서는 시스템 내의 하드디스크 디렉토리 안에 파일 형태로 저장하여서, 데이타베이스가 불필요하게 커지는 것을 방지하고, 인증서를 관리하기 쉽도록 한다.

2.1.1 데이타베이스 구조

데이타베이스는 "사용자의 이름", "아이디", "패스워드", "개인정보", "사용자의 비밀키", "인증서 저장 위치" 등을 저장한다. 이때, "이름", "아이디", "개인정보"는 그냥 평문으로 데이타베이스에 저장하지만, "패스워드"와 "사용자 비밀키", "인증서의 저장위치"와 같이 외부의 침입으로부터 보호해야 하는 정보들은 비대칭 암호화 기법인 RSA 알고리즘을 사용하여 시스템의 공개키(512byte 사용)로 암호화하여 저장하도록 한다. 이렇게 암호화된 정보를 사용하고자 할 경우에는, 시스템의 비밀키로 해독하여 사용하도록 한다. [그림 2]는 인증서 관리시스템의 데이타베이스의 보습을 보여주는 예로, 회색부분은 RSA 알고리즘을 사용하여 시스템의 공개키로 암호화된 정보들을 나타낸다.



[그림 3] SPEKE을 활용한 Log in

2.1.2 디렉토리에 저장된 인증서

인증서들은 시스템 내의 하드디스크 안에 파일 형태로 저장된다. 이때, 인증서들은 RSA 알고리즘을 사용하여, 시스템의 공개키(512byte)로 암호화하여 저장하도록 한다.

따라서, 외부인이 인증서 관리 시스템에 침입하여 인증서를 찾아내어도, 시스템의 비밀키가 없으므로 인증서를 해독할 수 없게 되어 시스템의 보안 단계를 높이도록 한다.

2.2 인증서 관리 시스템의 운용

3.1 절에서 설명한 인증서 관리 시스템은 사용자가 인증서와 비밀키를 요청할 때, 다음 네 단계를 거쳐서 사용자에게 안전하게 인증서를 제공하도록 설계한다.

사용자가 시스템에 로그인 하려고 하면, 사용자의 아이디와 패스워드로 신원을 확인한다. 사용자의 신원이 확인되면, 인증서의 위치를 저장하고 있는 데이타베이스에 접근하여 인증서의 위치를 알아내고 비밀키를 얻는다. 그 다음 해당 디렉토리에서 인증서와 비밀키를 찾아온다. 마지막으로, 인증서와 비밀키를 사용자에게 제공한다.

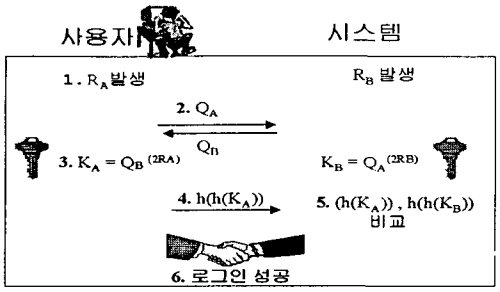
2.2.1 로그인 부분

로그인은 사용자의 아이디와 패스워드를 입력 받아서 이루워 지도록 한다. 하지만 이 방법은 패스워드가 간단하면 쉽게 노출될 수 있고, 반면에 패스워드가 복잡하면 사용자가 기억하기 힘들다는 문제가 있다. 따라서, 로그인 부분에서는 SPEKE 논문에서 제안한 방법을 활용하여, 간단한 패스워드를 사용하면서도 보안단계를 높이도록 한다. SPEKE은 간단한 패스워드로 복잡한 키를 임의로 만든 후, 이 키가 사용자와 시스템간에 공유되는 지를 검증하여, 상호 인증을 확인하는 방법이다.[1].

[SPEKE 을 사용한 로그인 단계]

1. 사용자(A)와 시스템(B)에서 각각 난수 R_A, R_B를 발생시킨다.
2. 각각 자신이 가지고 패스워드에 Q_A = 공유키(2RA) Q_B = 공유키(2RB)를 만들어 상대방 에게 전달한다. (시스템은 사용자가 입력 ID를 사용하여 데이타베이스에 저장된 사용자의 패스워드를 알아낸다.)
3. 사용자(A)에서는 $K_A = Q_B$ (2RA) 로 키를 생성하고, 시스템(B)에서는 $K_B = Q_A$(2RB)로 키를 만들어낸다.
4. 생성된 키가 서로 같은지 검증하기 위하여, 사용자(A)는 단방향 해쉬 알고리즘인 MD5 를 사용하여 생성된 $h(h(K_A))$를 시스템(B)에게 전달한다.
5. 시스템(B)은 전달된 $h(h(K_A))$와 자신이 생성한 $h(h(K_B))$의 값이 같은지를 확인하고, 두개가 서로 같으면 사용자의 패스워드가 맞으므로 로그인 시킨다.

[그림 3]은 SPEKE을 사용한 로그인 과정 을 나타내고 있다.



