Journal of Korea Multimedia Society Vol. 8, No. 7, July 2005(pp. 869-881)

확장된 RBAC를 이용한 XML 문서에 대한 접근제어

반용호†, 김종훈††

요 약

XML(eXtensible Markup Language)은 인터넷상에서 문서를 표현하고 교환하기 위한 표준으로 자리 잡았다. XML 문서는 내부적으로 민감성의 등급을 달리하는 정보를 포함하는 구조로 이루어져 있으므로, XML 문서를 사용하는 사용자 그룹의 제한적인 접근 및 공유를 위한 방법이 요구된다. 이를 해결하기 위해서는 XML 문서에 대한 접근제어 정책을 규정하고 수행하기 위한 모델과 메커니즘이 필요하다. XML 문서에 사용되는 접근제어 메커니즘은 사용자가 소유하고 있는 권한부여 정보에 의존하여 문서를 안전하고 선택적으로 배포할 수 있는 기능을 지원해야 한다. 본 논문에서는 XML 문서가 인터넷 환경에서 안전하게 보호될 수 있도록 하는 접근제어 모델과 메커니즘을 제안한다. 제안된 모델에서는 XML 문서에 대해 권한에 부합하는 접근제어 수행을 위하여 RBAC를 기반으로 하는 접근제어 정책을 사용한다. 본 논문에서 제안된 모델 및 메커니즘은 XML 문서뿐 아니라 XML 문서 내부의 컴포넌트인 엘리먼트, 속성, 링크 등의 세부요소들에 대한 권한정의를 통하여 XML 문서의 안전한 사용을 보장한다.

Access Control for XML Documents Using Extended RBAC

YongHo Ban†, JongHun Kim††

ABSTRACT

XML(eXtensible Markup Language) has emerged as a prevalent standard for document representation and exchange on the Internet. XML documents contain information of different sensitivity degrees, so that XML Document must selectively shared by user communities. There is thus the need for models and mechanisms enabling the specification and enforcement of access control policies for XML documents. Mechanisms are also required enabling a secure and selective dissemination of documents to users, according to the authorizations which the users have. In this paper, we give an account of access control model and mechanisms, which XML documents can be securely protected in web environments. We make RBAC Based access Control polices to the problem of secure and selective access of XML documents. The proposed model and mechanism guarantee that the secure use for XML documents through definition of authority for element, attribute, link within XML document as well as XML document.

Key words: XML, Schema(스키마), Access Control(접근제어), Security(보안), RBAC

1. 서 론

World Wide Web 컨소시엄에 의해 제안된 XML (eXtensible Markup Language)은 인터넷상에서 문서를 표현하고 교환하기 위한 표준으로 자리 잡았다 [18]. XML은 HTML과 같은 기존에 주로 사용되는 문서와는 달리 정보 저장 측면에서 구조적인 구성을 보장하기 때문에 정보 교환 뿐만 아니라 정보 검색에

※ 교신저자(Corresponding Author) : 반용호, 주소 : 부산시 사하구 하단2동 840번지(604-714), 전화 : 051)200-5590, FAX : 051)200-7783, E-mail : gaussian@donga.ac.kr
접수일 : 2004년 7월 19일, 완료일 : 2005년 1월 19일
† 준회원, 동아대학교 컴퓨터공학과 박사수료
†† 종신회원, 동아대학교 컴퓨터공학과 교수
(E-mail : jhkim@dau.ac.kr)
※ 본 논문은 2001년도 정보통신(IT)사업 연구과제의 연구비 지원에 의하여 연구되었음.

도 적극적으로 활용되고 있다. 또한 XML은 재사용성 및 타 기종 시스템으로의 이식성이 뛰어나 기존의 문서 표현방식에 비해 효율적이며 특정 응용이나 시스템에 구애받지 않으므로 호환성면에서도 기존 방식에 비해 장점을 가지고 있다. XML이 가진 이러한 특징들은 현재 관심을 모으고 있는 웹 서비스, 전자상거래, 기업응용시스템통합(EAI) 그리고 유비쿼터스 컴퓨팅 분야에서 기반 기술의 하나로 XML을 채택하게 하는 근간이 될 것이다. 그러나 XML 문서는 단순한 텍스트 데이터이며, 인터넷을 통해 공유되는 특징을 가지고 있다. 이것은 XML로 구성된 정보는 모든 사용자에게 개방되어 있음을 의미하며, 특히 악의적인 사용자에게 XML로 구성된 정보가 노출될 수 있는 위험성 역시 내포하고 있음을 의미한다. 이를 해결하기 위해서 XML에 관련된 다양한 보안 기법들에 대한 연구가 진행되고 있는데, 이중 대표적인 것은 W3C에서 표준화가 이루어진 XML 전자서명(Signature), XML 암호화(Encryption) 등을 예로 들 수 있다. 또한 SAML(Security Assertion Markup Language), XACML(XML Access Control Markup Language), XrML(eXtensible Rights Markup Language) 등과 같은 다양한 XML과 관련된 보안 기술들이 OASIS, IBM, Apache등과 같은 여러 기관 및 단체에 의해 활발하게 연구가 진행되고 있다[19-21]. 그러나 기존에 진행된 연구에서는 XML 문서에 대한 무결성 및 기밀성 유지를 위한 방법에만 그 연구의 초점이 집중되었을 뿐, XML 문서에 포함되어 있는 각각의 요소에 대한 민감성의 수준을 정의하고, 정의된 민감성의 수준에 따라 문서의 각 요소에 대해 차별화된 보안 정책을 적용하는 방식에 대해서는 아직까지 기본적인 수준에서 연구가 진행되고 있다[2,4,9,11,12].

