전자 문서의 안전한 교환을 위한 보안 환경 구축*

류가현°, 김상진, 오희국
한양대학교 전자계산학과

Secure Environment for Exchanging Digital Documents

Gahyun Ryu°, Sangjin Kim, Heekuck Oh
Department of Computer Science and Engineering, Hanyang University

요 약

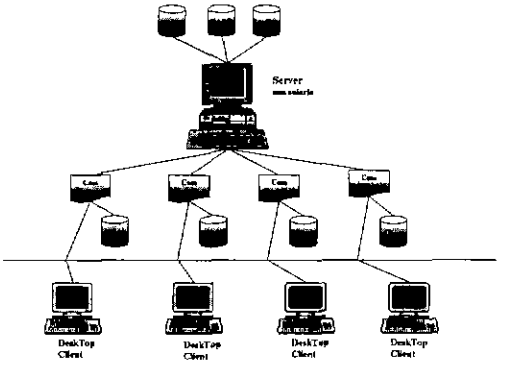
이 논문은 안전한 통신을 보장하고, 전자결제 기능을 제공하는 SEEDS(Secure Environment for Exchanging Digital Signatures)의 설계와 구현에 대해 기술한다. SEEDS는 일반 기업이나 단체에서 근거리망을 통해 서류를 결재하여 교환하는데 개발된 전자결재시스템이다. 따라서 SEEDS는 사무 환경 특성에 적합한 시스템 구조와 암호화 프로토콜을 사용한다. SEEDS의 결재 프로토콜은 공중키 암호화 알고리즘을 기반으로 하는 중재결재 방식을 사용하며, 결재된 서류뿐만 아니라 통신 메시지의 보안을 위해 새롭게 개발한 키 교환과 메시지 전송 프로토콜을 사용한다.

1. 서론

컴퓨터 통신의 발달로 네트워크를 이용한 정보 교환이 대중화되었으며, 교환되는 정보 또한 다양화되어 가고 있다. 그러나 통신 채널로 전달되는 정보는 아무에게나 노출되어 있다고 하여도 과언이 아니며 권한 없는 사용자에 의해 남용될 소지가 있다 이처럼 정보 교환에서 보안이 중요하지만 보안에 대한 인식이 부족하여 적절한 보안 조치를 취하지 못하는 것이 현실이다. 따라서 이 논문에서는 안전하게 전자문서를 교환할 수 있는 환경을 구축하는데 사용할 수 있는 전자결재시스템 SEEDS의 설계와 구현에 대해 기술한다.

2. 시스템의 구조

SEEDS는 사무 환경의 특성과 클라이언트의 특성을 고려하여 그림 1과 같은 별 구조의 통신구조를 채택하며 중재 결재 방식[1]을 사용하고 있다. 다시 말하면 사무 환경에서는 서류를 중앙에서 관리할 필요성이 있으며 중재 결재 방식에서는 서버가 증인 역할을 해야한다. 그리고 클라이언트는 항상 온라인 상태를 유지하지 않기 때문에 별 구조를 사용해야 한다. 따라서 SEEDS의 중앙 서버는 SEEDS에서 필요한 각종 보안 메카니즘의 인증 서버 역할과 전자 결재에 대한 증인 역할을 한다. 뿐만 아니라 클라이언트 시스템들간에 통신을 중재하여 주며 전자결재시스템에서 필요한 정보를 관리하는 데이터베이스 서버 역할도 한다. SEEDS의 클라이언트는 문서를 작성, 수정, 열람, 그리고 결재할 수 있는 기능을 가지고 있으며 전자우편 클라이언트와 유사하게 구현되어 있다.



