XML 서명을 이용한 접근 제어 모델

오흥룡*, 염흥열*

*순천향대학교 정보보호학과

Access Control Model using XML signature

Heung-Ryong Oh*, Heung-Youl Youm*

*Department of Information Security SoonChunHyang Univ.

요 약

정보통신 기술의 발전 및 인터넷의 급속한 발전으로 사회 각 분야에서 인터넷을 통해 전송되는 데이터의 안전성을 위한 보안 기술들이 필요시 되고 있다. 이를 해결하기 위한 기술들의 하나로 XML(eXtensible Markup Language) 보안 기술들이 많이 활용되고 있다. XML에서 제공되는 많은 보안 기능중에 XML 서명 기술은 비XML 문서를 서명하거나 필요한 부분에만 서명이 가능하므로 전송되는 데이터의 안전성을 위해 많이 사용되고 있다. 본 논문에서는 PKI 환경에서 강조되는 암호학적으로 안전한 인증 메커니즘을 위해 X.509 인증서를 XML 서명 기술에 활용하여 사용자간에 인증을 하고 사용자 자원 및 공통 자원에 접근 가능한 접근 제어 모델과 이 모델에 적용 가능한 DTD(Document Type Definition)를 정의하는데 목적이 있다.

I. 서론

인터넷의 급속한 발전으로 사회 각 분야에서 인터넷을 통해 전송되는 데이터의 안전성을 위한 보안 기술 연구가 활발히 이루어지고 있다. 이들 연구 중 특히 전자상거래에서 많이 활용되고 있는 XML 서명 기술은 비XML 문서를 서명하거나 필요한 부분에만 서명이 가능하므로 전송되는 데이터의 안전성을 위해 많이 활용되고 있다.

PKI에서 강조되는 것은 사용자간에 신원을 암호학적으로 안전하게 인증하는데 있다. 따라서 사용자들의 인증 증명서를 관리하거나 사용자간에 인증을 통해 서로에게 접근 권한을 부여하기 위한 기술들이 필요하다. 이에 대한 연구로 X.509 인증서를 XML 서명 기술에 활용하여 통신 주체들간에 인증하고 서로에게 접근 권한을 부여하는 연구가 진행되고 있다[1].

본 논문의 구성은 다음과 같다. 2장에서는 연구 배경으로 XML DTD의 정의와 XML 서명 기술을 살펴보고, 3장에서는 본 논문에서 제안하는 XML 서명을 이용한 접근 제어 모델과 이 모델에서 활용 가능한 DTD 정의 및 DTD를 통해 발행되는 4가지 인증서의 기능을 설명하고 4장에서 본 연구의 결론과 향후 연구 방향을 제시한다.

II. 연구 배경

1. XML DTD

XML 문서에 수반되는 DTD의 목적은 문서 타입 정의로 사용되는 문서의 위치와 XML 문서의 구조를 정의하는 집합이라고 할 수 있다. DTD의 정의는 중요하다. 왜냐하면 DTD는 XML 문서의 형태가 어떻게 이루어지고 어떤 기능들로 이루어졌는지를 한눈에 파악할 수 있기 때문이다.

DTD의 구성은 XML 문서에서 사용되는 모든 엘리먼트들을 정의하는 엘리먼트 타입 선언(Element Type Declarations), 엘리먼트에 대한 범위와 타입을 제한할 수 있게 하는 애트리뷰트 리스

트 선언(Attribute List Declaration), 실제적인 객체이고 XML 문서 내부에서 포인터 역할을 하는 엔티티 선언(Entity Declaration), 파서에게 특정한 정보의 포맷을 처리하도록 외부 애플리케이션을 할당하게 하는 표기선언(Notation Declaration)으로 구성되어 있다[11].

2. XML signature

XML 서명은 간접 지정을 통해 임의의 데이터 객체에 적용된다. 데이터 객체에 다이제스트 값이 계산되어 그 결과 값이 다른 정보와 함께 원소 안에 들어가고, 다음에 그 원소가 다이제스트되어 암호학적으로 서명된다[2]. XML 서명은 그림 1과 같은 구조의 서명 원소들로 표현된다(여기서 "?" : 0 or 1번, "+" : 1번 이상, "*" : 0번 이상의 출현을 나타낸다).