본 논문에서 XML 문서에 접근하고자 하는 사용자의 접근권한을 미리 설정된 보안정책에 따라 판단하여 해당 사용자가 가진 권한에 따라 해당 문서에 대한 접근 범위를 결정하여 전체 문서에 대한 접근 허용 또는 거부뿐만 아니라, 문서를 구성하는 각각의 요소에 대한 수준별 접근제어를 수행하는 방법에 대한 연구 결과를 제시한다. 일반적으로 접근제어 시스템은 보안 정책, 보안 모델, 보안 메커니즘으로 구성되는데, 본 논문에서는 접근제어 시스템 구현을 위하여 Ravi. S 등이 제안한 역할기반 접근제어 정책(Role-based Access Control Policy)을 XML 문서에 적합한 형태로 재정의 하고 이를 바탕으로 보안 모델 및 메커니즘을 정의하였다. 역할기반 접근제어 정책은 임의 접근제어 정책 및 강제 접근제어 정책과 달리 각 사용자에게 역할을 부여하여 역할에 따라 권한을 할당하는 방식으로 임의 접근제어와 강제 접근제어 방식에 비해 권한부여 관리 및 성능 면에서 효율성을 가진다[1,8]. 또한 본 논문에서는 객체를 XML schema와 이를 구현한 인스턴스 문서로 정의하여 문서 처리 면에서도 별도의 프로그램을 요구하지 않도록 함으로써 기존의 방식에 비해 효율성을 가지도록 하였다. 본 논문에서 제안된 접근제어 메커니즘이 XML 문서의 기밀성, 무결성을 보장하기 위해 진행된 기존의 연구에 접목된다면 보다 효율적으로 XML 문서를 보호할 수 있도록 하였다.

본 논문의 전체적인 구성은 다음과 같다. 서론에서 연구의 목적 필요성을 간략히 서술하고, 2절에서 기존에 연구된 접근제어 시스템과 XML과 관련된 배경 지식 및 보안 요건에 대하여 언급한다. 3절에서는 본 논문에서 사용되는 주체, 객체, 권한부여, 연산, 제약, 충돌 등을 정의하고 각 구성요소의 특징들을 언급한다. 4절에서는 XML 문서를 위한 접근제어 모델을 제안하고 제안된 모델이 가지고 있는 주요 특징들을 기술한다. 5절에서는 제안된 접근제어 모델을 기반으로 XML 문서를 권한에 따라 관리하는 접근제어 메커니즘에 대하여 언급하고, 제안된 메커니즘이 실제 환경에서 어떻게 적용될 수 있을지를 기술한다. 6절에서는 제안된 모델 및 메커니즘을 기반으로 참조구현 된 결과를 제시하고, 이전에 제안된 방식과 본 논문에서 제안된 방식을 비교함으로서 제안된 방식이 가지는 특징과 차이점을 기술하고, 마지막으로 결론 및 향후에 추가적으로 진행되어야 할 연구방향을 제시한다.

2. 기본 개념

2.1 역할기반 접근제어

접근제어란 식별 및 인증을 통해 시스템 접근이 허용된 사용자가 허가된 범위 내의 정보에만 접근할 수 있도록 하는 기술적인 방법으로 정의된다. 시스템 관리자는 접근제어를 통해 기밀 정보가 권한 없는 사용자에게 유출되는 것을 막는 기밀성, 권한 없는 사용자가 불법적으로 중요 정보의 변조를 방지하는

무결성, 그리고 권한을 부여받은 사용자가 정보를 사용할 수 있도록 보장하는 가용성을 제공할 수 있다. Ravi S. 등에 의해 제안된 역할기반 접근제어의 중심적인 개념은 사용자가 기업이나 조직의 정보 자원을 임의로 접근할 수 없도록 하는 것이다[1,8]. 즉, 접근권한이 역할에 부여되고 사용자는 적절한 역할에 소속됨으로써 역할의 수행에 필요한 최소 자원만을 접근할 수 있도록 한다. 이러한 방식은 권한 관리를 매우 단순화 시켜주고 기업의 특정한 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. RBAC 모델은 사용자, 역할, 허가의 3가지 요소를 기본으로 사용자와 역할, 역할과 허가 상의 관계를 사용자 할당과 권한 할당으로 표현한다. 특히 한 사용자가 특정 멤버의 구성원으로 활동하는 동안 할당된 역할들의 사상을 세션(Session)이라고 한다. 기본적인 RBAC 모델에 역할계층구조, 제약조건이라는 특성이 추가될 수 있는데 역할계층(Role Hierarchy)은 권한과 책임에 대한 조직 내의 순서를 나타낼 수 있는 가장 일반적인 방법으로 트리구조로 나타내며 상위역할의 권한은 하위 역할에 상속될 수 있다. 제약조건(Constraints)은 접근제어 정책이 실제 시스템 환경에 적용 가능하도록 하는 사전 규약들로 사용자 할당, 권한 할당, 그리고 접근제어 세션에 적용된다. 그림 1은 앞에서 언급된 요소들로 구성된 RBAC96 모델을 보여준다.

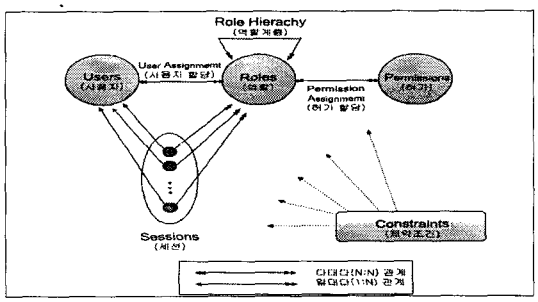

그림 1. RBAC96 모델

2.2 XPath

XPath는 XML 문서의 각 부분에 접근하기 위해 W3C에 의해 제안된 언어로, 데이터 집합에서 사용자가 요구하는 정보만을 추출해 낼 수 있는 유연한 방법을 제공한다. 즉, XPath는 특정 XML 문서로부터 어떤 데이터를 추출해낼지 기술하는 방법을 제공하는 언어이다. 여러 개의 XML 문서로부터 사용자 요구사항에 맞는 새로운 문서를 생성하는 경우 원본 문서의 어느 부분이라도 접근 할 수 있는 XPath를 이용한다면, 문서의 한 부분을 삭제하고 변환하는 수준을 넘어 완전히 새로운 문서를 생성할 수 있는 장점을 이용할 수 있다. XPath가 제공하는 이러한 특징은 특정 XML 문서에 대한 권한별 접근을 수행하고, 그 결과에 따라 권한에 부합하는 정보만을 추출하는데 사용할 수 있다. XPath 수식은 하나의 로케이션 패스로 구성되는데, 이는 다시 연속된 하나 이상의 로케이션 스텝으로 이루어지며, 각각의 로케이션 스텝들은 '/' 기호에 의해 구분된다[17].

