서버의 안전성 향상을 위한 전자서명 검증 에이전트의 구현

이용준°, 이옥경, 정재동, 오해석
숭실대학교 대학원 컴퓨터학과
yjlee°@koscom.co.kr, oklee017@lycos.co.kr, jdd@koscom.co.kr, oh@computing.ssu.ac.kr

Implementation of Digital Signature Verify Agent for Safety of Server

Yong-Jun Lee°, Jea-Dong Jong, Hea-Suk Oh
Dept. of Computing, Graduate School, Soongsil University

요 약

PKI(Public Key Infrastructure)기반의 공인인증서가 활성화됨에 따라 인터넷뱅킹, 증권거래시스템, 전자입찰, 전자민원 등 신원확인이 요구되는 어플리케이션에 전자서명이 적용되고 있다.

다수의 사용자가 인증서로 로그인과 부인방지를 위한 전자서명을 전송함으로써 서버는 상대적으로 많은 전자서명과 인증서검증을 수행해야 한다. 프로그램 단계에서 검증을 수행하기 위해서는 검증모듈의 API(Application Programming Interface)를 호출해야 하며, 이는 프로그램의 복잡도를 증가시킨다. 이러한 문제는 어플리케이션과 검증이 독립적이지 않기 때문에 검증에 관련된 장애가 발생하면 어플리케이션은 검증모듈에 종속적으로 장애가 발생하게 된다.

본 논문의 검증 에이전트는 서버환경이 서버-클라이언트 또는 웹 어플리케이션에 독립적으로 전자서명 검증을 담당한다. 따라서 어플리케이션은 검증이 필요한 시점에 검증 에이전트를 호출하게 됨으로써 프로그램의 복잡도를 줄이고 서버의 안정성을 향상시킨다.

1. 서론

최근 정보통신의 발전의 영향으로 실생활의 서비스가 온라인으로 전환되고 있다. 개방형 네트워크에서의 통신은 정보의 위변조, 노출, 부인 등 각종 역기능에 의한 위험이 예상되고 있다. 이에 따라 정보보호기술은 통신상의 안정성과 신뢰성을 제공하는 중요한 요소로 인식되고 있다.

사용자의 공개키를 안전하고 신뢰성 있게 전달하는 방법을 제공하는 PKI는는 정보보호기술의 핵심이며 인증(Authentication), 무결성(Integrity), 부인방지(Non-repudiation), 기밀성(Confidentiality), 가용성(Availability)의 기능을 제공한다[1].

전자서명은 개인키의 소유자만이 생성할 수 있다. 검증은 전자서명의 진위여부를 확인하는 과정이며 개인키에 합치하는 공개키의 획득하고 전자서명값에 대한 검증을 수행한다. 공개키 획득은 인증서유효성 검증을 통해서 이루어지게 되는데 인증서에 공개키를 가지고 있기 때문이다. 사용자의 개인키 유출, 분실, 자격변경, 키변경 등의 이유로 인증서 폐지가 가능하며 검증자는 수신한 인증서의 상태가 유효한 것인지를 확인해야 한다[2]. 인증서상태 확인을 위하여 CRL(Certificate Revocation List)[3], OCSP(Online Certificate Status Protocol)[4], SCVP(Simple Certificate Validation Protocol)[5]의 제안되었다.