[그림3] SPEKE을 활용한 로그인

2.2.2 데이터베이스로부터 비밀키와 인증서 위치 검색

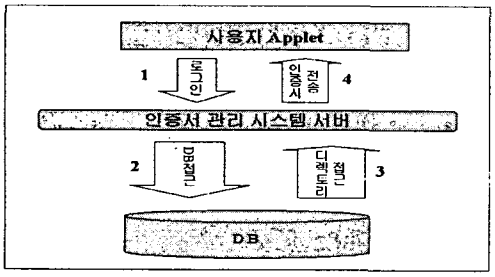
사용자의 인증이 확인되면, 인증서 관리 시스템의 서버는 데이타베이스에 접근하여 사용자의 인증서가 저장된 위치와 사용자 비밀키를 알아낸다. 이 정보들은 2.1.1 절에서 설명하였듯이, RSA 알고리즘 사용하여 512byte 의 시스템 공유키로 암호화 되어있으므로, 시스템의 비밀키(512byte)로 해독한다. [2].

2.2.3 인증서 파일 탐색

데이타베이스로부터 인증서의 저장위치를 알아낸 서버는, 인증서 파일을 찾아온다. 이 파일도 외부의 침입에 대비하여 시스템의 공유키로 암호화된 상태이므로 시스템의 비밀키(512byte)를 사용하여 해독한다. (RSA 알고리즘 사용) [2]. [그림 4]에 "3" 은 이 부분을 나타내는 것이다.

2.2.4 인증서와 비밀키를 사용자에게 전송

해독된 인증서와 비밀키를 사용자에게 안전하게 전송하기 위해서는, 다시 한 번 인증서와 비밀키를 대칭키 알고리즘인 DES 를 사용하여 암호화 한다. 이때, 키로는 로그인 할 때, SPEKE 을 사용하여 생성하였던 사용자와의 공유키([그림 3]의 K_B 가 이용된다. 암호화된 인증서는 사용자 측에 전송되어 사용자가 가지고 있는 시스템과의 공유키(K_A)로 해독되어(DES 알고리즘 사용) , 원래 사용자의 평문 인증서와 비밀키를 얻을 수 있다. [그림 4]는 인증서 관리 시스템이 운용되는 모습을 나타내고 있다.

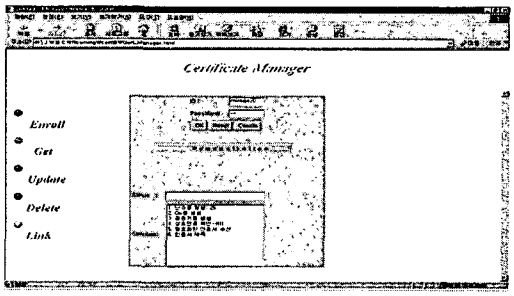


[그림4] 인증서 관리 시스템

3. 구현사례

3.1 구현화면

[그림 5]는 본 논문에서 제안한 인증서 관리 시스템의 구현화면으로, 웹 브라우저에서 사용자가 인증서 관리 시스템에 접속하였을때, 애플릿이 떠서 인증서를 받아올 수 있도록, HTML 과 자바 애플릿으로 구성되어 있다.



[그림 5] 인증서 관리 시스템의 구현 화면

3.2 구현방법

웹 페이지는 HTML 로 구현하였고, 이 HTML 문서 안에 자바 애플릿을 삽입하였다. 사용자(클라이언트)는 이 자바 애플릿을 사용하여 시스템에 로그인 하고, 인증서를 받아오도록 구성된다.

인증서 관리 시스템 서버는 자바 프로그램으로 구현하여, 클라이언트의 자바 애플릿과 소켓을 사용하여 통신하도록 하였다. 서버는 사용자의 SPEKE 에서 제안한 알고리즘을 활용하여 로그인을 처리하고, 데이타베이스를 사용하여 데이타베이스에 접근하도록 하였으며, 자바 패키지안에 있는 RSA 알고리즘과 DES 알고리즘을 사용하여 암호화하고 해독하는 과정을 처리하였다.

4. 결론

본 논문은 웹 기반 인증서 관리 시스템을 설계하고 구축하여 사용자가 어느 곳에서든지 자신의 인증서와 비밀키를 제공 받아 전자상거래를 할 수 있도록 했다는 점에서 의의가 있다.

시뮬레이션 프로그램으로 논문의 내용을 구현한 결과, 사용자가 로그인 할 때, 간단한 패스워드로 안전하게 시스템에 접근할 수 있음을 알 수 있었고, 또한 데이타베이스 안의 정보들과 인증서를 암호화하여 저장하여 보안 단계가 높은 시스템을 구축할 수 있다는 점 역시 검증할 수 있었다.

추후의 해결과제로는, 인증서의 암호화와 해독과정이 두 번 반복되어 오버헤드가 발생하므로, 이에 대한 해결 방안을 모색해야 한다.

[참고문헌]

[1] David P.Jablon, "Strong Password-Only Authenticated Key Exchange", March 2,1997, http://www.integritysciences.com/speke97.html

[2] Jonathan Knudsen, "java cryptography", May 1998

[3] http://www.entrust.com/roaming/

[4] RadRerlman, "Secure Password-Based Protocol for Downloading a Private Key", http://raedperlman@sun.com

[5] Christian Gilmore, David Kormann, Avidl D.Rubin, "Secure Remote Access to an Internal Web Server",{cgilmore,davek,rubin}@research.att.com