2.3 schema

XML은 SGML로부터 물려받은 DTD(XML Document Type Definition)를 기본적인 스키마 정의 메커니즘으로 채택하고 있지만 XML 문서의 자동화된 처리를 위해서는 DTD 보다 엄격하고 표현력이 풍부한 스키마 정의 메커니즘이 필요하다. W3C에 의해 제안된 XML schema 표준은 XML 문서의 스키마를 기술하는 정보 모델과 이 정보 모델을 XML로 표현하는 방법을 정의하고 있다. XML 스키마를 XML 언어로 기술함으로써 얻는 이점은 기존의 XML 소프트웨어와 API를 그대로 이용하여 XML 스키마의 메타데이터에 접근할 수 있다는 점이다. XML 스키마는 엘리먼트, 속성, 데이터타입, 그룹과 같은 XML 컴포넌트의 복잡한 제약을 지원한다. 또한, XML 스키마는 스키마와 엘리먼트 사이에서 풍부한 재사용 관계를 구성하기 위한 메커니즘을 제공한다. 이러한 특징 때문에 본 논문에서는 접근제어 모델에서 정의되는 객체의 일부분으로써 schema를 사용하였다[22].

2.4 XML 문서의 보안 요구사항

XML 문서에 대한 보안 요구사항은 다음과 같이 요약된다. 첫째 XML 문서가 가지는 구조적 특징으로 인한 것으로, XML 문서는 보안 민감성의 수준이 다른 각각의 요소를 포함할 수 있으므로 이를 만족하는 다양한 보안 계층이 지원되어야 한다. 이러한 요구사항을 만족시키기 위해서는 XML 문서를 위한 접근제어 메커니즘은 최소한의 단위로 보안 정책을 적용할 수 있는 충분한 유연성을 가져야 한다. 즉, XML

문서에 대한 접근제어 정책은 완전한 XML 문서뿐만 아니라, 해당 문서가 포함하는 최상위 노드, 엘리먼트, 하위 엘리먼트, 속성 및 링크로 연결된 외부 문서에 대하여 접근제어 정책이 전파되어 해당 정책이 즉시 적용됨으로서 권한에 부합하는 접근이 수행될 수 있어야 한다[2,4,12,14].

두 번째 요구사항은 XML 문서는 항상 미리 정의된 문서 형태로 구성될 수 없다는 점에서 기인한다. 일반적인 접근제어 정책은 문서 유형의 관점에서 정의되는 경우가 대부분이므로, 특정 위치에 있는 문서에 대하여 접근제어 정책을 적절하게 운용한다 하더라도 처리되지 않을 상황이 발생 할 수 있는데, 접근제어 메커니즘은 그런 상황에 적절히 대처해야 한다. 즉, 사용자의 요청에 따라 반환되어야 하는 문서 포맷의 사전 정의 없이 적절한 접근제어 정책의 적용이 가능한 메커니즘 및 이를 지원하는 동적인 접근제어 메커니즘이 요구된다[6,10].

세 번째 요구사항은 다중 사용자 환경에서의 접근제어 문제에 관한 것으로 동일한 스키마를 기반으로 생성된 인스턴스 문서에 대한 사용자별 접근제어가 반드시 필요하다. 즉, 동일한 스키마를 따라 생성된 인스턴스 문서의 정보는 동일한 역할을 부여 받은 사용자 별로 또 다른 접근제어를 수행 할 수 있는 메커니즘이 요구된다[13].(예를 들어, 동일한 부서에 소속된 의사는 동일한 역할을 부여 받게 되지만, 담당하는 환자의 정보는 모두 틀리므로, 각각의 의사별로 해당 환자의 기록을 볼 수 있도록 해야 한다.)

3. 제안된 모델의 구성요소

접근제어 모델 및 메커니즘의 개발은 주체와 객체에 대한 정의를 필요로 한다. 이번 절에서는 본 논문에서 제안된 모델에서 사용되는 주체와 객체 그리고 이들 사이에서 수행되는 연산에 대하여 기술한다.

3.1 주체(Subject)

접근제어 정책은 주체가 유효한 어떤 정보를 소유하고 있다는 개념을 기반으로 정의된다. 주체가 소유한 이 정보는 보안 목표를 위해 요구되는 주체의 속성들로 구성된다. 각각의 주체는 하나 혹은 그 이상의 유효한 정보와 연결된다. 이 유효 정보는 주체가 시스템에 등록되거나 동의하였을 때 할당된다. 본 논문에서는 주체를 다음과 같이 정의한다. S = (user_id, password, [purpose]) or (certificate, certificate_password, [purpose]). 주체는 시스템에 의해 미리 배포된 사용자 ID와 패스워드, 그리고 접근 목적으로 이루어진다. 여기서 '[]'는 생략될 수 있음을 나타낸다. 주체의 또 다른 표기는 인증기관(CA)으로부터 발급된 인증서와 전자서명 비밀번호, 접근 목적으로 이루어진 형태가 될 수도 있다. 인증서를 가지고 특정 시스템에 로그인하는 유형은 현재 그 사용 추세가 증가하고 있으므로 충분한 논의대상이 될 수 있지만, 본 논문의 주제를 벗어나므로 본 논문에서는 주체를 사용자 ID, 패스워드 그리고 접근 목적으로 이루어진 집합을 가진 사용자로 제한한다. E. Damiani 등은 주체를 <user-id, IP-address, sym-address>로 정의하고 있다[3]. 그러나 IP 주소의 경우 유일하게 식별된다는 특징을 지니고 있음에도 불구하고, 특정 서비스 환경에서는 정적할당이 아닌 동적할당이 이루어지는 경우가 있으므로 IP주소를 사용하여 주체를 구분하는 방식은 그 한계점이 존재한다.

3.2 객체(Object)

접근제어 정책에서 접근제어의 대상이 되는 자원들의 집합을 객체라고 정의한다. 본 논문에서는 XML 스키마와 스키마를 만족하는 인스턴스(Instance) 문서를 객체로 정의한다. XML 스키마와 기능적 특징이 유사한 DTD를 접근제어의 기본단위로 정의할 수도 있다. 그러나 EBNF 형식의 문법을 따르는 DTD 방식과 달리 XML 스키마는 W3C에서 제안된 표준 XML 문법을 사용하여 필요한 문서구조를 정의할 수 있는 특징을 가지고 있다. 이것은 XML 파서 만을 사용하여 XML 스키마를 처리할 수 있다는 것을 의미하므로 DTD를 접근제어 객체로 정의하는 것보다 효율적이다. 또한 XML 스키마 문서 역시 XML 문서이므로 XML 암호화, XML 전자서명과 같은 다른 XML 보안 표준과의 통합에도 유리하다. 이와 같은 문서 처리의 효율성과 통합 가능성을 고려하여 본 논문에서는 스키마를 객체의 일부분으로 정의하였다. 객체의 또 다른 부분은 스키마를 기반으로 구성되는 XML 인스턴스 문서로 정의된다. XML 문서가 아닌 일반적인 파일은 그 구조가 복잡하기 때문에, 파일의 일부분에 대한 접근을 제약하거나, 파일에 대한 부분적인 접근 권한을 부여하는 것은 매우

