PKI 환경에서 인증서 기반 권한 정책에 관한 연구

신명숙*·송기범*·이준**

*조선대학교 대학원 컴퓨터공학과
**조선대학교 전자정보공과대학 컴퓨터공학부

e-mail:msshin@hanafos.com

A Study on the Certificate Based Authorization Policy in a PKI Environment

Myeong-Sook Shin* · Gi-beom Song* · Joon Lee**
*Dept. of Computer Engineering, Graduate School, Chosun University
**School of Computer Engineering, Chosun University

요 약

권한 정책은 정책 인증서, 사용-조건 인증서, 속성 인증서로 구성되며, 안전하고 신뢰성 있는 사용자의 권한을 제공한다. 다양한 형태의 사용자 인증, 무결성, 부인 방지의 보안 서비스를 제공하는 공개키 기반 구조는 비대면한 상황에서 사용자의 인증을 위해서 좋은 해결책을 제시하여 주고 있지만 지역적으로 떨어져 있는 컴퓨팅 환경에서 권한에 대한 해결책을 제시하기에는 미흡한 것 또한 사실이다. 따라서 본 논문에서는 분산 환경에서 분산된 사용자들이 사용할 수 있는 AAS 권한 모델을 제안하고, 리눅스 기반 아파치 웹 서버에서 AAS 모듈을 설계하였다.

1. 서론

공개키 기반 구조(PKI:Public Key Infrastructure)는 전자상거래, 네트워크 보안등 다양한 응용분야에서 X.509 기반으로 구축 적용 발전시켜 나가고 있다. 인터넷과 같이 네트워크에 연결된 각 사용자 메시지에 대한 인증과 더불어 사용자들의 권한 서비스의 중요성이 증가함에 따라 중요한 요구사항으로 인식되고 있으며, 공개키 기반 구조에서 FPKI(Federal Public Key Infrastructure), ICE-TEL(Internetworking Public Key Certification Infrastructure), GOCPKI(Government of Canada PKI), PKAF(Public Key Authentication Framework) 등 선진 국가들은 정보 서비스 산업 경쟁력 강화를 위하여 공개키 기반 구조 연구를 지속적으로 수행하고 있다[1,2,3].

다양한 형태의 사용자 인증, 무결성, 부인 방지의 보안 서비스를 제공하는 공개키 기반 구조는 비대면한 상황에서 사용자의 인증을 위해서 좋은 해결책을 제시하여 주고 있지만 지역적으로 떨어져 있는 컴퓨팅 환경에서 권한에 대한 해결책을 제시하기에는 미흡한 것 또한 사실이다. 이러한 이유로 AAS(Authentication Authorization System) 권한 모델이 제안된다.

권한 시스템은 가상 조직의 광범위한 사용자의 식별자 사용과 실제 자원 게이트웨이로부터 원격 조정되는 독립적인 스테이크홀더가 접근 정책 설정을 용이하게 하는 것이며, 사용하기 쉬운 권한 서비스를 제공하는 것으로 협력과 계산 그리드 요구에 대응한다. 협력과 그리드의 다른 특성은 자원에 대한 접근 제어를 요구하며 자원 게이트웨이로부터 독립적, 원격 정의, 접근 정책을 허용한 권한 시스템이 바람직하다[1,2,3,4].

본 논문은 분산 환경에서 분산된 사용자들이 사용할 수 있는 권한 시스템을 개발하기 위하여 공개키 기반 구조 기본 구성 객체와 권한 정책에 대하여 살펴보았으며, 리눅스 기반 아파치 웹 서버에서 권한 모듈을 설계한다.

2. 인증 및 권한정책

2.1 인증

인증은 시스템을 보호하기 위해 서버가 사용자에게 사전에 약속된 정보를 제시 할 것을 요구함으로써 사용자를 확인하는 것이며, 본 논문에서 제안된 AAS 운용을 위해 요구되는 사항들로써, PKI와 X.509 인증서가 있다. 공개키 기반 구조의 기본 구성은 (그림 2-1)과 같으며 인증서발급, 사용 및 취소와 관련된 서비스를 제공한다. 공개키 기반 구조 환경을 구축하는 주요 객체는 인증기관(CA:Certification Authority), 인증서 저장소, 최종 사용자 그리고 e-commerce 서비스 제공자로 구성된다.

