웹 서비스에서의 권한 기반 접근 제어 시스템 설계 및 구현

김경남*, 유황빈*

*광운대학교, 컴퓨터과학과

Design and Implementation of Privilege Based Authorization System for Web Services

Kyung-nam Kim*, Hwang-bin Ryou*

*Department of Computer Science Kwangwoon Univ.

요 약

웹 서비스는 XML 기술을 기반으로 분산 컴퓨팅을 가능하게 함으로써, 웹을 통한 시스템 통합이 용이하고, CORBA, Java RMI, DCOM등과 같은 기존의 분산 컴퓨팅 모델을 대체 할 수 있는 새로운 대안으로 주목 받고 있다. 현재 웹 서비스 활성화에 있어 최대의 당면 과제는 보안 문제이며, 인터넷을 이용해서 이동하고 있는 많은 양의 데이터를 안전하게 지킬 수 있는 방법에 주로 초점이 맞추어져 있으나, 이러한 데이터의 안전한 교환 이외에도 웹 서비스 제공자와 이용자의 상호 인증 및 사용자에 대한 접근 제어 방안 또한 보안상 중요한 사항이다. 본 논문에서는 웹 서비스 환경에서 클라이언트에 대해 효과적으로 권한 기반 접근 제어를 수행하기 위한 시스템을 설계 및 구현하였다.

I. 서론

웹 서비스는 표준화된 XML 메시지를 통해 네트워크 상에서 접근 가능한 연산들의 집합을 기술하는 인터페이스로 정의된다. 기존의 웹은 HTTP, HTML, URL과 같은 기술을 통해 인터넷상에 분산되어 있는 정보 자원들에 대하여 표준화된 접근과 정보 표현 방법을 제공함으로써 여러 업체와 사용자들의 지지를 받으면서 많은 발전을 이루어 왔다. 현재에는 XML의 출현으로 인하여 웹상에서 구조화된 데이터의 전달이 가능하게 되었으며, URI를 통하여 웹상의 객체들에 대한 단일화된 접근이 가능하게 되었다. 웹 서비스는 현재의 XML 기술을 기반으로 기존의 웹 환경을 이용한 분산 컴퓨팅을 가능케 함으로써 웹을 통한 시스템 통합을 용이하게 하며, 이러한 특징으로 인하여 CORBA, Java RMI, DCOM 등과 같은 기존의 분산 컴퓨팅 모델을 대체 할 수 있는 새로운 기술로써 주목을 받고 있다[9].

웹 서비스가 활성화되기 위해서 본질적으로 해결해야 할 문제점들 중에 하나가 바로 웹 서비스 보안 문제이다. 웹 서비스가 이루어지기 전에 웹 서비스 클라이언트와 웹 서비스 제공자간의 상호 인증, 비밀성, 무결성, 부인방지, 접근제어 등의 보안 기능이 만족되어야 한다. 서비스 이용자와 웹 서비스 사이의 메시지 교환에 있어, 교환되는 메시지의 안전성 보장을 위해 XML 전자서명, XML 암호화 등의 기술을 이용할 수 있으며 전송 계층 레벨의 보안 서비스인 SSL/TLS또는 IPSec-VPN 등을 사용함으로써 전송되는 XML 메시지에 대한 안전성이 보장될 수 있다. 그러나 XML 전자서명, XML 암호화, SSL/TLS, IPSec-VPN 모두 접근제어 기능은 구체적으로 제공하고 있지 못하다.

본 논문에서는 웹 서비스 환경에서 클라이언트에 대해 효과적으로 권한 기반 접근 제어를 수행하기 위한 방안을 제안한다. 서비스 공개 및 검색 단계에서의 보안 요구사항 기술을 위해 WSDL 문서를 확장하였으며, SOAP 요청 메시지에 신원 및 권한 정보를 표현 할 수 있도록 헤더 요소들을 정의하였다. 또한 클라이언트의 접근 권한을 판단하

는 웹 서비스인 AA 서버와 SOAP 게이트웨이 내부에 위치하여 클라이언트의 접근을 제어하는 접근 제어 필터를 설계 및 구현하였다.