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
     (<Reference URI?>
       (<Transforms>)?
       <DigestMethod>
       <DigestValue>
       </Reference>)+
  </SignedInfo>
  <SignatureValue>
 (<KeyInfo>)?
 (<Object ID?>)*
</Signature>
```

그림 1 : XML 서명

서명은 URI들을 통해 데이터 객체와 연관되어지고 XML 문서 내에서 서명은 프래그먼트 식별자를 통해 데이터 객체와 연관지어진다. 즉 서명 안에 데이터를 포함시킬 수 있고, 서명되어진 데이터를 서명할 수 있다. 또한 외부의 네트워크 데이터나 비XML 문서의 데이터에 대해서도 서명이 가능한 것이다[3, 7].

<Signature>는 차후에 ID 고유성 유효 조건을 범하는 충돌이 발생하지 않도록 서명 속성값(ID)을 부여하는 것이다. <SignedInfo>는 실제로 서명되는 정보로 서명에 대한 검증과 서명 안에 각 레퍼런스(Reference)의 다이제스트에 대한 검증들이다. <SignatureValue>는 Base64 부호화된 서명값을 나타낸다. <CanonicalizationMethod>는 서명 작업의 일부로 서명정보를 다이제스트하기 전에 그 원소를 정규화할 때 사용되는 알고리즘이다. <SignatureMethod>는 정규화된 서명정보를 서명값으로 변화할 때 사용되는 알고림즘이다. 이것은 다이제스트 알고리즘과 암호키를 사용하는 알고리즘, 그리고 패딩 등과 같은 기타 알고리즘 등의 조합이다. <Reference>는 URI 속성으로 서명될 데이터 객체를 식별하며, 그 속성은 선택 사항이다. 즉 이 속성은 서명에서 많아야 1개인데 이는 참조와 객체가 모호함이 없이 짝지어질 수 있도록 하기 위한 제한이다. <Transforms>는 데이터가 다이제스트 되기 전에 그에 적용된 처리 단계의 순서화된 목록이며, 이 목록은 선택 사항이다. <DigestMethod>는 변형형식에 적용된 이후 다이제스트 값을 만들어내기 위해 데이터에 적용되는 알고리즘이고, 데이터를 서명자의 키에 연결하는 부분이다. <KeyInfo>는 수신자들이 서명 검증에 사용될 키를 얻을 수 있도록 해 주는 부분이다. <Object>는 어플리케이션 시스템에 의한 다이제스트 계산 관련정보, 서명생성시의 부가정보를 나타내는 부분이다.

Ⅲ. 접근 제어 모델

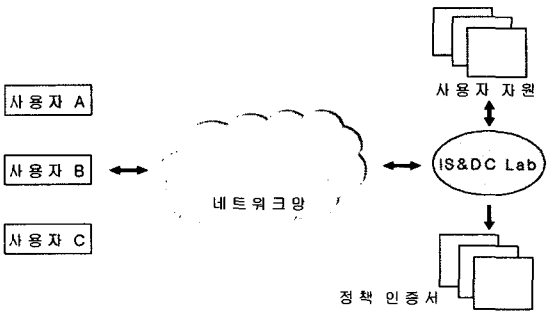

그림 2 : 접근 제어 모델

그림 2는 어느 기관이나 한 부서에서 XML 서명을 이용하여 4가지의 인증서를 사용하여 사용자들의 신원을 증명하고 사용자의 자원이나 사용자의 공통 자원에 접근 가능한 접근 제어 모델이다. 사용자의 인증서와 자원을 관리하는 IS&DC Lab (관리자)은 절대적으로 안전하다는 가정하에 설계되었다. 먼저 사용자 A가 다른 사용자나 공통 자원에 접근하기 위해 IS&DC Lab에게 접근을 요청하면 IS&DC Lab은 사용자 A가 직접 XML 서명을 이용하여 만든 정책 인증서를 통해 신원을 파악하고 올바른 사용자라고 확인되면 사용자 A가 접근하고자 하는 자원에 대한 속성 인증서를 통해 검색을 하고 자기가 관리하는 자원일 경우 접근 권한을 부여하는 방식이다. 물론 사용자 A가 적합하지 않거나 사용자 A가 요구하는 자원을 자기가 관리하지 않는다면 권한 부여가 이루어지지 않을

것이다.

본 논문에서 제안된 접근 제어 모델에서 IS&DC Lab이 관리하는 4가지 인증서 형태는 정책 인증서(Policy certificate), 사용 상태 인증서(Use-condition certificate), 속성 인증서(Attribute certificate), 실행 인증서(Capability certificate)라고 지칭하겠다. 정책 인증서는 사용자가 스스로 XML로 서명한 인증서로 최소의 정보를 가지고 사용자들의 신원을 확인하기 위한 인증서이고 사용 상태 인증서는 사용자들이 접근하고자 하는 자원에 대한 접근 제어를 관리하고 실제적으로 사용 가능한 기능들과 범위를 관리하기 위한 인증서이다. 속성 인증서는 사용자들이 가지고 있는 자원과 공통 자원에 대한 속성값과 이를 이용하기 위해 사용자들이 IS&DC Lab에 요청시 필요한 자원을 검색하는데 이용되는 인증서이고 실행 인증서는 권한을 부여받은 자원에 대한 실행 과정과 실행 상태를 나타내는 인증서들로 구성되어 있다. 제안된 인증 체계는 IETF에서 제안한 속성 인증서[12] 보다 더 간단한 인증서 형태이다.