현재 많은 보안관련 업체에서는 기존 온라인 서비스에 PKI의 기능을 제공하여 개발하고 있다. 고부가 온라인서비스인 인터넷뱅킹, 증권거래시스템, 전자입찰, 전자민원 등 신원확인이 요구되는 어플리케이션에 PKI적용은 의무화가 되었다. 기존 PKI기반 어플리케이션은 인증서검증과 전자서명검증에 따르는 서버측의 부담이 집중되고 있다. 다수의 사용자가 인증서로 로그인과 부인방지를 위한 전자서명을 전송함으로써 서버는 상대적으로 많은 전자서명과 인증서 검증을 수행하기 때문이다. 프로그램 단계에서 검증을 수행하기 위해서는 검증모듈의 API(Application Programming Interface)를 호출해야 하며, 이는 프로그램의 복잡도를 증가시킨다. 이러한 문제는 어플리케이션과 검증이 독립적이지 않기 때문이다. 따라서, 또한 인증서상태검증을 제공하는 CRL, OCSP의 통신장애 및 에러가 발생했을 때, 어플리케이션이 함께 영향을 받는다[6]. 검증에 관련된 장애시 어플리케이션은 검증모듈에 종속적으로 장애가 발생하게 된다. 본 논문의 검증 에이전트는 서버환경이 서버-클라이언트 또는 웹 어플리케이션에 독립적으로 전자서명 검증을 담당한다. 따라서 어플리케이션은 검증이 필요한 시점에 검증 에이전트를 호출하게 됨으로써 프로그램의 복잡도를 줄이고 서버의 안정성을 향상시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 PKI기반 어플리케이션에 대해 분석한다. 3장에서는 서버측 안전성을 제공하는 검증 에이전트를 제안한다. 4장에서는 기대효과를 제시한다. 5장에서는 결론을 맺는다.

2. 관련연구

온라인서비스중 금융거래와 증명서비스를 제공하는 인터넷뱅킹, 증권거래시스템, 전자입찰, 전자민원은 보안기능을 제공하기 위해 PKI가 적용이 의무화되었다[7]. 다수의 사용자가 생성한 전자서명을 전송함으로써 서버는 상대적으로 많은 전자서명과 인증서 검증을 수행해야 한

다. 본 장에서는 PKI기반 어플리케이션의 특성을 분류하고 서버측 검증과정의 집중화에 대한 문제점을 제시한다.

2.1 PKI기반 어플리케이션

신원확인을 위한 로그인과 부인방지 목적의 데이터 또는 문서의 전자서명을 생성과 이를 검증함으로써, 보다 안전한 어플리케이션의 구현이 가능하다. 전자서명 생성측과 검증측은 어플리케이션의 특성에 따라서 서버, 클라이언트, 쌍방에서 이루에 지게 된다. 이때 서버에 집중화되는 어플리케이션은 상대적으로 많은 검증을 수행해야 하는 부담이 있다. <표 1>에서는 PKI기반 어플리케이션의 특성을 비교하였다.

<표 1> PKI기반 어플리케이션의 검증측 비교

인터넷뱅킹	사용자 → 뱅킹시스템	서버
사이버트레이딩	사용자 → 증권거래시스템	서버
전자입찰	입찰자 → 입찰시스템	서버
전자상거래	사용자 → 쇼핑몰	서버
민원행정	사용자 → 전자정부	서버
전자결제	사용자 → 결제시스템	서버
의료정보화	의사 → 의료정보시스템	서버
보안메일	사용자 ↔ 사용자	쌍방
전자계약	기업 ↔ 기업	쌍방
전자무역	기업 ↔ 기업	쌍방
전자영수증	쇼핑몰 → 사용자	클라이언트
전자세금계산서	기업 → 기업	클라이언트
전자출판	출판사 → 사용자	클라이언트

2.2 전자서명과 검증

전자서명은 전자문서나 메시지를 보낸 사람의 신원확인과 그 내용이 전송중에 변조되지 않았다는 무결성을 제공하기 위해 사용된다. 전자서명에서는 해당 문서의 기밀성은 제공하지 않는다. 단지 메시지의 변조 여부와 송신자의 신원만 확인하는데 목적이 있다. 전자서명을 사용함으로써 부가적으로 부인방지가 가능하게 된다. 전자서명은 공개키 암호 알고리즘에 기반을 두고 있는데, 개인키로 서명을 하고 공개키를 사용해서 서명검증을 한다[8].

서명검증 과정은 서명자 인증서가 유효성을 검증한 후 전자서명 검증을 한다. 인증서 유효성 검증은 신뢰할 수 있는 최상위 기관까지의 인증서 경로를 생성하고 경로 검증을 한다. 이때 발급자 인증서로부터 사용자 인증서까지의 상태검증을 CRL(Certificate Revocation List), OCSP(Online Certificate Status Protocol)을 통해 확인한다. 이 모든 과정이 성공하면 전자서명에 대하여 검증을 수행하게 된다.