II. 웹 서비스 보안 기술

1. XML 전자서명

XML 전자서명(XML Signature)은 XML을 비롯한 다양한 형태의 전자문서에 대해 XML 형태의 전자서명을 생성하고 검증할 수 있는 XML 기반의 전자서명 기법으로 전자문서에 대해 인증, 무결성, 부인방지 등의 정보보호 서비스를 제공해 준다. XML 전자서명은 XML을 비롯한 다양한 디지털 컨텐츠에 대해 적용 가능하며 하나 혹은 그 이상의 리소스들에 대해 서명을 할 수 있다[4, 7].

2. XKMS

XKMS(XML Key Management Specification)는 XML 키 관리 명세로서 암호 기능이 있는 XML 애플리케이션을 인증하기 위한 포괄적이고 개방적이며 표준적인 접근방식을 취한다. 전자서명 및 공개키 암호화를 사용하는 XML 기반 애플리케이션에서 PKI와의 연동이 용이하도록 공개키 관리를 위한 프로토콜을 정의한다. 주요 목적은 전자서명을 검증하거나 데이터를 암호화하기 위해 사용되는 공개키 사용자에게 필요한 키의 위치를 명시하고, 이름이나 속성 정보를 해당 비밀키 소유자와 관련지어 주는 것이다[18].

3. XACML

XACML(eXtensible Access Control Markup Language)은 정책 언어와 접근 제어 판단을 위한 요청/응답 언어를 정의한다. 정책 언어는 일반적인 접근 제어 요구사항 그리고 새로운 기능, 데이터 형식과 그것들을 연계할 수 있는 로직등을 기술하는데 사용된다. 요청/응답 언어는 클라이언트의 요청의 허용 여부에 대한 질의에 응답하며, 결과를 해석한다. 응답 메시지는 클라이언트의 요청의 허용 여부를 나타내며 Permit, Deny, Indeterminate, Not Applicable로 구분된다.[16]

III. 시스템 설계

1. WSDL 확장

WSDL(Web Services Description Language)은 서비스 제공자가 각기 다른 프로토콜과 인코딩 방법 위에서 웹 서비스 요청에 대한 기본적인 형식을 기술하는 방법을 제공한다. WSDL은 웹 서비스의 기능, 위치, 호출 방식 등에 대한 내용만을 기술하고 있을 뿐 웹 서비스를 이용하는데 필요한 보안 요구사항들은 기술하지 않고 있다.

본 논문에서는 WSDL에 웹 서비스의 보안과 관련된 사항들을 추가적으로 기술하여 클라이언트가 서비스 검색 단계에서부터 보안요구사항을 고려하여 보안 서비스의 적용을 용이하게 하였다.

