XML 기반 권한 정책 모듈 구현

신명숙 · 이 준*

Implement of XML-Based Authorization Policy Module

Myeong-Sook Shin · Joon Lee*

이 논문은 2004년도 조선대학교 학술 연구비의 지원을 받아 연구되었음.

요 약

PKI는 비대면한 상황에서 사용자의 인증을 위해서 좋은 해결책을 제시하여 주고 있지만 지역적으로 떨어져 있는 컴퓨팅 환경에서 권한에 대한 해결책을 제시하기에는 미흡하다. 특히 다양한 형태의 사용자 인증, 무결성, 부인 방지의 보안 서비스를 제공하지만 권한 정책은 너무 많은 정보로 인한 복잡도 때문에 쉽게 이해할 수 있는 XML을 사용하여 단순하고 읽기 쉬운 인증서를 만들며 DOM 트리로부터 정보를 얻어 XML 분석 및 표준화된 메소드 이용을 쉽게 한다. 본 논문에서는 분산된 자원을 분산된 사용자의 권한 부여 해결책으로 사용할 수 있는 AAS 모델을 제시하고, 리눅스 기반 아파치 웹 서버에서 XML을 이용하여 권한 정책 모듈을 구현하였다.

ABSTRACT

PKI provides good resolutions for the authentication of user in the situation not to meet each other, but it is not enough to provide the resolution of authorization in distributed computing environments. Especially, we offer a variety forms of the user Authentication, the Integrity and a security service of the Non-Repudiation, but an authorization policy, because of the complexity with a lot of information, using an understandable XML, makes a simple and easy certificate to read, and we get the information from DOM tree and do a XML analysis and stardardized-method usage easily. In this paper, we provide the AAS model being able to use with the solution of the distributed users' authorization, and we implement an authorization policy module, using XML, in the Linux-based Apache Web server.

키워드

PKI, XML, SSL, Authentication, Authorization

Ⅰ. 서 론

PKI(Public Key Infrastructure)는 전자상거래, 네트워크 보안 등 다양한 응용분야에서 X.509 기반으로 발전시켜 나가고 있다[1].

다양한 형태의 사용자 인증, 무결성, 부인 방지의

* 조선대학교 전자정보공과대학 컴퓨터공학과

접수일자 : 2005. 6. 17

보안 서비스를 제공하는 PKI는 비대면한 상황에서 사용자 인증을 위해 좋은 해결책을 제시하여 주고 있지만 지역적으로 떨어져 있는 컴퓨팅 환경에서 권한에 대한 해결책을 제시하기에는 미흡한 것 또한 사실이다. 이러한 문제의 해결 방안으로 지리적으로 분산된 자원을 분산된 사용자들에 의해 사용할 수 있는 권한(authorization) 시스템으로 TLS(Transport Layer Security)와 같은 보안 프로토콜상에서 상호간 인증을 제공한다[2].

본 논문에서 제안된 AAS 모델에서의 권한 정책은 많은 정보의 복잡도 때문에 XML (eXtensible Markup Language)을 이용해 단순하고 읽기 쉬운 인증서를 만든다. XML은 데이터를 구조적인 방법으로 기술할 수 있으며, 데이터의 저장 및 교환이 매우 유연하며, SGML(Standard Generalized Markup Lamguage)의 부분 집합으로서 W3C(World-Wide Web Consortium)에 의해 제안된 언어이다[3]. 또한 DOM 트리로부터 정보를 얻어낼 때, XML 분석과 표준화된 메소드 이용이 용이하다[4].

특히 분산 환경에서 사용할 수 있는 AAS 모델을 제시하고 리눅스 기반 아파치 웹 서버에서 권한 정책 모듈인 AAS 모듈을 XML을 이용하여 구현한다

본 논문의 2장에서는 인증 및 권한 정책 구현의 관련기술인 XML과 SSL(Secure Socket Layer)에 대하여 설명하였고, 3장에서는 웹 기반에서 인증 및 권한 정책 모듈 설계 및 구현하였다. 마지막으로 4장에서는 결론 및 향후 연구 방향을 기술하였다.