어렵다. 따라서 일반적인 파일에 대한 접근제어는 파일 전체에 대하여 일괄적으로 수행된다. 그러나 XML은 트리 모양을 한 계층 구조를 가지고 있으며, XML 문서는 여러 개의 엘리먼트로 구성된다. 각각의 엘리먼트는 트리의 노드에 정확히 일치한다. 각 노드는 XML 데이터의 한 부분을 표현하므로, 데이터의 각 부분에 대한 접근제어는 각각의 엘리먼트에 대한 접근 권한을 할당함으로써 가능하다. 이것은 세분화된 객체의 정의를 통하여 보안 수준별 접근제어를 수행 할 수 있다는 것을 의미한다.

3.3 연산(Operation)

권한부여(Authorization)는 주체에게 무엇을 할 수 있거나 소유할 수 있는 역할을 부여하는 과정으로, 특정 사용자에게 적절한 권한부여를 위하여 시스템 관리자는 정의된 역할에 따라 접근제어 객체에 대한 권한을 미리 설정해야 한다. 권한부여 과정에서 객체 표현은 XPath를 통해 이루어지며, 특정 사용자의 역할은 해당 사용자에 의해 제출된 사용자 정보를 통해 판단된다. 이런 과정 후에 사용자는 해당 객체에 대한 다양한 권한을 수행할 수 있는데, 이러한 작업 유형을 연산(Operation)이라고 한다. 접근제어 모델에서의 연산 유형은 사용되는 모델의 형태에 따라 다양하게 정의할 수 있다. 논 논문에서 정의된 연산 유형을 표 1에서 제시한다.

스키마와 XML 문서에 대한 연산은 해당 연산의 종류에 따라 우선순위를 부여할 수 있다. XML 스키마와 XML 문서에 대한 권한 부여는 스키마 생성(SG), 스키마 읽기(SR), 인스턴스 생성(IG), 인스턴스 읽기(IR), 인스턴스 수정(IW)의 형태를 가진다. 스키마 문서의 권한 유형과 인스턴스 문서의 권한 유형에서는 스키마 문서의 권한이 인스턴스 문서의 권한을 지배하고 있으며, 이들의 관계를 계층 구조로도 표현 될 수 있다. 이 경우 상위 권한 유형은 하위 권한 유형에 대하여 묵시적인 지배 권한을 가지게 된다. XML 문서에 대한 연산은 크게 두 가지로 형태로 다시 나누어지는데, XML 문서(인스턴스 계층)에 대한 연산 유형과 XML 문서 내부의 노드(엘리먼트 계층)에 대한 연산으로 정의할 수 있다. XML 문서 내부의 각 노드(엘리먼트, 속성)에 대한 권한 부여 유형은 두 가지 유형으로 정의된다. 이것은 XML 문서에 정의된 각 노드의 엘리먼트 혹은 속성을 읽을 수 있는 연산인 엘리먼트 읽기(ER), XML 문서의 엘리먼트나 속성을 수정할 수 있는 엘리먼트 수정(EW)이다. 문서 내의 각 노드에 위치한 엘리먼트에 대한 권한의 유형은 해당 엘리먼트를 포함하고 있는 문서의 권한과 상호 결합된 형태로 존재한다. 여기서, 문서 전체에 정의된 권한과 노드에 위치한 엘리먼트에 개별적으로 정의된 권한 사이에서 권한 부여의 중복이 발생할 수 있는데, 여기서도 상위의 권한은 하위의 권한에 대하여 묵시적인 지배 권한을 가지게 된다.

표 1. 연산 유형과 연산의 의미

연산 유형		연산의 의미	우선순위
schema level	schema generate (SG)	schema 생성	0
	schema read (SR)	schema 읽기	0
	schema delete (SD)	schema 삭제	0
instance level	instance generate (IG)	instance 생성	1
	instance read (IR)	instance 읽기	2
	instance write (IW)	instance 수정	3
element level	element read (ER)	element 읽기	2
	element write (EW)	element 수정	3

4. RBAC를 확장한 XML 문서 접근제어 모델 (RBACFX)

본 절에서는 앞서 논의된 역할기반 접근제어 정책을 기반으로 XML 문서에 대한 접근제어를 수행하는 접근제어 모델을 제안한다.

4.1 XML 문서를 위한 접근제어 모델(RBACFX)

본 논문에서 제안한 모델은 Ravi S. 등이 제안한 RBAC96 모델을 확장한 형태로 기존의 역할기반 접근제어의 정의를 따른다[1]. 제안된 모델의 기본 컴포넌트는 RBAC96 모델과 마찬가지로 사용자, 역할, 허가로 구분된다. 해당 컴포넌트들의 관계 역시 RBAC96모델에서와 같이 각 사용자에게 특정 역할을 부여하는 사용자할당, 역할에 임의의 권한을 할당하는 허가할당으로 구분된다. 역할 계층 및 제약조건 또한 RBAC96 모델에서처럼 적용될 수 있다. 그러나 본 논문에서 제안된 모델과 기존에 제안된 모델과의 주요한 차이점은 기본적으로 접근제어의 대상에 있다. RBAC96 모델은 문서의 일부가 아닌 전체 문서

에 대한 접근을 허가하거나 반대로 전체 문서에 대한 접근을 거부하도록 하는 것에 기본적인 초점을 둔 반면에, 본 논문에서 제안하는 모델은 주어진 XML 문서 구조를 해석하여 각 엘리먼트 별로 접근제어를 수행 할 수 있도록 함으로써 저장소에 저장된 전체 문서뿐만 아니라 문서의 일부분에 대한 접근제어를 수행할 수 있도록 하였다. 즉, 본 논문에서 제안된 모델은 기존에 제시된 모델에서 고찰되지 않았던 광역 접근제어 개념 및 지역 접근제어 개념이 혼합되어 보다 세밀한 접근제어를 수행할 수 있도록 설계된 모델이다. 또한, 본 논문에서 제안된 모델은 XML 스키마에 대한 접근제어 연산을 추가하였다. 접근제어가 적용되는 객체가 XML 인스턴스와 XML 스키마로 구분되면서 제안된 모델의 퍼미션(Permission) 컴포넌트 역시 확장되었다. 확장된 퍼미션 컴포넌트는 스키마 퍼미션과 인스턴스 퍼미션으로 구분되며, 이들 퍼미션의 정의와 각 퍼미션 간의 관계는 다음절에서 기술한다. 본 논문에서 제안된 XML 문서를 위한 접근제어 모델의 구조는 그림 2와 같이 나타낼 수 있다.