3. 제안하는 검증 에이전트

본 논문에서는 대규모의 검증이 기존 방식에 대비하여 복잡도를 감소시키기 이루어지는 서버에서 안정성을 보장하고 위해 검증 에이전트를 제안한다.

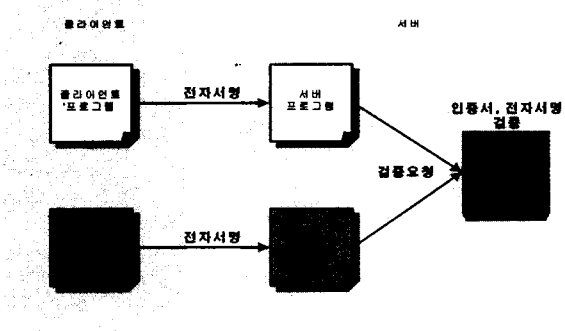
3.1 기존 전자서명과 검증 적용방식

TCP/IP기반의 서버-클라이언트 어플리케이션의 경우는 윈도우환경에는 DLL(Dynamic Link Library)를 링크하여 전자서명 또는 검증이 필요한 경우에 호출을 하는 방식을 사용한다. 유닉스환경에서는 정적 라이브러리와 동적 라이브러리를 링크하여 사용한다.

HTTP기반의 웹 어플리케이션에서는 ActiveX를 이용하여 전자서명 API를 호출하고 서버에서는 Java의 경우 Class와 Jar형식의 라이브러리를 사용하거나 ASP는 DLL의 API를 호출한다.

3.2 검증 에이전트의 구성도

제안하는 검증 에이전트는 서버측에서 클라이언트-서버 또는 웹 어플리케이션이 요청하는 인증서와 전자서명 검증을 담당한다. 클라이언트가 생성한 전자서명과 인증서를 서버에서는 검증 요청을 검증 에이전트에게 요청하면 검증을 수행하고 그 결과를 응답한다. 검증 에이전트는 인증 경로 생성과 검증을 수행하고 인증서 상태 확인에 대해서는 기존의 표준방안인 CRL, OCSP, SCVP등을 통해 결과를 응답한다. [그림 1]은 검증 에이전트의 구조를 명시한다.



[그림 1] 검증 에이전트의 구조

3.3 검증 에이전트의 기능

● 인증서 상태 확인

인증서의 상태를 검증하는 모듈로써 CRL, OCSP, SCVP를 통해 검증한다. 검증방법의 선택은 사용자의 인증서로 판단한다.

● 인증 경로 생성

인증 경로 생성을 하는 모듈로써 CA의 인증서로 이루어진 인증기관 인증경로와 요청에 의한 응답으로 생성된 인증경로를 사용하여 인증경로를 생성한다.

● 인증 경로 검증

인증 경로 검증하는 모듈로써 인증 경로가 주어지면

인증서 정책 정보, 인증서 정책 맵핑 정보를 이용하여 검증한다.

- 인증서 정책 맵핑

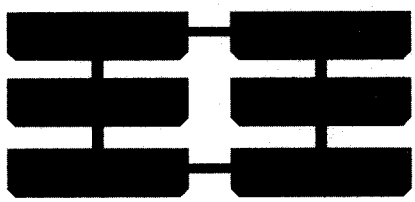
정책을 관리하는 모듈로써 인증서 정책 정보와 인증서 정책 맵핑 정보로 검증한다.

- 전자서명검증

인증서 유효성 검증이 완료되면 원문과 전자서명을 암호 알고리즘으로 수행하여 무결성 진위여부를 결정한다.

- 결과응답

인증서와 전자서명의 검증된 결과로 성공과 실패에 대해 요청한 서버 프로그램에게 응답한다. 만약 검증관련 장애가 발생시 장애발생원인을 응답한다.