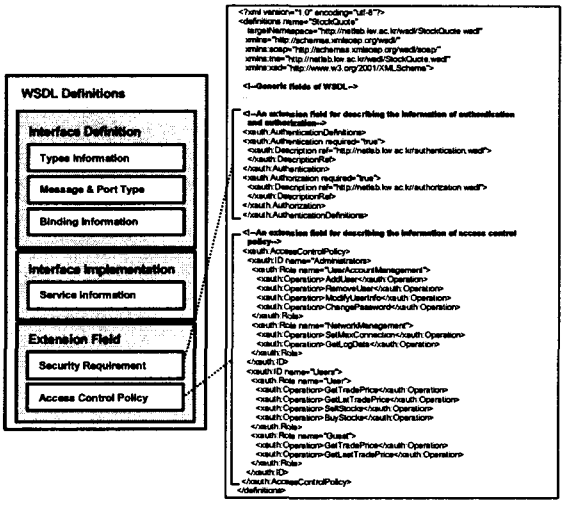

그림 2: 확장된 WSDL의 구조와 예

2. SOAP 헤더 정의

SOAP(Simple Object Access Protocol)은 XML로 인코딩한 데이터를 전송하는 일정한 방법을 정의한 표준 프로토콜이다. SOAP은 HTTP를 기반으로 원격 프로시저 호출을 수행하는 방법 역시 정의하고 있다. SOAP 메시지는 헤더와 바디로 구성된다. 헤더에는 메소드 호출과 관계없는 보조 정보들을 기술하는데 사용되며 필요에 의해 새로운 요소들을 정의할 수 있다.[12]

```
<soap:Header>
 <xauth:credentials soap:mustUnderstand="1">
  <xauth:PublicKeyCertificateInfo>
   <issuer>
    CN=XKMS Server, OU=Netlab, O=Kwangwoon Univ.,
    L=Seoul, C=kr
   </issuer>
   <serialNumber></serialNumber>
  <xauth:PublicKeyCertificateInfo>
  <xauth:AttributeCertificateInfo>
   <issuer>
    CN=AA Server, OU=Netlab, O=Kwangwoon Univ.,
    L=Seoul, C=kr
   </issuer>
   <serialNumber></serialNumber>
  <xauth:AttributeCertificateInfo>
 </xauth:credentials>
</soap:Header>
```

그림 2: SOAP 헤더

본 논문에서는 사용자 인증 및 인가에 필요한 정보를 기술하기 위해 SOAP 헤더에 credentials 라는 요소를 정의하였다. credentials요소에는 인증에 필요한 공개키 및 속성 인증서의 일련번호와 발급자등에 대한 정보가 기술된다.

3. 접근 제어 필터 설계

접근 제어 필터는 실제적인 접근 제어를 시행하는 PEP(Policy Enforcement Point) 역할을 한다. 클라이언트의 요청에 대한 접근 제어는 서비스 호출 모듈 이전에 이루어져야하므로 접근 제어 필터는 SOAP 요청 분석 모듈과 서비스 호출 모듈 사이에 위치한다.

SOAP 요청 메시지의 헤더에 기술되어있는 클라이언트의 인증 및 인가에 필요한 정보를 분석하여 접근을 허용하거나 거부하게 된다. 접근을 허용할 경우 해당 요청을 서비스 호출 모듈로 넘기게 되고 거부할 경우 SOAP 오류를 생성하여 클라이언트에 전송하게 된다. 접근 제어 필터 이용 시 이미 구현된 서비스를 수정하지 않고도 접근 제어를 수행할 수 있다. 그림 3은 접근 제어 필터의 구조 및 동작 과정을 나타낸 것이다.

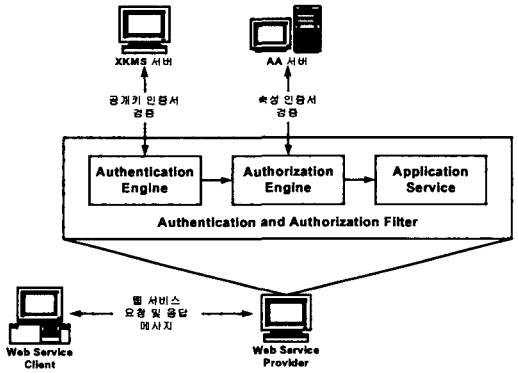

그림 3: 접근 제어 필터의 구조 및 동작

4. AA 서버 설계

AA 서버(Attribute Authority Server)는 서비스 준비 단계에서 클라이언트에 속성 인증서를 발급해주는 역할을 하며, 서비스 이용 단계에서 클라이언트의 접근 권한을 판정하는 PDP(Policy Decision Point) 역할을 한다. 서비스 제공자는 UDDI에 서비스 공개 시 AA서버에 대한 사항과 접근 제어 정책에 대한 정보를 WSDL에 추가적으로 기술한다. 그림 4는 AA 서버를 통해 웹 서비스를 이용하려는 클라이언트에 대한 접근 제어를 수행하는 과정을 나타낸다.

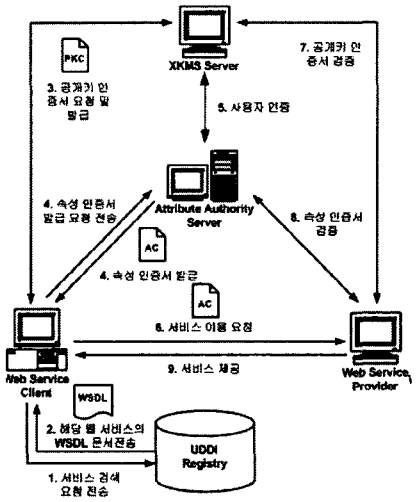

그림 4: AA 서버의 구조 및 동작

1) 클라이언트는 UDDI를 통해 자신에게 필요한 웹 서비스를 검색하고, UDDI부터 해당 웹 서비스의 WSDL 문서를 수신한다. 이 때 전달받는 WSDL 문서는 확장된 WSDL 문서로 기본적인 웹 서비스의 인터페이스 기술 이외에 사용자 인증과 인가에 필요한 부가 정보를 포함하고 있다.