부록 A의 DTD는 관리자 그룹으로 <IS&DC Lab>이라 하였고, 정의된 DTD의 최상의 루트엘리먼트는 <LabCertificate>라고 정의하였다. 다음의 하위 엘리먼트 <SignablePart>는 사용되는 인증서의 타입을 결정하고, <Header> 엘리먼트는 선택된 인증서의 정보와 인증서에 사용되는 서명 알고리즘과 정규화 알고리즘에 대한 정보를 포함하고 있다. 본 논문에서 사용되는 인증서에는 서명 알고리즘으로 RSA-SHA1, DSA-SHA1을 사용한다. 이후 하위 엘리먼트는 4가지 인증서를 만들기 위한 세부 항목들이다.

1. 정책 인증서

정책 인증서는 스스로 서명한 인증서로 최소의 정보를 가지고 상호간에 신원을 증명하기 위한 인증서이다. 각각의 사용자들은 적어도 1개 이상의 정책 인증서를 가지고 있어야하고 이 인증서에는 발행된 인증기관의 정보로써 사용자 주체의 공개키가 포함된 X.509 인증서와 인증기관에서 인증서를 저장하기 위한 디렉토리와 인증서 폐지를 위한 디렉토리 정보가 포함되어 있다.

사용자가 계층적인 구조로 이루어진 경우 1개 이상의 정책 인증서로 이루어질 수 있으며, 트리 형태로 최상의 정책 인증서를 루트 인증서로 해서 여러 개의 하위 정책 인증서로 구성될 수도 있다. 물론 범위 설정에 따라 상위 자원에 대한 권한이 부여되면 하위 자원에 대해서도 자동으로 권한이 부여될 수도 있다.

부록 A의 DTD를 바탕으로 정책 인증서를 작성하면 <SignablePart>에서 <PolicyCert>가 선택되어지고 <Header>엘리먼트를 통해 인증서의 정보와 사용된 알고리즘을 나타내고, 사용자의 구분을 위해 X.509 인증서와 같이 <Issuer>엘리먼트 안에 <UuerDN>과 <CADN>으로 자기의 소속을 구분한다. <UseCondIssuerGroup>은 적어도 하나 이상의 사용 상태 인증서를 포함하고 있는 인증서 디렉토리이고 <AttrDirs>는 속성 인증서 조회를 위한 URL 목록을 저장하는 디렉토리이다. <CAInfo>는 인증기관의 정보를 나타내는 엘리먼트로 하위 엘리먼트로는 인증기관의 식별 이름을 나타내는 <CADN>과 사용자 주체의 공개키가 포함된 인증서를 나타내는 <X509Certificate>, 인증기관에서 인증서를 저장하기 위한 디렉토리를 나타내는 <IdDirs>, 인증서 폐지를 위한 디렉토리로 <CRLDirs>들로 구성되어 있다.

2. 사용 상태 인증서

사용 상태 인증서는 사용자간에 접근 제어를 위해 적어도 1개 이상 만들어야 한다. 이 인증서는 상대방에게 실제적인 권한을 얻기 위해 사용자의 속성을 구성하는 인증서이다. X.509 인증서에서 사용자의 구분을 위한 구성 요소로 C=국가, O=기관, OU=부서, CN=사용자 이름으로 각각의 속성 DN으로 구분하고 <AttributeInfo>에서 <X509>을 선택하여 사용자의 속성 위치를 구분한다[9].

부록 A의 DTD를 바탕으로 사용 상태 인증서를 작성하면 <SignablePart>에서 <UseCondition Cert>가 선택되어지고 <Header>는 앞에서와 같이 인증서의 정보를 나타내고 <UseConditionCert > 엘리먼트 안에는 사용자가 사용하고자 하는 속성 값이 적합한지를 판별하는 <Condition>과 자원에 대한 읽기, 쓰기, 실행을 나타내는 <Rights>이 있다. <UseConditionCert>의 속성으로 scope는 부여받은 권한의 범위 설정을 하기 위한 부분으로 "local"일 경우는 해당 위치만을 의미하고 "subtree "일 경우는 하위 부분까지의 범위를 의미한다. enable은 먼저 접근한 사용자가 권한을 부여받아 그곳의 자원을 사용중일 때는 다른 사용자가 그 자원에 대해 접근을 허용할 것인지 말 것인지를 나타낸다. <Condition>에는 이 자원에 대한 접근 제한을 나타내는 <Constraint>와 자원에 대한 속성 정보를 나타내는 <AttributeInfo>로 구성되어 있다. 이 속성 정보에는 제한 범위에서 사용되는 속성 <AttrName>과 속성값 <AttrValue>, 외부 권한을 평가하기 위한 속성 <Principal>, 외부 권한을 주기 위한 인수 <ExtArgs>, 속성에 대한 평가를 위한 <STANDARD>, <X509>, <EXT_A

UTH>등으로 구성되어 있다.

3. 속성 인증서, 실행 인증서