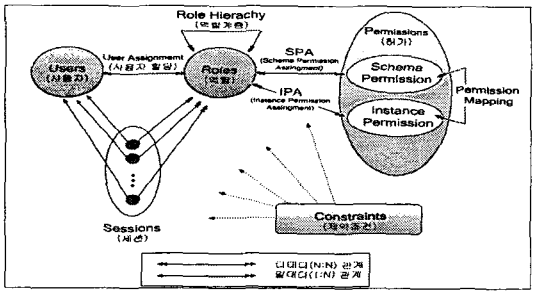

그림 2. XML을 위한 접근제어 모델

4.2 접근제어 객체

3장에서 논의한 바와 같이 본 논문에서의 객체는 크게 두 가지로 나누어진다. SO(schema Object)는 XML 스키마 또는 스키마 컴포넌트를 의미한다. 스키마 객체는 XPath 표현에 의해 생성된다. IO(Instance Object)는 XML 인스턴스 또는 인스턴스 컴포넌트를 의미한다. SO와 IO를 서로 연결하는 과정을 인스턴스 매핑(Instance Mapping)이라 한다.

4.3 퍼미션(Permission), 할당, 보안계층

그림 3의 모델은 2가지의 퍼미션을 가진다. 이것은 XML 스키마에 대한 퍼미션과 인스턴스에 대한 것으로 스키마에 대한 퍼미션은 XML 스키마 객체에 대하여 수행 가능한 연산집합을 의미하고, 인스턴스 퍼미션은 XML Instance 객체에 대하여 수행 가능한 연산집합을 의미한다. 스키마 퍼미션을 역할에 할당하는 과정을 SPA(Schema Permission Assignment)라 한다. 하나의 스키마 퍼미션에 여러 개의 역할을 부여할 수 있으며, 한 개의 역할 역시 여러 개의 스키마 퍼미션을 가질 수 있으므로 SPA는 다대다(n-to-n)의 관계를 이룬다. 인스턴스 퍼미션을 역할에 할당하는 과정을 IPA(Instance Permission Assignment)라 한다. SPA와 마찬가지로 역할 컴포넌트와 다대다 관계를 가진다. 하나의 XML 스키마는 자신의 스키마 구조를 따르는 여러 개의 인스턴스 문서를 생성할 수 있다. 특정 스키마와 이들 스키마를 따르는 퍼미션 역시 일정한 관계를 가지는데 이들 사이의 연관성을 퍼미션 매핑(Permission Mapping)이라고 한다. SOH(Security Object Hierarchy)는 보안 관리자에 의해 정의된 스키마 객체간의 부분적인 순서 관계를 말한다. SOH는 스키마 객체사이의 재사용 관계를 기반으로 한다. 재사용 관계는 비순환적이며 재귀적인 관계를 가지는 부분적인 순서로서 간주될 수 있으며, 낮은 계층의 객체 상에 정의된 권한 부여가 상위 계층 객체로 전달되게 된다. 일부 제약들이 SPA와 IPA에 대하여 정의된다. SPA는 역할과 SP사이에서 정의되는 명시적인 역할의 권한배정을 의미 하며, IPA는 묵시적인 역할의 권한배정을 의미한다.

4.4 접근제어 수행 알고리즘

제안된 모델에서 접근제어를 수행을 위한 연산 알고리즘의 일부를 그림 4에서 보인다. 이 알고리즘은 사용자를 역할에 배정하는 부분과 해당 문서에 대하여 접근제어를 수행하여 사용자에게 반환되는 문서를 생성하는 부분으로 구성된다. 알고리즘의 입력은 사용자의 ID, 사용자에게 할당된 역할, 접근 대상 객체, 요청 연산으로 구성되며, 알고리즘 수행 결과는 최종적으로 사용자에게 반환되는 문서(수행 결과에 따라서 정보가 없는 빈 문서)가 된다. 먼저 특정 문서에 대한 접근을 요청한 사용자의 역할이 역할 계층에 정의되어 있는지를 검사한다. 만약 역할이 역할 배정 정보에 정의되지 않은 사용자이면 접근제어 수행 알

고리즘은 연산을 중단하고 접근이 거부되었음을 알리는 정보를 반환한다. 적절한 역할이 배정된 사용자인 경우, 해당 역할이 스키마 레벨에서의 접근 가능 여부를 판단한다. 스키마 레벨에서의 접근 배정되지 않은 역할은 스키마를 따르는 인스턴스 문서의 접근이 불가능함을 의미하므로 접근제어 수행 알고리즘은 연산을 중단하고 접근이 거부되었음을 알리는 정보를 반환한다. 스키마에 대한 접근이 허용된 역할은 인스턴스에 대한 접근이 허용된 사용자이므로, 부여된 역할에 사용자의 정보를 결합하여 인스턴스에 대한 접근 연산을 수행하고 접근 연산 결과를 반환한다.

```
/* 사용자 -역할 배정 검사 */
  IF r ∉ roles( u, UR.xml ) THEN
    ACCESS 거부
  END IF
/* 역할 - 스키마 문서 할당 검사 */
  permission_check01 = EPA_check( UR.xml, schema_
permission_info.xml )
  IF
    permission_check01 = not permission
  THEN
    ACCESS 거부
  ELSE
/* 출력 가능한 문서를 생성하는 부분 */
  t = parse(traget.xml)
  // 각 하위 트리 t에 대하여
   recursive_access(t, rootnood)
   so = sm(rootnood)
   IF so ∉ s_permission(r, PR.xml, RH.xmll) THEN
          rootnode에 대한 접근 허용
          add(traget.xml, root_node);
          IF (leaf ≠ root_node ) THEN
                  FOR(s_tree st ∈ t )
                        recursive_acc(st, sub_node);
                  END FOR
          END IF
   ELSE
        ACCESS 거부
  // 출력 문서의 스키마와 일치하는지 검증 <optional>
   IF target'.xml ∈ im(traget_schema.xml) THEN
          return target'xml
   ELSE
          document(target'.xml)에 대한 ACCESS 거부
  END IF
```

그림 3. 접근제어 수행을 위한 연산 알고리즘

5. RBACFX 모델에 대한 응용 메커니즘

본 절에서는 본 논문에서 제안된 접근제어 모델인 RBAC*FX*을 기반으로 XML 문서에 접근제어를 적용하기 위한 접근제어 메커니즘을 기술한다. 제안된 접근제어 메커니즘은 사용자 서비스 요청 과정, 사용자 할당 과정, 허가할당 과정, 권한부여 과정의 총 4 단계로 구성된다.