[그림 2] 검증 에이전트의 기능

3.3 시스템 통합 관리

서버-클라이언트와 웹 어플리케이션이 통합되어 관리하는 환경에서 기존의 검증 API호출 방식을 적용 하면 프로그램 복잡도가 증가한다. 이 때문에 통합관리에 부담이 되는데 검증 에이전트는 단순한 검증 요청과 응답으로 통신하기 때문에 통합관리가 가능하다.

3.4 장애처리

인증서상태 확인 과정에서 CRL, OCSP, SCVP의 외부의 정보를 획득해서 처리해야 하는데 이때 장애가 발생할 수 있다. 기존의 적용 방식에서는 어플리케이션과 검증이 종속적이기 때문에 소스 레벨에서 장애처리가 이루어져야 한다. 그러나 검증 에이전트에서는 응답시 에러만을 전송해주면 처리가 가능하기 때문에 서버 환경의 안정성을 보장한다.

4. 기대효과

기존의 방식은 어플리케이션과 검증처리가 종속적인 것에 반하여 제안한 검증 에이전트는 독립적인 특징을 가지고 있다. 프로그램의 복잡도 측면에서 검증 에이전트는 검증시 마다 API를 호출하는 기존 방식에 비해 요청과 응답으로 간략화되었다. 개발언어에 있어서도 검증 에이전트와 기존 어플리케이션의 통신 프로토콜만 정의해 주면 되기 때문에 독립적이다. 서버-클라이언트와 웹의 통합관리의 제공의 면에서도 우수하다. 검증관련 장애시 기존의 방식은 소스에서 예외처리를 해야 하는 부담이 있다. 따라서 제안한 검증 에이전트는 <표 2>에서 기술한 것과 같은 기대효과를 나타낸다.

<표 3> 제안하는 검증 에이전트와 기존방식의 비교

	기존 검증 방식	검증 에이전트
서버 안전성	보장안됨	보장
프로그램 복잡도	높음	낮음
개발언어 독립성	종속적	독립적
시스템 통합	보장안됨	보장
장애처리	복잡	간략
검증속도	고속	저속

5. 결론

최근 PKI(Public Key Infrastructure)기반의 공인인증서가 활성화됨에 따라 인터넷뱅킹, 증권거래시스템, 전자입찰, 전자민원 등 신원확인이 요구되는 어플리케이션에 전자서명이 적용되고 있다. 다수의 사용자가 인증서로 로그인과 부인방지를 위한 전자서명을 전송함으로써 서버는 상대적으로 많은 전자서명과 인증서 검증을 수행해야 한다. 프로그램 단계에서 검증을 수행하기 위해서는 검증모듈의 API(Application Programming Interface)를 호출해야 하며, 이는 프로그램의 복잡도를 증가시킨다. 이러한 문제는 어플리케이션과 검증이 독립적이지 않기 때문이다. 따라서 검증에 관련된 장애가 발생하면 어플리케이션은 검증모듈에 종속적으로 장애가 발생하게 된다. 본 논문의 검증 에이전트는 서버환경이 서버-클라이언트 또는 웹 어플리케이션에 독립적으로 전자서명 검증을 담당한다. 따라서 어플리케이션은 검증이 필요한 시점에 검증 에이전트를 호출하게 됨으로써 프로그램의 복잡도를 줄이고 서버의 안정성을 향상시킨다.

참고문헌

[1] Vishwa Prasad & Sreenivasa Potakamuri & Michael Ahern. "Scalable Policy Driven and General Purpose Public Key Infrastructure(PKI)", IEEE, 2000.

[2] Ray Hunt. "PKI and Digital Certification Infrastructure", IEEE, 2001.

[3] RFC2459, Certificate and CRL Profile, 1999.

[4] RFC2560, Online Certificate Status Protocol, 2001.

[5] Draft, Simple Certificate Validation Protocol, 2002.

[6] Andre Arnes, Svein J. Knapskog. "Selecting Revocation Solutions for PKI", NORSEC 2000.

[7] Albert Levi & M. Ufuk Caglayan. "An Efficient, Dynamic and Trust Preserving Public Key Infrastructure", IEEE, 2000.

[8] RFC2528, Representation of Key Exchange Algorithm Keys in Internet X.509 Public Key Infrastructure Certificates, 1999.