속성 인증서는 사용자의 속성값으로 사용자들이 필요한 자원을 IS&DC Lab에게 요청했을 때 적합한 자원이 있는지를 검색하기 위해 사용되는 인증서이다. 기본적인 구조는 앞의 인증들과 비슷하게 정의되고 <AttributeCert> 엘리먼트 안에는 이 속성을 사용하고자 하는 주체를 나타내는 <Subject AndCA>, 속성을 사용하는 방법과 시간적인 제한을 나태는 <Condition>들로 구성되어 있다.

실행 인증서는 권한을 부여받은 사용자가 자원에 대한 실행 과정과 실행 상태에 대한 정보를 담고 있는 인증서이다. 기본적인 구조는 앞의 인증서들과 비슷하게 정의되고 <CapabilityCert> 엘리먼트 안에는 사용되는 자원의 이름을 나타내는 <ResourceName>, 권한을 부여받은 주체를 나타내는 <SubjectAndCA>와 실행과정을 나타내는 <Action>, 실행 상태를 나타내는 <CondtionActions>들로 구성되어 있다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 XML의 DTD정의와 서명 기술에 대해 분석하였으며, 이를 근거로 XML 서명을 이용하여 4 가지의 인증서를 만들어서 사용자들의 신원을 증명하고 사용자들이 요구하는 자원에 대한 검색과 접근 권한을 부여하는 접근 제어 모델을 제안하였다. 또한 사용자들의 위치를 구분하기 위하여 X.509 인증서의 형태로 사용자와 인증기관을 DN으로 구분할 수 있도록 DTD에 X.509 인증서를 활용하였다.

부록에 제안된 XML 서명을 이용한 DTD 정의는 실제로 한 기관이나 부서에서 서로의 신원을 증명하거나 공통 자원과 다른 사람의 자원에 대해서 접근하여서 이를 이용하고자 할 때 유용하게 활용될 것으로 생각된다.

향후 부록에 제안 XML DTD를 이용하여 인증서를 발행하고 실제로 서로간에 접근 제어를 관리할 수 있는 Aphche 서버를 위한 보안 모듈과 Java를 이용한 네트워크 프로그램 개발이 필요할 것으로 기대된다. 또한 제안된 모델에서 하나의 관리자가 모든 인증서를 관리할 경우 계속해서 인증서의 수가 증가되면 시스템 속도면에서 문제가 있을 수 있으므로 이에 대한 연구가 필요하다.

참고문헌

[1] Mary R. Thompson, Srilekha Mudumbai, Abdelilah Essiari, Willie Chin, "Authorization Policy in a PKI Environment", 1st Annual PKI research Workshop, 2002.3.

[2] TTA 표준 초안, "확장성 생성 언어 전자서명 구문과 처리", 2002.11.

[3] IBM developer-Works, http://www-106.ibm.com/developerworks/library/s-xmlsec.html/, IBM 홈페이지, 2003.

[4] Yori Demchenko, "XML Security in IODEF", INCH WG, IETF56, 2003.3.

[5] 김주한, 문기영, "XML 기반 접근제어 기술 동향", 정보보호학회지, 13권 4호, PP.68-73, 2003.8.

[6] 박남제, 문기영, 송승원, 송유진, 원동호, "안전한 전자거래를 위한 XML 키 관리 기술", 정보보호학회지, 13권 3호, PP.72-82, 2003.6.

[7] 송유진, 주재훈 공저, "전자화폐 : 전자상거래 보안응용", 동국대학교출판부 2001.

[8] 이만영, 김지홍, 송유진, 염홍열, 이임영 공저, "전자상거래 보안 기술", 한국정보보호학회 총서 권 3, 생능출판사 1999, 9.

[9] R.Housley, W.Polk, W.Ford, D.Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile<draft-ietf-pkix-new-part1-12.txt", http://www.ietf.org/internet-drafts/draft-ietf-pkix-new-part1-12.txt.

[10] Apache Software Foundation, http://www.apache.org/, Apache 홈페이지.

[11] 김천식, 임도빈 공저, "XML", 도서출판대림 2000.4.

[12] S.Farrell, R.Housley, An Internet Attribute Certificate Profile for Authorization, <draft-ietf-pkix-ac509prof-09.txt>, June, 2001 http://www.itef.org/internet-draft-ietf-pkix-ac509prof-09.txt.

< 부록 >