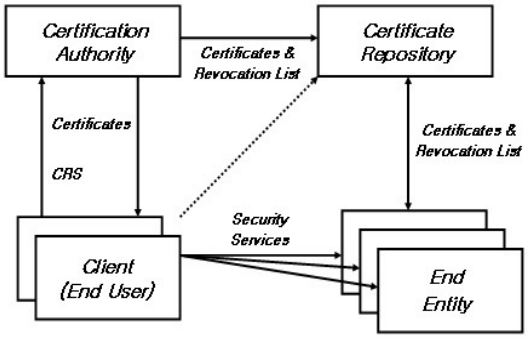

그림 2-1. 공개키 기반 구조 기본 구성 객체

공개키 기반 구조에서 사용되는 X.509v3 인증서 형식은 (그림 2-2)에서와 같이 인증서의 발급 대상이 되는 객체를 식별하여 공개키를 연관시킬 수 있는 표준적인 정보들이 포함된다[2,4].

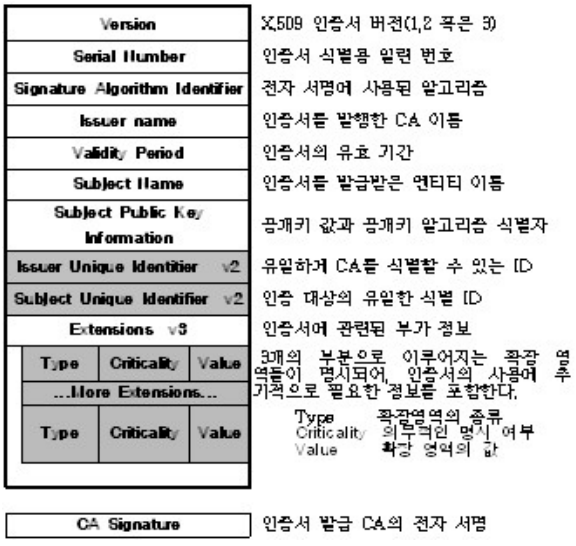

그림 2-2. X.509v3 인증서 형식

2.2 권한

권한은 인증 절차 후 제공된 토큰을 기반으로 자원의 사용자 접근 권한을 판별하여 허가하는 것으로써 최근 인증과 함께 중요한 보안 요구사항으로 인식되고 있다.

권한 시스템 방식은 pull 모델과 push 모델이 있는데, pull 모델은 속성 인증서가 생성되었을 때 디렉토리에 속성 인증서를 제시하는 방식이다. 따라서 속성 인증서를 사용하는 응용 서비스는 속성 인증서가 필요할 때 디렉토리에서 인증서를 검색하여 사용한다. 반면 push 모델은 사용자가 응용 서비스에 접근할 때 속성 인증서를 직접 전달하는 방식으로, 이 방식은 사용자가 응용 서비스에 접근할 때 사용자 이름과 패스워드를 전달하는 것과 같은 방식이다. pull 모델은 클라이언트 또는 클라이언트/서버 프로토콜의 변경 없이 구현 될 수 있다는 장점을 가지고 있으며 클라이언트의 권한이 서버 도메인 내에서 할당되어야만 하는 경우에 특히 적합한 모델로써 AAS에서 사용하고 있다[2,3].

2.3 SSL(SecureSocketLayer)

SSL은 응용 프로토콜과 TCP/IP사이에 위치하며 데이터의 암호화, 서버의 인증, 메시지의 무결성을 제공하며, 서버에 대한 인증은 반드시 수행되지만 클라이언트에 대한 인증은 반드시 선택적으로 수행할 수 있도록 해준다. SSL은 서버와 클라이언트 양쪽의 TCP/IP 연결을 위해서 핸드쉐이크 프로토콜을 수행하여 양쪽은 암호화 통신에 합의하고, 암호화 통신과 인증에 필요한 값들을 초기화한다. 초기화 후, SSL은 응용 프로토콜에서 생성해 낸 바이트 스트림의 암호화와 복호화를 수행하게 된다. 즉 HTTP 요청과 HTTP 응답에 포함되는 모든 정보들이 암호화되어 전송됨을 의미한다. (그림 2-3)은 핸드쉐이크 프로토콜의 수행 과정을 보여준다[3,4].