<그림 1> SEEDS의 시스템 구성도

SEEDS에서는 버클리 대학의 데이터베이스 코드를 기반으로 하고 있는 비정형 데이터베이스 관리기인 XDB를 사용하여 필요한 각종 데이터베이스를 구현하였다. 서버에는 결재와 열람이 완료된 문서를 영구저장하기 위한 문서 데이타베이스, 각 사용자의 공중키를 보관하고 있는 공중키 데이터베이스, 시스템에서 발생하는 중요 사건들을 기록하는 로그 데이터베이스

* 이 논문은 정보통신부의 출연금으로 수행한 초고속정보통신 응용기술개발사업의 연구결과이다.

가 있다. 또한 교환되는 문서들을 일시적으로 보관하는 사용자 문서함이 있다. 이 중 로그데이타베이스는 XDB를 사용하여 구현하지 않고, 파일시스템을 이용하여 구현하였다. 클라이언트에는 서버의 공중키 데이터베이스를 캐시하는 공중키 캐시 데이터베이스가 있으며, 수신한 문서를 보관하는 문서함들이 있다 서버는 유닉스 환경에서 구현 되어 있으며, 현재 SUN시의 Solaris2.4에서 운영되고 있다. 클라이언트는 마이크로소프트사의 윈도우즈 95/98/NT용으로 개발되어 있다

3. 전자결제 프로토콜과 키 교환 프로토콜

SEEDS에서는 RSA(Rivest-Shamir-Alderman) 공중키 암호화 알고리즘을 기반으로 하는 중재 결제 방식[2]을 사용하며, 결제하는 문서의 은밀성을 보장하는 봉합 결제 방식과 보장하지 않는 공개 결제 방식들을 제공한다. 중재 결제 방식이란 믿을 수 있는 서버가 중인으로 참여하는 결제 방식을 말한다. 그러므로 중재 결제 방식을 사용하는 SEEDS에서도 중앙 서버가 중인으로 결제과정에 항상 참여한다. 중앙 서버는 결제된 문서를 다른 클라이언트에게 중계하기 전에 결제를 검증하여 문제가 없을 경우에만 전달하여 준다. SEEDS의 결제 방식과 기존 중재 결제 방식과의 차이점은 결제의 은밀성을 보장하기 위해 수신측 공중키로 암호화하지 않고, 교환되는 모든 정보의 보안을 위해 사용되고 있는 메시지 교환 프로토콜을 이용하여 은밀성을 보장한다는 것이다. 사용하는 결제 블록 구조에는 결제 문서의 다이제스트, 결제 시간, 그리고 결제자 이름이 포함된다.

Message 1. $A \rightarrow S$: A, N_a

Message 2. $S \rightarrow A$: $\{\{N_a, N_s, A, K_{as}\}_{-K_s}\}_{+K_a}, \{N_a\}_{K_{as}}$

Message 3. $A \rightarrow B$: $\{N_s\}_{K_{as}}$

<그림 2> SEEDS 키 교환 프로토콜

Message 1. $A \rightarrow S$: $\{H(M), M\}_{K_{as}}$

Message 2. $S \rightarrow A$: $\{H(M'), M'\}_{K_{as}}$

<그림 3> SEEDS의 메시지 교환 프로토콜

SEEDS에서는 교환되는 모든 정보의 보안을 보장하기 위해 별 구조에 적합하도록 고안한 그림 2와 3에 기술된 키 교환 프로토콜과 메시지 교환 프로토콜을 사용한다. SEEDS의 키교환 프로토콜은 교환되는 정보의 은밀성과 무결성을 보장하여 주면서 사용자를 인증하여 준다. 사용자 인증은 공중키 시스템을 활용하여 제공하고 있으며, 재전송 공격에 대한 방어력을 높이기 위해 nonce 도전 기법[3]과 naming 기법을 적용하였다. SEEDS는 타임스템프를 사용하는 Kerberos 시스템[4]과는 달리 시스템들간에 클럭동기화를 요구하지 않으면서 메시지의 최근성을 보장하고 있다. 키 교환 프로토콜을 이용하여 세션키를 교환하였으며 그림 3에 기술된 메시지 교환 프로토콜을 이용하여 정보를 교환한다. 그림 3에서 M은 클라이언트가 서버에게 전달할 정보를 나타내며, M'은 서버의 M에 대한 응답이다 전달되는 정보의 무결성을 보장하기 위해 클라이언트, 서버 모두 메시지 다이제스트 H(M)을 M과 함께 세션키로 암호화하여 전달한다.