```
<?xml version="1.0" encoding="euc-kr"?>
<!-- 여기서는 IS&DC-Lab의 다음과 같은 접근 제어 모델 DTD를 정의한다.
Policy Certificates, UseCondition Certificates, Attribute Certificates, Capability Certificates -->
<!-- Note: 1개 이상(+), 0개 이상(*), 0이나 1번만(?) -->

<!ELEMENT LabCertificate (SignablePart)
<!ELEMENT SignablePart (Header, (PolicyCert | UseConditonCert | AttributeCert | CapabilityCert))>

<!ELEMENT Header (Version, ID, Issuer, ValidityPeriod)>
<!ATTLIST Header
        Type (policyCertificate | useCondCertificate | attributeCertificate | capabilityCertificate) #REQUIRED
        SignatureDigestAlg (RSA-SHA1 | DSA-SHA1) #REQUIRED
        CannoAlg (CAN) #REQUIRED>
<!ENTITY RSA-SHA1 "http://www.w3.org/2003.09/xmldsig#rsa-sha1" >
<!ENTITY DSA-SHA1 "http://www.w3.org/2000.09/xmldsig#dsa-sha1" >
<!ENTITY CAN "http://www.w3.org/TR/2001/REC-xml-cl4n-20010315" >

<!ELEMENT PolicyCert (UserName, CAInfo*, UseCondIssuerGroup+, AttrDirs*)>
<!-- UserName : 정책 인증서의 사용자 이름
    CAInfo : 인증기관에서 발행된 X.509인증서와 DN
    UseCondIssuerGroup : 적어도 하나이상의 UseConditionCert 포함하고 있는 인증서 디렉토리
    AttrDirs : 속성 인증서 조회를 위한 URL 목록 -->

<!ELEMENT UseConditionCert (UserName, Condition, Rights, SubjectCA*)>
<!ELEMENT Rights (read*, write*, execute*)>
<!-- Condition : 사용자가 사용하고자 하는 속성 값이 적합한지를 표현
    Rights : 부여된 권한 목록 -->

<!ATTLIST UseConditionCert scope (local | subtree) #REQUIRED enable (true | false) #REQUIRED>
<!-- scope : 부여된 자원에 대한 서브디렉토리의 범위
    enable : 사용자가 자원에 접근중 일 때 다른 사용자의 접근 여부 판별 -->

<!ELEMENT AttributeCert (SubjectAndCA, AttrName, AttrValue, Condition*)>
<!-- SubjectAndCA : 이 속성을 사용하고자 하는 주체
    AttrName : 속성이름
    AttrValue : 속성값
    Condition : 속성을 사용하는 방법과 시간적인 제약 -->