Ⅱ. 인증 및 관련 기술

2.1 PKI 구성

인증은 시스템을 보호하기 위해 서버가 사용자에게 사전에 약속된 정보를 제시 할 것을 요구함으로써 사용자를 확인하는 것이다. 공개키 기반 구조의 기본 구성은 그림 2-1과 같으며 인증서발급, 사용 및 취소와 관련된 서비스를 제공한다. 공개키 기반 구조 환경을 구성하는 주요 객체는 인증기관, 인증서 저장소, 최종 사용자 그리고 전자상거래 서비스 제공자로 구성되며 구성 객체의 기본적인 트랜잭션을 보여준다.

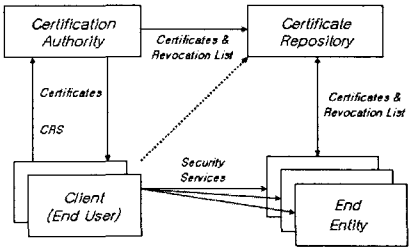


그림 2-1. 공개키 기반 구조 기본 구성 객체
Fig. 2-1 Basic structure object of PKI

인증서는 공개키의 합법성을 보증하는데 이용 된다. 서명을 확인하는 사람은 인증서의 서명을 확인하여 서명에 위조나 변조가 없다는 사실을 확인한다. 현재 인증 시스템에서 사용되는 인증서 양식은 ITU-T X.509v3 형식을 따르고 있다. X.509v3 인증서 형식은 인증서 발급 대상이 되는 객체를 식별하여 공개키를 연관 시킬 수 있는 정보들이 포함된다.

2.2 XML

XML을 기반으로 하는 어플리케이션의 증가와 함께 XML에 적용 가능한 다양한 보안 기술이 제안 및 개발되고 있다. 특히, 최근에는 XML 문서 자체에 대한 전자서명이나 암호화 등에서 확장되어 XML 문서에 대한 접근 제어 기법들에 대한 연구가 진행되고 있다.

XML은 SGML의 복잡성을 제거하고, HTML의 고정된 태그의 한계에서 벗어나 문서 구조를 정의할 수 있다. 이와 같은 장점 때문에 W3C에서는 XML을 웹 데이터의 표준으로 제정했다. XML의 표준화 이후 많은 데이터가 XML로 표현되기 시작했고, 현재는 웹상에서 다양한 XML 데이터가 존재하고 있다.

본장에서는 마크업 문서 표준인 XML 및 XML 문서를 사용해 실제 어플리케이션에서 사용하기 위한 도구로써 DOM에 대해 살펴본다.

DOM은 HTML 문서 또는 XML 문서와 같이 인터넷 문서에 대하여 문서의 내용 및 구조를 객체로 표현하고 그 객체를 핸들링 할 수 있는, 즉 동적으로 문서의 내용과 구조, 스타일을 바꿀 수 있게 하는 플랫폼에 독립적이고 언어 중립적인 인터페이스이다. 또한 DOM은 트리 구조의 문서에서 노드를 노드 객체로 만들고 그 객체에 대한 접근 방법을 제공하며, XML 문서의 요소, 텍스트 등을 노드로 만들고 이들을 접근하고 조

작할 수 있는 API들을 제공한다. 어플리케이션에서는 DOM을 이용하여 보다 편리하게 XML 문서를 다룰 수 있다.

DOM의 중요한 목적은 광범위한 환경과 어플리케이션에서 표준적인 프로그래밍 인터페이스를 제공하기 위한 것으로 개발자는 문서의 생성, 문서 구조의 탐색, 내용의 추가, 삭제, 수정을 API인 DOM을 통해 수행 할 수 있다. 또한 XML은 각각의 요소들을 트리 구조로 표현하며, 원하는 요소에 대하여 트리간의 이동을 통해 찾을 수 있기 때문에 검색, 수정, 삭제가 용이하다. 본 논문에서는 사용-조건 인증서의 XML 문서에 대하여 DOM 트리를 사용한다.