4. 통신 메시지 구조

SEEDS에서 교환되는 통신 메시지의 종류는 크게 문서 교환과 관련된 메시지, 시간 요청 메시지, 공중키 관련 메시지로 분류할 수 있으며, 모든 메시지는 앞 절에서 설명한 메시지 교환 프로토콜에 따라 교환된다. 이 때 키 교환 프로토콜과 마찬가지로 nonce를 이용하여 메시지의 최근성을 보장한다. 메시지 교환 프로토콜에서 M은 다음과 같은 형태를 지니고 있으며,

$$Msg\ Type + N_a + [Content]$$

그것의 응답 M'은 다음과 같다.

$$Ack\ Type + N_a + [Content]$$

예를 들어 SEEDS에서는 결제 시간, 문서 생성 시간 등 필요한 시간은 항상 서버에게 요청을 하여 얻어서 사용한다 이것은 각 클라이언트에서 사용되는 시간을 논리적으로 동기화하기 위해서 이런 방식을 사용하고 있으며, 그 형태는 다음과 같다

M : DS_TIME_REQ + Na

M'' DS_ACK_SUCC + Na + time

5. 소프트웨어 구성도

SEEDS의 소프트웨어 모듈은 그림 4와 같이 서버 시스템을 구성하는 소프트웨어 모듈들과 클라이언트 시스템을 구성하는 소프트웨어의 모듈들로 나뉘어진다. 클라이언트 시스템에는 암호화 모듈, 전자 결제 모듈, 데이터베이스 인터페이스 모듈, 사용자 인터페이스 모듈, 그리고 통신 모듈이 있다. SEEDS에서는 RSA, IDEA, SHA 알고리즘과 Micali-Schnorr pseudo random 비트 생성기를[5-6] 이용하여 필요한 모든 보안 기능을 제공한다. 암호화 모듈은 위에 열거한 알고리즘들을 구현한 라이브러리이다. 전자 결제 모듈은 암호화 모듈, 통신 모듈 등

과 상호 작용하여 문서를 암호화하고 결재 블록을 생성해주고, 이를 복호화하여 무결성과 인증을 확인해주며, 키 교환 프로토콜과 메시지 교환 프로토콜을 수행하여 주는 모듈이다. 데이터베이스 인터페이스 모듈은 클라이언트에 중복되어 있는 공중키 캐시 데이터베이스와 수신한 문서들이 저장되어 있는 문서함들을 접근하기 위한 인터페이스를 제공하며, 사용자 인터페이스 모듈은 문서를 작성, 열람, 결재하는 그래픽 사용자 인터페이스와 관련된 모듈이다.

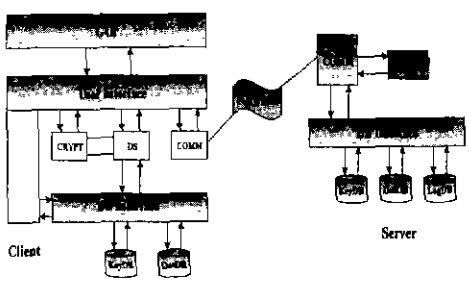
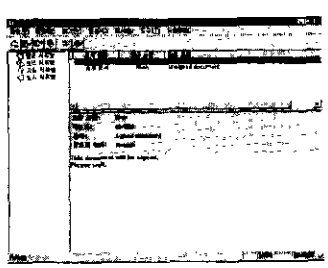