5.1 사용자(User) 서비스 요청 과정

사용자는 사전 등록 절차를 통해 ID와 Password를 부여 받았으며, 서비스 요청에 앞서 부여받은 ID와 Password로 인증(Authentication)된 상태라고 가정한다. 인증된 사용자는 XML 객체에 대한 서비스를 요청할 수 있는데, 먼저 그림 4와 같은 형태의 정보를 권한부여 서버에 전송한다.

그림 4에서 ID는 사용자가 사전에 부여받은 식별값, Req-Target은 사용자가 접근하고자 하는 대상 객체, Req-Operation은 사용자가 요청한 대상 객체에 연산 유형을 의미한다. ID는 해당 사용자에게 부여된 유일한 값으로 중복되지 않아야 한다. ID가 중복되는 경우 역할 및 권한 할당에 중복 및 오류가 생길 수 있기 때문에 각 사용자 별로 하나의 유일한 ID가 부여되어야 한다. Req-Target는 사용자가 요청한 서비스의 대상이 되는 객체(Object)로 XML schema와 문서(document)에 대한 XPath 표현으로 정의된다. 또한 Req-Operation은 Req-Target에 대해 어떤 연산이 수행될 것인지를 나타낸다.

```
Input: { ID, Req-Target, Req-Operation }
```

그림 4. 사용자 요청 값

5.2 사용자 할당(User Assignment) 과정

역할기반 접근제어(Role-Based Access Control) 정책에서 사용자는 하나 이상의 역할(Role)을 할당받는다. 사용자의 역할은 사용자가 서버에 보낸 요청 정보(User-Req) 중 ID 값을 분석하여 해당 사용자에게 할당하게 된다. 이 경우 역할 할당은 사용자 정보 저장소에 정보를 이용한다. 사용자 정보 저장소는 사용자에게 부여된 ID와 역할에 대한 정보를 담고 있다. 그림 5는 사용자 요청 정보(User-Req)의 ID값과 일치하는 사용자 정보 저장소의 역할 값을 읽어오는 과정을 보여준다.

```
① input : { ID }
   ⇩
② output : Role{ R1, R2, ···, Rn }
```

그림 5. 사용자 할당(User Assignment)

그림 5에서 ①은 사용자 서비스 요청단계에서 사용자로부터 전송받은 사용자 식별 값, ②에서, Role{}은 단일 사용자에게 부여되는 역할들의 모음인 역할 집합을 의미하며, R_1, R_2, …,R_n 사용자에게 부여된 역할 값들을 나타낸다. 단일 사용자는 한개 이상의 역할을 동시에 부여받을 수 있으며 Role{∅}인 경우는 존재하지 않는다. 즉 모든 사용자는 하나 이상의 역할을 부여 받아야 한다.

5.3 허가 할당(Permission Assignment) 과정

사용자는 사용자 할당 과정을 통해 하나의 이상의 역할을 부여 받았다. 하지만 이들 역할이 실제 객체에 대한 연산을 수행할 수 있도록 권한을 할당 받지 못한다면 사용자는 어떤 연산도 수행할 수가 없다. 따라서 각 역할에는 XML 데이터 객체에 대한 허가를 할당해주어야 한다. 그림 6은 해당 역할에 접근권한부여 과정을 보여준다.

③은 사용자 할당 단계에서 사용자가 부여받은 역할 집합, ④,⑤,⑥은 특정 역할 Rn에 할당된 권한부여 객체와 연산에 대한 목록을 보여준다. 예를 들어 $PA(R_1)$={document(1)-root/element1, read}와 같이 나타난 경우 현재 역할 R_1에 XML 문서 document(1)의 element-1에 대해 읽기 연산 권한이 부여되었음을 나타낸다.

```
③ input : Role{ R1, R2, ···, Rn }
  ⇩
output :
④ PA(R1)={document(1)-root/element-1, read}
⑤ PA(R2)={document(2)-root/element-2, write}
  ...
⑥ PA(Rn)={document(n)-root/element-n, delete}
```

그림 6. 허가 할당 (Permission Assignment)

5.4 권한부여(Authorization) 과정

권한부여 과정에서는 사용자의 역할에 부여된 객체 연산 범위와 사용자가 요청한 객체에 대한 연산을 비교를 하여 실제 객체에 대한 접근을 부여할 것인지에 대한 여부를 결정한다. 그리고 이들 여부에 대한 결과 값을 사용자에게 전송한다. 그림 7은 사용자의 요청에 대한 권한 부여과정을 나타낸다.

⑦은 첫 번째 입력 값으로 사용자 서비스 요청 과정에서 부여받은 값으로 요청 객체와 요청 연산타입

```
⑦ input(1) : {Req-Target, Req-Operation}
⑧ input(2) : {document(1) - root/element-1, read}
⑨            {document(2) - root/element-2, write}
              ...
⑩            {document(n) - root/element-n, delete}
             ⇩
⑪ output : if input(1)∩input(2)≠∅ then
⑫            {Req-Target, Req-Operation, +}
⑬       if input(1)∩input(2)=∅ then
⑭            {Req-Target, Req-Operation, -}
```

그림 7. 요청에 대한 권한부여(Authorization)

을 나타내고 ⑧,⑨,⑩은 허가 할당 과정의 결과로, 허가를 부여받은 객체와 연산타입, ⑪은 첫 번째 입력 값(input(1))과 두 번째 입력 값(input(2))을 비교하여 동일한 값이 있으면 사용자 서비스 요청 단계에서 전송된 요청 객체와 요청연산에 권한부여를 허가하는 "+" 기호를 나타낸다. ⑬,⑭는 첫 번째 입력 값(input(1))과 두 번째 입력 값(input(2))을 비교하여 동일한 값이 없으면 사용자 서비스 요청 단계에서 전송된 요청 객체와 요청연산에 권한부여를 거부하는 "-" 기호를 나타낸다.