2.3 SSL

SSL은 응용 프로토콜과 TCP/IP사이에 위치하며 데이터의 암호화, 서버의 인증, 메시지의 무결성을 제공한다[5]. 또한 서버에 대한 인증은 반드시 수행되지만 클라이언트에 대한 인증은 선택적으로 수행 할 수 있도록 해준다. SSL은 서버와 클라이언트 양쪽의 TCP/IP 연결을 위해서 핸드쉐이크 프로토콜을 수행하여 종단간 암호화 통신에 합의하고, 암호화 통신과 인증에 필요한 값들을 초기화한다. 초기화 후, SSL은 응용 프로토콜에서 생성한 바이트 스트림의 암호화와 복호화를 수행하게 된다. 즉 HTTP 요청과 HTTP 응답에 포함되는 모든 정보들이 암호화되어 전송됨을 의미한다. 그림 2-2는 핸드쉐이크 프로토콜의 수행 과정을 보여준다.

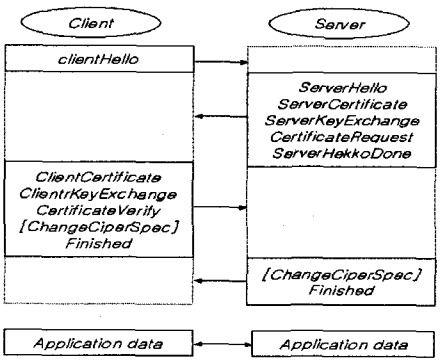


그림 2-2. SSL 핸드쉐이크 프로토콜
Fig.2-2 SSL handshake protocol

Ⅲ. 인증 및 권한 정책 모듈 설계 및 구현

3.1 AAS 모델

AAS 모델은 다중 관리 도메인들과 다중 스테이크홀더들을 포함하는 복잡한 권한 문제들을 처리하고, 접근 제어 정책들을 반영한다. 또한 다른 스테이크홀더들에 대하여 독립적으로 접근 제어의 요구를 허용하며, 무결성에 대한 표준, 부인 방지, 접근 제어 요구사항들의 규약을 제공하기 위해서 제시하는데, AAS 모델에서 AAS는 전자 서명된 인증과 정책 인증서, 사용자 속성 인증서, 자원 사용-조건 인증서로 구성된다. 그리고 AAS 모델은 그림 3-1과 같이 사용자 요청에 따라 자원으로 접근되며 먼저 인증된다. 자원 게이트웨이는 AAS를 접속한 후 정책 인증서에 위치되며, 자원의 사용 조건 인증서들을 모은다. 그리고 나서 AAS 정책 엔진은 접근 제어 결정을 정한다[6].

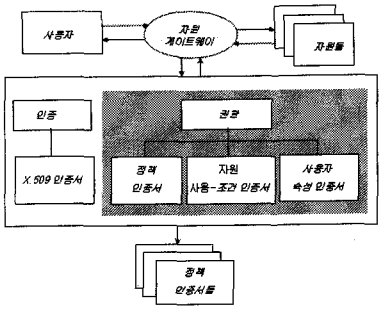


그림 3-1. AAS 모델
Fig.3-1 AAS model

3.2 웹 인증

분산 환경에서 사용자 접근 권한 시스템을 개발하기 위하여 공개키 기반구조의 아파치 웹 서버 인증 및 권한 정책 모듈을 설계한다[9-10].

표준 웹 인증과 접근 제어는 요청이 일어난 도메인에서 사용자가 자신의 이름과 패스워드를 제공하면, 웹 브라우저가 서버 머신에 저장된 사용자 정보와 대조하여 매치하는 인증에 기반한다. 이를 위해서 기존의 아파치를 이용하여 모듈을 구현하는데, 아파치 웹 서버가 광범위하게 사용되며 높은 성능의 프리웨어 서버는 API(Application Program Interface)로 만들어진다. 제3의 프로그래머가 새로운 서버 기능성을 추가하며, 동일하게 사용 가능한 확장 API를 사용하면서 인증,

접근 제어 등 서버 기능 모듈로써 구현된다. 그리고 모듈은 정적으로나 동적으로 서버에 링크된다.