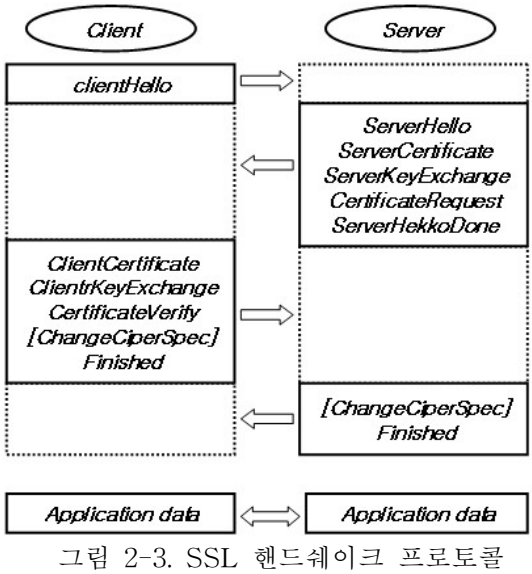

그림 2-3. SSL 핸드쉐이크 프로토콜

3. 웹기반인증및권한정책모듈설계

3.1 AAS모델설계

AAS는 전자 서명된 인증과 정책 인증서, 사용자 속성 인증서, 자원 사용-조건 인증서들인 권한에 기

초한다[5,6,7].

AAS 모델은 (그림 3-1)에서와 같이 사용자 요청에 따라 자원으로 접근되며 인증한다. 그리고 나서 자원 게이트웨이는 AAS를 접속한 후 정책 인증서에 위치되며 자원의 사용 조건 인증서들을 모은다. 그리고 AAS 정책 엔진은 접근 제어 결정을 한다.

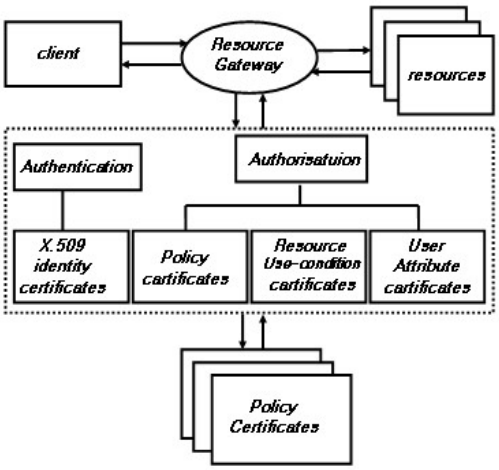

그림 3-1. AAS 모델

3.2 웹 인증 설계 구현

분산 환경에서 사용자 접근 권한 시스템을 개발하기 위하여 공개키 기반구조의 아파치 웹 서버 인증 및 권한 모듈을 설계한다[5,6,7].

표준 웹 인증과 접근 제어는 요청이 일어난 도메인에서 사용자가 자신의 이름과 패스워드를 제공하면, 웹 브라우저가 서버 머신에 저장된 사용자 정보와 대조하여 메치하는 인증에 기반한다.

이를 위해서 기존의 아파치를 이용하여 모듈이 구현되는데, 아파치 웹 서버가 광범위하게 사용되며 높은 성능의 프리웨어 서버는 API(Application Program Interface)로 만들어진다. 서드 파티 프로그래머가 새로운 서버 기능성을 추가하며, 서드 파티로 동일하게 사용 가능한 확장 API를 사용하면서 인증, 접근 제어 등 서버의 기능인 대부분이 모듈로써 구현된다. 또한 모듈은 정적으로나 동적으로 서버에 링크된다.

아파치의 인증 모듈 중 기초 모듈은 mod_auth이며, mod_auth은 웹에서 사용자와 패스워드를 특정한 패스워드와 그룹 파일로 메치한다. 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 패스된다. 또 다른 사용자 인증은 다이제스트 인증인데 mod_auth_digest에 의해서 구현되며, 이것은 대체로 잘 이용되지 않는다.

mod_ssl이 아파치 웹 서버로 추가될 때, 클라이언트와 서버 사이의 통신은 SSL의 위 계층인 HTTP가 한다. SSL 일반적인 상거래 이용에서 서버는 식별자 인증서와 비공개 키 소유가 요구되는데, 그것들은 암호화된 통신 채널을 설정하는데 이용된다.

SSL은 모드에서 실행할 수 있으며, 그것은 클라이언트가 인증서의 제안을 요구하고 비공개키 소유에 대한 증명을 요구한다. 이 모드가 사용될 때, mod_ssl은 클라이언트 인증서에 기반한 접근 제어를 제공할 수 있다. 그리고 사용자 이름이 클라이언트의 X.509 인증서의 주체(subject)로 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 3-2)에서 처럼 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

3.3 웹 권한 정책 모듈

권한 정책은 정책 인증서, 사용-조건 인증서, 그리고 속성 인증서 등 세 부분으로 구성하여 XML로 구현한다[5,6,7].