2) 공개키 인증서를 발급 받지 못한 경우에는 WSDL 문서에 기술되어 있는 XKMS 서버의 위치 정보를 통해 XKMS 서버로부터 공개키 인증서를 발급 받는다.

3) WSDL 문서에 기술되어 있는 AA 서버의 위치 정보를 통해 AA 서버에 접근할 수 있으며, 속성 인증서를 발급 받지 못한 경우 AA 서버로부터 속성 인증서를 발급 받는다.

4) 웹 서비스 클라이언트가 AA 서버로부터 속성 인증서를 발급 받기 위해서는 인증을 받아야 하는데, 전자 서명 을 이용한다. AA 서버는 웹 서비스 클라이언트의 공개키 인증서의 검색 및 유효성 검증은 XKMS 서버에 접속하여 수행한다.

5) 클라이언트는 웹 서비스 제공자에 서비스를 요청하며, 서비스 이용 시 요구되는 인증 및 인가를 위해, SOAP 요청 메시지에 속성 인증서의 holder 필드와 issuer 필드를 포함하여, 웹 서비스 제공자에 전송한다.

6) 웹 서비스 제공자는 XKMS 서버에 접속한 뒤 holder 필드 값을 이용해 공개키 인증서를 검색하고 유효성을 검증하여 클라이언트를 인증한다.

7) 웹 서비스 제공자는 AA 서버를 통해 클라이언

트의 속성 인증서를 검색하고, 유효성을 검증한다.

8) 속성 인증서가 유효하면, 속성 인증서 내의 속성 정보를 검색하여, 웹 서비스 클라이언트의 Service Authentication Informations, Access Identity, Roles, Clearance 정보 등을 가지고, 권한 기반의 접근 제어를 수행한다.

5. 속성 인증서 프로파일 정의

본 논문에서는 XML 기반의 인증서 프로파일을 정의 하였다. XML 기반의 프로파일을 사용하여 XML 기반인 웹 서비스에서의 속성 인증서의 이용을 용이하도록 하였다. 표 1은 본 논문에서 사용하는 속성 인증서의 프로파일을 나타낸 것이다.

표 1: 속성 인증서 프로파일

필드	설명
version	속성인증서의 버전
holder	속성인증서의 소유자
issuer	속성인증서의 발행자
signature	속성인증서에서 사용되는 전자서명 알고리즘
serialNumber	속성인증서 일련번호
attrCertValidity Period	속성인증서 유효기간
attributes	속성 정보들
extensions	확장 필드들

■ 속성 정보 : 본 논문에서 사용되는 속성 인증서의 속성 정보 필드는 Service Authentication Infos, Access Identity, Roles, Clearance 이다.

■ 확장 필드 정보 : 본 논문에서 사용되는 속성 인증서의 확장 필드는 Authority Key Identifier, No Revocation Available 필드이다. Authority Key Identifier 필드는 AA 서버의 공개키 인증서의 Subject Key Identifier 필드의 값과 동일하게 설정하며, No Revocation Available 필드는 Null로 설정한다.

그림 5는 본 논문에서 제안한 시스템에서 사용되는 속성 인증서의 예를 나타낸다.

그림 5: 속성 인증서 예

IV. 구현

본 논문에서는 접근 제어를 수행하기 위한 접근 제어 필터와 사용자의 접근 권한을 판정하기 위한 AA 서버 그리고 권한 기반 접근 제어 시스템을 적용하기 위한 웹 서비스인 Horoscope 서비스를 구현하였다. 전체 시스템은 모두 JDK 1.4.2 버전을 이용하여 자바로 구현하였다. 웹 서비스 제공자 및 AA 서버에 사용되는 웹 서버로는 Tomcat을 사용하였으며, SOAP 엔진은 Apache-SOAP 사용하였다. 정책 정보를 저장하기 위한 디렉토리 서버로는 openLDAP 사용하였다. 서비스 이용 요청 시 SOAP 헤더에 서명을 하기 위해 필요한 XML 전자서명 라이브러리로는 Neudist의 xmlsig 라이브러리를 사용하였다.