그림 4 소프트웨어 구성도

서버 시스템은 통신 모듈, 데이터베이스 인터페이스 모듈, 암호화 모듈, 파일 관련 모듈 그리고 관리자 모듈로 구성되어 있다. 통신 모듈은 각 클라이언트와 메시지를 교환하기 위해 필요한 TCP/IP 소켓과 관련된 모듈이다. 데이터베이스 인터페이스 모듈은 서버가 관리하는 공중키 데이터베이스와 문서 데이터베이스의 창구 역할을 하는 모듈이며, 서버의 암호화 모듈은 클라이언트와 마찬가지로 각종 암호화 알고리즘을 구현한 라이브러리이다. 파일 관련 모듈은 로그 데이터베이스에 접근하기 위한 인터페이스를 제공하며, 파일 잠금 기능을 제공한다. 이 시스템에서 데이터베이스를 비롯하여 필요한 모든 동기화 메카니즘은 이식성을 높이기 위해 C 라이브러리에서 제공하는 파일 잠금 기법을 사용하고 있다. 그리고 관리자 모듈은 시스템 관리자에게 데이터베이스 관리를 용이하게 해주는 툴(tool)들을 제공하는 모듈이다.



<그림 5> 클라이언트 인터페이스

6. 사용자 인터페이스

클라이언트 시스템의 사용자 인터페이스는 그림 5와 같이 현재 널리 사용되고 있는 전자우편과 유사한 모습을 지니고 있다. 사용자는 이 시스템을 이용하여 안전하게 문서를 결재하고 전달할 수 있다.

7. 결론 및 향후 연구과제

본 논문에서는 사무환경에서 활용될 수 있는 전자 결재 시스템 SEEDS의 설계와 구현에 대해 기술하였다. SEEDS는 전자 문서를 안전하게 교환하여 줄뿐만 아니라 사무 환경에서 활용될 수 있는 전자 결재 서비스를 제공하고 있다. SEEDS는 사무환경과 클라이언트 시스템의 특성을 고려하여 중재 결재 방식에 적합한 별 구조의 클라이언트-서버 구조를 지니고 있다. 필요한 보안 서비스를 제공하고 RSA, IDEA, SHA, 그리고 Micali-Schnorr의 pseudorandom 비트 생성기를 사용하며, 새로운 암호화 프로토콜을 설계 구현하였다. 개발된 암호화 프로토콜은 교환되는 정보의 은밀성과 무결성을 보장하여 주며, 사용자를 인증하여 준다. 뿐만 아니라 메시지 재전송 공격과 메시지 변조 공격 등에 대하여 충분히 고려된 프로토콜이다. 향후 과제로는 시스템 활용 측면뿐만 아니라 현재 제공되고 있는 보안 방법을 검토하고, 필요에 따라서는 보다 더 효율적이고 안전한 방법으로 확장을 모색하여 보아야 한다. 또한 인터페이스 측면에서 사용자의 편리성, 시스템 실용성 측면에서 확장할 요소가 있다. 이 외에 인터넷과의 연동도 고려해 볼 필요가 있다.

참고 문헌

[1] Selim G. Akl, "Digital signitures: a tutorial survey," *IEEE Computer*, vol. 16, no 2, pp. 15-24. Feb., 1983.

[2] R L. Rivest, A. Shamir, and L. aldeman, "A method for obtaining digital signatures and public key cryptosystems," *Comm of the ACM*, vol. 21, no. 2, pp. 120-126, Feb 1978.

[3] Martin Abadi and Roger Needham, "Prudent engineering practice for cryptographic protocols," DEC System Research Center Report No. 125, June 1994.

[4] B. Clifford Neuman and Thordore Ts'o. "Kerberos: an authentication service for computer network." *IEEE Comm. Mag*, vol. 32, no. 9, pp. 33-38, Sept., 1994.

[5] Xuejia Lai and James Massey, "A proposal for a new block encryption standard," *Proc of Advances in Cryptography(EUROCRYPT'92)*, pp. 55-70, 1992.

[6] S. Micali and C. P. Schnorr, "Efficient, perfect polynomial random number generator," *Journal of Cryptology*, vol. 3, pp. 157-172. 1991.