<!ELEMENT CapabilityCert (ResourceName, SubjectAndCA, Action*, ConditionalActions*)>
<!-- ResourceName : 사용되는 자원의 이름
    SubjectAndCA : 권한을 부여받은 주체
    Action : 실행과정
    ConditionActions : 실행 상태  -->

<!ELEMENT ConditonalActions (Condition, Actions)>
<!ATTLIST ConditonalActions critical (true | false) #REQUIRED>
<!-- Condition : 속성을 사용하는 방법과 시간적인 제약
    Actions : 접근 부여받은 자원의 실행 과정
    Critical : 실행 과정일 때 다른 사용자의 접근을 여부를 판별 -->
```

부록 A : XML 서명을 이용한 접근 제어 모델 DTD

```
<!ELEMENT CAInfo (CADN, X509Certificate+, IdDirs*, CRLDirs*)>
<!-- CADN : CA의 식별 이름
     X509Certificate : 공개키를 포함한 인증서
     IdDirs : CA에서 인증서를 저장하는 기본 디렉토리
     CRLDirs : 인증서 폐지를 위한 0개 이상의 디렉토리 -->

<!ELEMENT Condition (Constraint, AttributeInfo+)>
<!-- Condition : 사용자와 CA가 속성과 속성값이 무엇이며 이를 만족하는지를 나타내는 논리값
     Constraint : 자원에 대한 제한 나타냄 -->

<!ELEMENT CRLDirs (URL+)> <!-- 인증서 폐지를 위한 0개 이상의 디렉토리 -->
<!ELEMENT AttrDirs (URL+)> <!-- 속성 인증서를 제공하기 위한 0개 이상의 디렉토리 -->
<!ELEMENT IdDirs (URL+)> <!-- 사용자 식별 인증서를 제공하기 위한 0개 이상의 디렉토리 -->
<!ELEMENT UseCondIssuerGroup (Principal+, URL+)> <!-- 관리 그룹을 위한 인증서 디렉토리 -->

<!ELEMENT AttributeInfo (AttrName, AttrValue, (CADN | Principal), AttrDirs*, ExtArgs*)>
<!ATTLIST AttributeInfo type (STANDARD | X509 | EXT_AUTH) #REQUIRED>
<!-- AttrName : 제한 범위에서 사용되는 속성 이름
     AttrValue : 제한 범위에서 사용되는 속성값
     CADN : X509 속성을 포함하고 있는 CA의 식별 이름
     Principal : 외부 권한을 평가하기 위한 속성 이름
     AttriDirs : 속성 인증서를 제공하기 위한 디렉토리
     ExtArgs : 외부 권한을 주기 위한 인수

     STANDARD : 어떤 시스템에 의해 평가되어진 속성
     X509 : X509 인증서의 속성(O, OU, CN)
     EXT_AUTH : 외부 권한에 의해 평가되어진 속성 -->

<!ELEMENT ValidityPeriod EMPTY> <!-- 인증서의 유효기간 -->
<!ATTLIST ValidityPeriod
    start CDATA #REQUIRED
    end CDATA #REQUIRED>
<!ELEMENT ExtArgs (String+)>
<!ELEMENT ID EMPTY> <!-- 모든 인증서에 부여되는 독자적인 ID -->
<!ATTLIST ID id CDATA #REQUIRED>
<!ELEMENT Version EMPTY> <!-- 인증서 포맷 버전 -->
<!ATTLIST Version ver CDATA #REQUIRED>
<!ELEMENT Issuer (UserDN, CADN, URL*)>
<!ELEMENT Principal (UserDN, CADN)>
<!ELEMENT SubjectAndCA (UserDN, CADN)>
<!ELEMENT URL (#PCDATA)> <!-- protocol, host, port and file name -->
<!ELEMENT CADN (#PCDATA)>
<!ELEMENT SubjectCA (#PCDATA)>
<!ELEMENT X509Certificate (#PCDATA)>
<!ELEMENT UserDN (#PCDATA)>
<!ELEMENT ResourceName (#PCDATA)>
<!ELEMENT read (#PCDATA)>
<!ELEMENT write (#PCDATA)>
<!ELEMENT execute (#PCDATA)>
```

부록 A : XML 서명을 이용한 접근 제어 모델 DTD(계속)