접근 제어 필터는 Apache-SOAP에서 SOAP 게이트웨이 역할을 하는 rpcrouter 서블릿을 수정하여 구현하였다. rpcrouter 서블릿의 SOAP 요청 메시지 분석 모듈과 서비스 호출 모듈 사이에 접근 제어 필터 기능을 구현하여 접근 권한이 없는 메시지를 필터링 할 수 있게 하였다. 접근 제어 필터는 SOAP 요청 메시지의 헤더 부분에 기술된 credentials 요소에 있는 정보들을 이용하여 AA 서버에 접근 권한 판정 요청을 전송한 뒤 접근 허용 여부에 대한 응답을 받게 된다. 접근이 허용된

경우에만 SOAP 요청을 서비스 호출 모듈로 보내 호출 결과를 클라이언트에 전송하며, 그렇지 않은 경우에는 서비스를 호출하지 않고 SOAP 오류를 생성하여 접근이 거부되었음을 클라이언트에게 통보한다.

AA 서버는 자바를 이용하여 구현하였으며 Apache-SOAP에 서비스로 등록하여 하나의 독립된 웹 서비스로 동작하도록 하였다. 서비스 준비 단계에서 클라이언트에 대한 인증서 발급 기능과 서비스 동작 단계에서 클라이언트에 대한 정보를 통해 접근 권한을 판정하는 기능을 한다. 접근 권한은 서비스 준비 단계에서 서비스 제공자가 WSDL에 기술한 접근 제어 정책 정보를 디렉토리 서버에 저장하여 사용한다.

그림 6와 그림 7은 Horosocpe 서비스에서 클라이언트의 접근이 허용되고 거부되는 각 경우에 대한 SOAP 응답 메시지를 TCP Tunnel/Monitor 도구를 이용해 저장한 화면이다.

접근이 허용되었을 SOAP 응답 메시지의 getHoroscopeResponse라는 요소를 통해 서비스 호출에 대한 결과를 확인할 수 있다. 하지만 접근이 거부되는 경우에는 SOAP 오류 메시지가 전송되며 faultcode와 faultstring이라는 요소를 통해 오류 코드와 오류에 대한 설명을 확인할 수 있다.

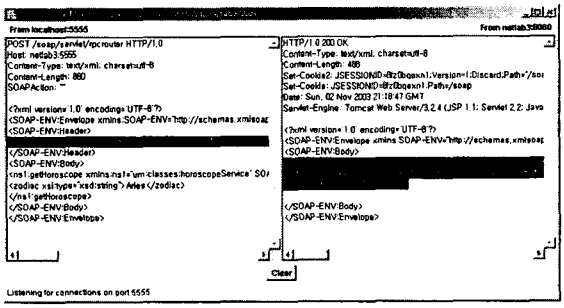

그림 6: 접근이 허용되는 경우의 SOAP 메시지

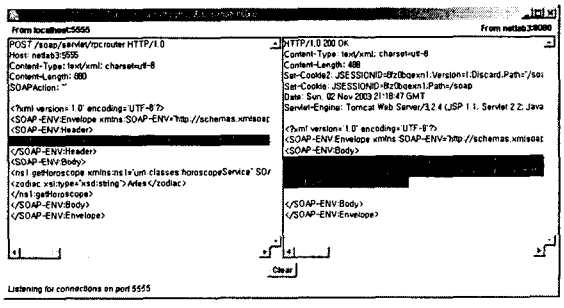

그림 7: 접근이 거부되는 경우의 SOAP 메시지

V. 결론

본 논문에서는 웹 서비스 환경에 보다 적합한 접근 제어 서비스를 제공하기 위해 권한 기반 접근 제어 시스템을 설계하고 구현하였으며, 웹 서비스 공개 및 검색 단계에서 보안 서비스에 대한 요구 사항을 기술하고 확인할 수 있도록 관련기술들을 확장하였다.