6. 참조 구현 및 제안된 방식의 고찰

6.1 RBACFX를 적용한 접근제어 시스템의 구현

그림 8은 병원에서 제안된 접근제어 시스템을 적용한 예를 보여준다. 그림 9에서 붉은 점선으로 처리된 부분에서 사용되는 정보는 관리자에 의해 사전에 처리가 필요한 부분임을 나타낸다. 즉, 사용자 정보와 역할에 대한 정의, 접근 제어 대상 객체에 대한 등급은 보안 정책, 정보의 민감성에 따라 관리자에 의하여 사전에 정의되어야 하는 부분이다.

그림 9는 병원의 환자 정보를 보여주는 원본 인스턴스 문서이다. 원본 문서에서 <Patient Name="Ban">인 환자를 담당하는 의사는 <Doctor> Dr.Kim </Doctor>과 간호사 <Nurse> Miss. Kim</Nurse>이며, 해당 의사와 간호사는 각자에게 지정된 접근제어 규칙에 따라 그림 11, 13과 같은 최종적으로 접근이 허용된 문서가 반환 된다.

정상적인 로그인이 이루어진 사용자는 사용자 정보를 바탕으로 5.2절에서 기술된 방법으로 관리자에 의해 미리 정의된 역할을 할당 받는다. 이 과정을 통해 Dr.Kim이라는 사용자는 의사의 역할을 부여 받게

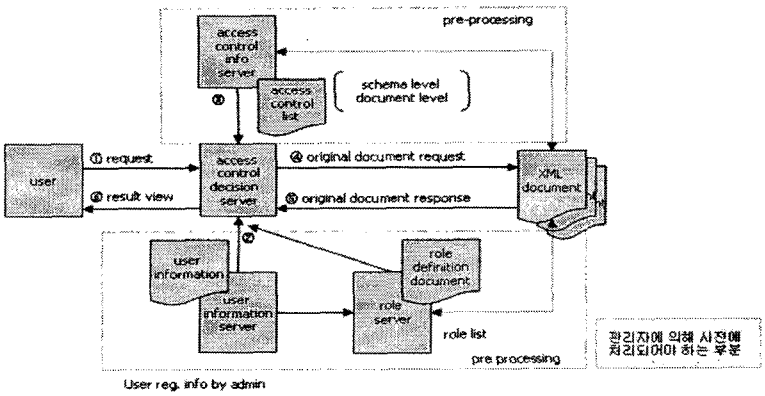

그림 8. RBACFX를 적용한 접근제어시스템의 구성 예

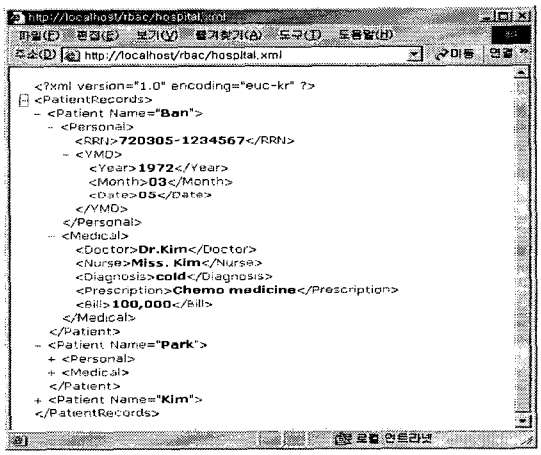

그림 9. D 병원의 환자 정보 - 원본 문서

된다. 의사의 역할이 부여된 사용자에 대하여 5.3절과 5.4절에 에서 언급된 메커니즘을 통해 그림 10과 같은 접근제어 규칙이 생성된다. 생성된 접근 규칙은 스키마 레벨, 인스턴스 레벨, 엘리먼트 레벨에서의 접근가능 여부를 기술하고 있다. 그림 10에서 의사의

```
/-- access Rule of Doctor- Dr.Kim --/
/-- rule in  schema level --/
<DOC, PaRec.xs:PatientRecords, read, +>
/-- rule in instance level --/
<DOC, PaRec.xml:PatientRecords/Patient="Ban", read, r>
/-- rule in element level --/
Patient/personal/RRN, read, -
Patient/personal/YMD, read, r
Patient/Medical, read, r
Patient/Medical/Bill, read, -
/--end of Dr.Kim's access Rule --/
/--last update in 04.10.04.01.30-PM--/
```

그림 10. Dr.Kim에게 부여된 접근 규칙

역할을 부여받은 Dr.Kim이라는 사용자는 PaRec.xs라는 스키마 파일에 접근이 허용되어 있다. 또한 RaRec.xs를 기반으로 생성된 인스턴스 문서인 PaRec.xml의 Patient 엘리먼트 속성이 "Ban"인 엘리먼트의 하위 엘리먼트에 접근이 허가되었다. 엘리먼트 단위의 규칙에서는 Patient/personal/RRN에 대한 접근이 제한되어 있지만, YMD 엘리먼트와 하위 엘리먼트까지 접근이 허용됨을 보여준다. 비용에 관련된 정보인 Medical의 Bill 엘리먼트는 접근이 허용되지 않음을 보여준다.

그림 12는 Nurse 역할을 부여받은 사용자 Miss.kim에 대하여 생성된 접근제어 규칙이다. 목록에서 Nurse 역할은 환자의 <RRN> 엘리먼트에 대한 접근이 금지되어 있음을 알 수 있다. 그림 12에서 정의된 규칙에 따라 그림 13과 같은 접근 가능한 결과 문서를 얻을 수 있다.