아파치의 인증 모듈 중 기초 모듈은 mod_auth이며, mod_auth은 웹에서 사용자와 패스워드를 특정한 패스워드와 그룹 파일로 매치한다. mod_auth_dbm와 mod_auth_db 모듈은 데이터베이스에서 사용자들을 찾음으로써 더 큰 확장성을 제공한다. LDAP 디렉토리, Oracle, Mysql 데이터베이스와 커버로스 사용자를 인증하는데 사용 가능한 모듈이 있으며, 이런 스키마 전부는 사용자 이름과 패스워드가 평문으로 네트워크상에 전송된다. 또 다른 사용자 인증은 다이제스트 인증으로 mod_auth_digest에 의해서 구현되며, 이 방법은 잘 이용되지 않는다.

mod_ssl이 아파치 웹 서버로 추가될 때, 클라이언트와 서버 사이의 통신은 SSL 상위 계층인 응용프로토콜에서 한다. SSL의 일반적인 상거래 이용에서 서버는 식별자 인증서와 비공개 키 소유가 요구되며, 또한 암호화된 통신 채널을 설정하는데 이용된다.

SSL은 이것들을 모드에서 실행할 수 있으며, 클라이언트가 인증서의 제안을 요구하고 비공개키 소유에 대한 증명을 요구한다. 이 모드가 사용될 때, mod_ssl은 클라이언트 인증서에 기반한 접근 제어를 제공한다. 그리고 사용자 이름이 클라이언트의 X.509 인증서의 주체로 사용될 때 mod_ssl은 FakeBasicAuth 옵션을 구현할 수 있다. 반면 SSL 핸드쉐이크가 클라이언트 인증서를 검증한 이래 어떤 패스워드도 사용자에게서 얻게 되는 것을 요구하지 않는다. mod_ssl에서 SSLRequire는 그림 3-2와 같이 접근 허용을 위한 제약조건을 명시한다.

```
<Directory /foo>
  SSLRequireSSL
  SSLRequire  %{SSL_CLIENT_S_DN_O}  eq  "LBNL"
          and %{SSL_CLIENT_S_DN_OU}
              in {"DSD", "ICSD", "NERSC"}
</Directory>
```

그림 3-2. SSLRequire 지시어
Fig.3-2 Example of SSLRequire directive

3.3 권한 정책 모듈 설계 및 구현

본 논문에서 권한 정책 모듈 구현은 리눅스 기반 아파치 웹 서버를 사용하였으며, 권한 정책 모듈 구현으로 사용한 XML은 트리 형태의 데이터 구조로 이루어져 있다. 이와 같은 형태의 자료 구조를 접근하기 위해서 W3C에서는 DOM이라는 인터페이스를 채택하였다. DOM 인터페이스는 트리 조작에 필요한 각종 함수들을 제공함으로써 응용 프로그램이 쉽게 노드 탐색을 하도록 도와주며, XML 문서의 구조 변경 등과 같은 다양한 트리 조작이 가능하도록 해준다. 그림 3-3은 사용-조건 인증서의 XML 문서에 대한 DOM 트리를 보여준다.

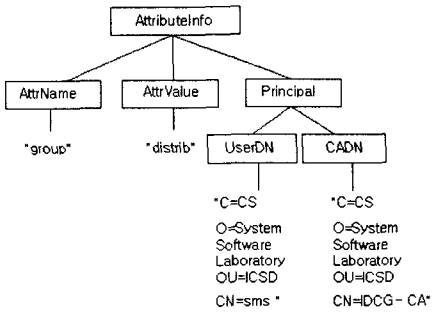


그림 3-3. 사용-조건 인증서에 대한 DOM 트리
Fig.3-3 DOM tree of Use-condition certificate

권한 정책은 정책 인증서, 사용-조건 인증서, 그리고 속성 인증서 등 세 부분으로 구성되며 정책 언어는 XML로 구현한다.