권한 기반 접근 제어 시스템을 위해 접근 제어 기능 수행을 위한 접근 제어 필터를 구현하였으며 접근 제어 정책 정보를 통해 클라이언트의 접근 권한을 판정하기 위한 AA 서버를 구현하였다. 웹 서비스 공개 및 검색 단계에서 보안 서비스에 대한 요구 사항 기술을 위해 WSDL을 확장하였다. WSDL 확장 영역에는 사용자 인증 및 인가에 필요한 정보와 해당 서비스에 대한 접근 제어 정책 정보를 기술할 수 있도록 하였다. 또한 서비스 요청 시 SOAP 요청 메시지에 사용자 인증 및 인가에 필요한 정보를 첨부하기 위해 SOAP 헤더에 관련 요소들을 정의하였다.

WSDL문서에 보안 서비스에 대한 요구사항을 기술함으로써 접근 제어 서비스를 위한 준비 과정을 용이하게 하였으며, 접근 제어 필터를 이용하여 기존에 제공되던 서비스에 수정을 가하지 않고도 접근 제어 기능을 추가 할 수 있게 하였다. 이를 통해 서비스 준비 단계에서의 서비스 제공자 및 클라이언트의 편의성을 향상시켰다.

또한 SOAP 헤더에 포함되는 사용자 정보로 인증서의 일련번호만을 사용함으로써 인증서 전체를 전송하는 PUSH 방식에 비해 네트워크 부하를 줄여 서비스 호출이 빈번한 웹 서비스 환경에 적합하게 하였다. 서비스 이용자의 권한 표현에 XML 기반의 인증서를 사용하여 기존의 ASN.1 형식의 인증서를 이용할 때의 디코딩 과정을 필요로 하지 않으며, 권한 판정 과정에서 XML 형식의 인증서를 그대로 사용함으로 사용자 권한 정보를 권한 판정을 위해 특정 형식(XACML-context)으로 변환해야하는 XACML에 비해 처리 속도를 향상하였다.

향후 연구 과제로는 보다 효율적인 클라이언트 접근 권한 판정을 위해 접근 제어 정책 테이블의 구조를 개선해야 하며, 접근 제어 필터에 캐쉬를 두어 빈번한 요청에 대해서는 AA 서버에 접근 권한 판정을 요청을 하지 않고 접근 제어 필터에서 바로 처리를 할 수 있도록 한다면 처리속도를 향상시킬 수 있을 것이다.

참고문헌

[1] A. Selkirk, "Using XML security mechanisms", *BT Technol Journal*, July, 2001.

[2] American National Standard for Financial Services X9.45, "Enhanced Management Controls Using Digital Signatures and Attribute Certificates", February 1999.

[3] Ariba Inc., Microsoft Corp. "UDDI Technical White Paper", uddi.org, September 2000.

[4] Blake Dournaee, "XML Security", RSA Press, 2002.

[5] Bloor Research, "Web Services Gotchas", July, 2002

[6] C.Francis and D. Pinkas, "Attribute Certificate Policy extension", IETF PKIX Internet-Draft, April 2003.

[7] Donald E. Eastlake 3rd, et al., "XML-Signature Syntax and Processing", IETF RFC3275, March, 2002.

[8] Eric Armstrong, et al, "The Java Web Services Tutorial", February, 2003

[10] ISO/IEC 9594-8/ITU-T Recommendation "Information Technology Open System Interconnection : The Directory : Public-key and attribute certificate Framework", October 2001.

[11] Mark O'Neill, "Securing Web Services", March, 2002.

[12] Martin Gudgin, et al., "SOAP Version 1.2 Part 1: Messaging Framework", Working Draft, W3C, June 2002.

[13] MS, IBM, "Security in a Web Services World: A Proposed Architecture and Roadmap", Aprial, 2002.

[14] Roberto Chinnici, et al., "Web Services Description Language (WSDL) Version 1.2", Candidate Recommendation, W3C, December 2002.

[15] S. Farrell and R. Housley, "An Internet Attribute Certificate Profile for Authorization", RFC 3281, April 2002.

[16] Simon Godik, et al., "eXtensible Access Control Markup Language, " OASIS Standard, February 2003.

[17] Todd Sunsted, "Building Security into Web Services", August, 2001.

[18] World Wide Web Consortium "XML Key Management Specification(XKMS) Version 2.0" W3C Working Draft 18 April 2003.