정책 인증서는 자원에 대한 기관의 소스를 명시함으로써 신뢰된 체인을 폐쇄하는데 사용되며, 사용-조건 인증서는 자원에 대한 접근을 제어하는 제약조건을 포함한다. 그리고 속성 인증서는 (그림 3-3)의 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 표현하며, 이런 조건들을 배경으로 메치한 속성 인증서들의 기관을 서명, 권리들은 자원들에 적용한 동작들의 리스트이다, 도한 사용자의 식별자 인증서 컴포넌트들은 CN=, O=, OU= 등이며, 정책 인증서에 정의되고, 사용자 속성 인증서들에 포함된 추가적인 파라미터들 역할 또는 그룹 등을 포함한다.

```
<AttributeInfo Type="AAS">
  <AttrName>group</AttrName>
  <AttrValue>distrib</AttrValue>
  <Principal>
    <UserDN>/C=US/O=Lawrence Berkeley National
          Laboratory /OU=ICSD/CN=Strilekha
    </UserDN>
    <CADN>/C=US/O=Lawrence Berkeley National
          Laboratory /OU=ICSD/CN=IDCG- CA
</CADN>
  </Principal>
</AttributeInfo>
```

그림 3-3. 사용-조건 인증서

그리고 속성 인증서는 속성 인증서가 사용자에게 적용하는 사용자의 식별자(발급자 이름), 속성을 정의한 속성-값 쌍, 인증서의 주체가 정의된 속성의 소유를 주장한 사람이나 기관에 의한 전자 서명을 포함하며, 웹 환경에서 권한 모듈을 설계하여 보다 편리하고 안전한 권한을 제공하고자 한다.

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

AAS 모듈 작업은 아파치 모듈로서 웹서버에 권한 자격들을 제공하며, 아파치 웹서버에서 동작 가능하다. 또한 동적으로 공유된 오브젝트 모듈로써 구현 가능하며 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, 커멘드 테이블, 검사 접근 루틴에 대한 콜백에 대한 호출로 구성되고, 두개의 글로벌 지시는 다음과 같다.

- AASConf : 정책 엔진을 구성하기 위해서 사용한 구성 파일의 이름을 공급한다.
- AASResources : 문서 트리의 어떤 부분을 제어할 수 있도록 명시하는데 이용한다.

AAS 모듈은 안전한 아파치 웹 서버를 요구하며, 웹 서버는 AAS 모듈을 호출하기 전에 클라이언트의 X.509 인증서를 인증한다. 또한 웹에서 접근 제어는 아파치 모듈인 AAS 모듈을 자유롭게 이용하여 달성할 수 있다.

4. 결론

공개키 기반 구조는 비대면한 상황에서 사용자 인증을 위해서는 좋은 해결책을 제시 하여주고 있지만 분산 환경에서 사용자들이 사용할 수 있는 권한에 대한 해결책을 제시하기에는 미흡한 것 또한 사실이다.

위와 같은 문제점을 해결하기 위해 본 논문에서는 웹을 근간으로 한 공개키 기반 구조 환경에서 X.509 공개키 인증서를 사용한 인증 서비스와 권한 정책 모듈인 AAS 모듈을 설계하였다. 이러한 AAS 권한 모듈은 기존의 한계를 벗어나 많은 이점을 제공하여 준다.

첫째, 가상 조직의 광범위한 사용자의 식별자 사용이며, 둘째, 자원 게이트웨이에서 원격 조정되는 독립적인 스테이크홀더가 접근 정책 설정을 용이하게 한다.

이와 같이 본 논문에서 AAS 모듈을 제안하고 AAS 모듈 설계를 통하여 분산 환경에서 사용자와 자원 사이의 안전한 권한 서비스를 구축하도록 하였으며, 앞으로 AAS 모듈 구현과 함께 이와 관련된 많은 연구가 이루어지기를 기대한다.

참고문헌

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

[2] J.J. Hwang, K.C. Wu, D.R Liu, "Access control with role attribute certificate", computer standards & Interfaces, Vol 22, pp43~53, 2000

[3] Farrell, Housley, "An Internet attribute certificate profile for authorization", draft-ietf-pkix-ac509prof-09.txt, June 2001

[4] Jamie Lewis, "Public Key Infrastructure Architecture", The Burton Group Network Strategy Overview, July 1997

[5] Apache. 2002a. Apache software foundation. http://www.apache.org.

[6] Apache. 2002b. Apache module registry. http://modules.apache.org/.

[7] Apache. 2002c. Apache XML project. http://xml.apache.org.