정책 인증서는 자원에 대한 기관의 소스를 명시함으로써 신뢰된 체인을 폐쇄하는데 사용되며, 사용-조건 인증서는 자원에 대한 접근을 제어하는 제약조건을 포함한다. 그리고 속성 인증서는 그림 3-4의 사용 제약조건의 만족을 요구하는 사용자에게 속성을 할당한다. 정책 인증서는 자원의 이름, 신뢰된 인증기관들의 리스트, 스테이크홀더들의 이름(또는 그룹), 속성 인증서 위치에 대한 선택 리스트를 포함한다. 그리고 사용-조건 인증서는 자원들에 적용하며 각각의 스테이크홀더는 적어도 하나의 사용-조건 인증서를 제공해야 하는데, 조건들은 Boolean으로 요구된 사용자 속성들의 정의를 표현한다. 또한 이런 조건들을 배경으로 매치한 속성 인증서들의 기관을 서명하며, 권리는 자원들에 적용한 동작들의 리스트이다. 사용자의 식별자 인증서 컴포넌트들은 CN=, O=, OU= 등이며, 정책 인증서에 정의되고, 사용자 속성 인증서들에 포함된 추가적인 파라미터들 역할 또는 그룹 등을 포함한다. 그리고

속성 인증서는 사용자에게 적용하는 사용자의 식별자(발급자 이름), 속성을 정의한 속성-값 쌍, 인증서의 주체가 정의된 속성의 소유를 주장한 사람이나 기관에 의한 전자 서명을 포함한다. 그리고 웹 환경에서 권한 정책이 서명된 XML 인증서에 포함되며 AAS 정책 엔진은 정책 인증서에 명시된 URL을 검색하고 각 인증서에 있는 발급자와 서명을 검증함으로써 모든 사용-조건을 찾는다. 만약 찾지 못하면 자원으로 접근은 부인된다.

```
<AttributeInfo Type="AAS">
 <AttrName>group</AttrName>
 <AttrValue>distrib</AttrValue>
 <Principal>
  <UserDN>/C=CS/O=System Software Laboratory
/OU=ICSD/CN=sms
  </UserDN>
  <CADN>/C=CS/O=System Software Laboratory
/OU=ICSD/CN=IDCG- CA
  </CADN>
 </Principal>
</AttributeInfo>
```

그림 3-4. 사용-조건 인증서
Fig.3-4 Use-condition certificate

권한 정책 모듈 mod_AAS는 세 개의 프로시저로 구성되는데 파일에서 문서를 메모리로 가져오는 처리를 하기 위한 인증서-기반 두개, 접근을 체크하기 위한 인증서-기반 하나를 정의한다.

mod_AAS는 아파치 모듈로서 웹 서버에 권한 자격들이 제공되며, 아파치 웹 서버에서 동작 한다. 또한 동적으로 공유된 오브젝트 모듈로 구현되며 스타트업이나 재시작 시간에 서버로 로드된다. 그리고 명시하지 않으면 웹 서버로의 모든 요청에 대하여 접근 제어 메커니즘으로 어떤 임의의 핸들러도 정의하지 않는다. 서버 구성에서 두개의 글로벌 지시와 검사 접근 콜백을 정의하는데, 권한 모듈 인터페이스는 디렉토리마다 구성, 커맨드 테이블, 검사 접근 루틴에 대한 콜백에 대한 호출로 구성된다.

mod_AAS는 안전한 웹 서버를 요구하며, 웹 서버는 mod_AAS를 호출하기 전에 클라이언트의 X.509 인증서를 인증한다. 또한 웹에서 접근 제어는 mod_AAS를 통하여 자유롭게 이용할 수 있다.

이와 같이 mod_AAS는 리눅스 기반의 아파치 웹 서버를 사용하였으며, 암호 라이브러리는 SSL을 사용하였다.

본 논문에서는 분산된 환경에서 AAS 모듈을 장착한 웹 서버 시스템과 non-AAS 시스템의 차이점을 나타내며, 웹 서버 접근에 있어서 1KB의 동일한 문서에 대하여 AAS 모듈이 장착된 웹 서버 시스템의 파일 패치 단계에서는 비효율적인 성향이 있었지만 패치 후 웹 서버에 접근자가 많을수록 현저한 차이점을 보인다. 표 1은 하나의 정책 인증서, 두개의 사용-조건 인증서, 두개의 속성 인증서, 그리고 네 개의 X.509 인증서로 구성된 일반적인 정책을 이용하여, 클라이언트, 웹 서버 그리고 인증서 서버들은 100 Mb/s LAN상에서 1KB 동일한 크기의 문서를 동일한 접근 횟수를 통하여 AAS/non-AAS 사용의 접근 횟수에 따른 패치 평균 시간을 비교한 것이다. 본 논문에서는 실험 환경에 따른 제약으로 분산 환경에서 많은 사용자 접근의 패치 평균 시간은 측정하지 못했으나, 위의 일반적인 정책을 이용한 측정에서 패치 때 이미 인증서들이 로드된 상태에서 많은 사용자 접근이 되므로 유리하다.

표 1. AAS 및 non-AAS 사용 패치 평균 시간
Table.1 Average Time to Document Fetch with and without AAS

Fetch \ Access Times	5	10	15	20
AAS	0.471	0.468	0.496	0.463
non-AAS	0.00385	0.00397	0.00413	0.00432

Ⅳ. 결 론

다양한 형태의 사용자 인증, 무결성, 부인 방지의 보안 서비스를 제공하는 PKI는 비대면한 상황에서 사용자 인증을 위해 좋은 해결책을 제시하여 주고 있지만 지역적으로 떨어져 있는 컴퓨팅 환경에서 권한에 대한 해결책을 제시하기에는 미흡한 것 또한 사실이다. 이러한 문제의 해결 방안으로 분산 환경에서 안전하고 신뢰성 있는 사용자의 권한을 제공하는 XML 기반 권한 정책 모듈 mod_AAS를 리눅스 기반 아파치 웹 서버를 사용하여 구현하였다.

본 논문에서는 인증 및 권한 정책 모듈 mod_AAS를

설계하여 분산 환경에서 접근제어에 대한 유연성 제공과 가상 조직의 광범위한 사용자의 식별자 사용, 자원 게이트웨이에서 원격 조정되는 독립적인 스테이크홀더의 접근 정책 설정을 용이하게 하였다.

이와 같이 권한 정책 모듈 설계 및 구현을 통하여 분산 환경에서 사용자와 자원 사이의 안전한 권한 서비스를 구축 하였으며, AAS 모듈 사용으로 분산 환경에서 보다 나은 권한의 성능 향상을 제시하였다. 향후 연구 방향으로는 개발한 AAS 모듈의 지속적인 보완이 필요하다.

참고문헌

[1] Kenneth G. Paterson, Geraint Price, "A comparison between traditional public key infrastructures and identity-based cryptography", Information Security Technical Report, Vol.8, No.3, pp. 57-72, 2003.

[2] Timothy G. Shoriak, "SSL TLS Protocol Enablement for key Recovery", Computer & Security, pp. 100-104, 2000.

[3] T.M. E A.G.A.J. Loeffen, "Succession in standardization: grafting XML onto SGML", Computer Standards & Interfaces, Vol.24, pp. 279-290, 2002.

[4] J. Fong, H.K. Wong and Z. Cheng, "Converting relational database into XML documents with DOM", Information and Software Technology, Vol.45, pp. 335-355, 2003.

[5] Chris Lesniewski-Lass, M. Frans Kaashoek, "SSL splitting: Securely serving data from untrusted caches", Computer Networks, Vol.48, pp. 763-779, 2005.

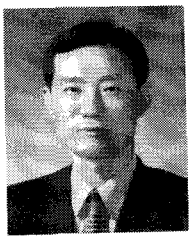
[6] Ryutov, Neuman, "Access control framework for distributed application", IETF, 2000.

저자소개

신명숙(Myeong-Sook Shin)

1992년 2월 광주대학교 전자계산학과(공학사)
1996년 2월 광주대학교 대학원 전자계산학과(공학석사)

2005년 12월 조선대학교 대학원 컴퓨터공학과 박사과정(현)
※관심분야 : 시스템소프트웨어, 유비쿼터스컴퓨팅, 정보보호



이 준(Joon Lee)

1979년 2월 조선대학교 전자공학과(공학사)
1981년 2월 조선대학교 대학원 전자공학과(공학석사)

1997년 2월 숭실대학교 대학원 전자계산학과 (공학박사)
1982년 3월 조선대학교 전자정보공과대학 컴퓨터공학부 교수(현)
※관심분야 : 운영체제, 정보보호, 유비쿼터스컴퓨팅