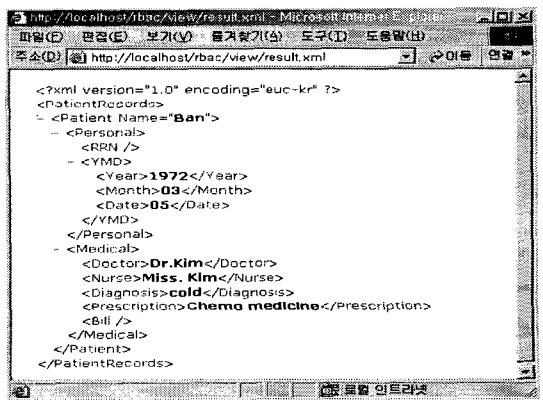

그림 11. Dr.Kim에게 허용된 문서

```
/-- access Rule of Nurse- Miss.Kim --/
/-- rule in schema level --/
<NUR, PaRec.xs:PatientRecords, read, +>
/-- rule in instance level --/
<NUR, PaRec.xs:PatientRecords/Patient="Ban",  read, r>
/-- rule in element level --/
Patient/personal, read, -
Patient/Medical, read, r
Patient/Medical/Bill, read, -
/--end of Miss.Kim's access Rule --/
/--last update in 04.10.04.02.32-PM--/
```

그림 12. 간호사 Kim에게 부여된 접근 규칙

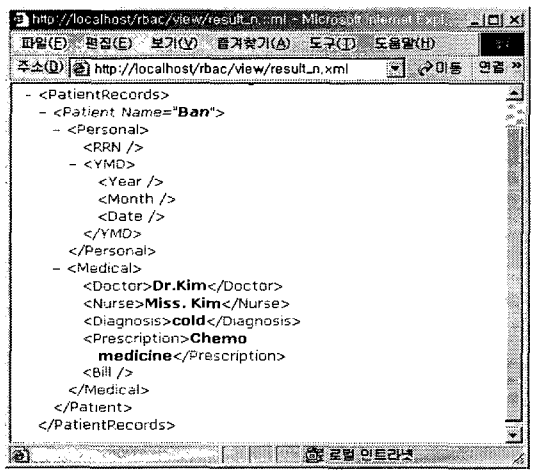

그림 13. 간호사 Miss.kim에게 허용된 문서

6.2 제안된 방식의 고찰

본 절에서는 본 논문에서 제안된 모델이 가지는 주요 특징을 기존 모델과 비교하여 본 논문에서 제안한 모델의 특징과 필요성을 고찰한다. 비교의 기준은 2절에서 제시된 XML 문서의 보안 요구사항의 만족 여부와 접근제어의 단위, 접근제어 수행의 주체, 객체, 접근 제어 수준, 연산, 엘리먼트의 표현, 기반 모델, 사용되는 정책의 수, XML 저장소의 사용여부 및 메커니즘 적용 방식이다. (제안된 논문들은 접근제어를 위한 개념적 모델들이므로 수행 속도등과 같은 수치적 비교는 고려하지 않았다.)

■ 접근제어의 단위

본 논문에서 제안하는 접근제어 모델과 메커니즘은 Ravi S. 등이 제안한 RBAC96 모델을 확장하여 XML 문서에 대한 접근제어를 수행하는 기법이다. RBAC96 모델은 문서의 일부가 아닌 전체 문서에 대한 접근을 허가하거나 반대로 전체 문서에 대한 접근을 거부하도록 하는 것에 기본적인 초점을 둔 모델이다. 그러나 본 논문에서 제안하는 모델은 접근하고자 하는 XML 문서 구조를 해석하여 각 엘리먼트 별로 접근제어를 수행 할 수 있도록 함으로써 저장소에 저장된 전체 문서뿐만 아니라 문서의 일부분에 대한 접근제어를 수행할 수 있도록 하였다. 즉, 본 논문에서 제안된 모델은 RBAC96에서 고찰되지 않았던 광역 접근제어(스키마 단위) 개념 및 지역 접근제어(엘리먼트 단위) 개념이 혼합되어 보다 세밀한 접근제어를 수행할 수 있도록 설계된 모델이다.

■ 접근제어 수행 주체

최근까지 XML 문서에 대한 접근제어를 수행하기 위한 여러 가지 접근방법들이 제안되었다[2-4,7,9-12,14,15]. 그러나 기존에 진행된 연구들이 가지는 공통적인 단점은 XML 문서에 대한 접근제어 수행 주체를 개별 사용자로 정의하고 있다는 점이다. 사용자와 접근제어 대상 객체의 관계를 1대1로 지정하는 형태로 접근제어 규칙을 정의하고 적용함으로써 사용자와 접근제어 대상객체 사이에서 1대1의 접근제어는 가능하지만, 대규모의 사용자 또는 대규모의 스키마 문서와 이를 따르는 인스턴스 문서를 가지는 환경에서는 적용이 불가능하다는 한계를 가진다. 그러나 본 논문에서 제안된 모델과 메커니즘은 접근제어 수행 주체를 사전에 부여된 역할 그룹으로 정의하고, 역할 계층에 대하여 명시적으로 접근 가능한 스키마의 목록을 정의함으로서 역할을 부여 받은 사용자가 접근 할 수 있는 스키마 목록이 실시간으로 정의되고, 스키마를 따르는 인스턴스 문서에 대한 접근 목록을 얻을 수 있도록 정의함으로써 대규모의 사용자 또는 대규모의 스키마 문서와 이를 따르는 인스턴스 문서를 가지는 환경에서 적용이 가능하다.

■ 접근제어 대상 객체와 규칙의 수

이전에도 RBAC를 기반으로 XML 문서에 대한 접근제어를 수행하고자하는 개념적 모델이 제안되었다[5,6,16]. 그러나 RBAC를 기반으로 XML 문서에 대한 접근제어를 수행하는 이전의 제안들과 본 논문에서 제안된 모델은 다음과 같은 차이점이 존재한다. Hao He 등이 [5]에서 제안한 논문은 레퍼지토리에 저장된 XML 문서에 대하여 RBAC 모델을 적용하여 접근제어를 수행하는 초기의 모델이다. 그러나 [5]에서 제안된 논문이 가지는 단점은 접근제어 수행의 대상을 인스턴스 문서 내부의 엘리먼트에 대

한 접근을 수행하기 위한 연산만을 제공하고 있다는 점이다. 또한 역할을 부여 받는 사용자에 대한 정의와 역할 정보, 역할 계층 등이 하나의 설정 파일에 모두 포함되어 사용자에 대한 변경이나 역할 정보에 대한 변경이 발생시 설정 파일을 일일이 재구성하여야 한다는 단점을 가지고 있다. 이러한 방식은 스키마를 따르는 인스턴스 문서가 생성 될 때 마다 인스턴스 문서에 대한 접근제어 규칙을 새롭게 지정해야 함을 의미한다. 만약 이러한 방식에 의해 접근 제어 규칙을 정의 한다면, 사용자의 수와 인스턴스 개수에 비례하여 규칙의 수가 증가 한다. 그러나 본 논문에서 제안된 접근제어 방식은 스키마에 대한 접근과 스키마를 따르는 인스턴스 문서에 대한 접근제어를 동시에 고려하여 접근제어를 수행하도록 하므로 광역접근제어 및 인스턴스 레벨 접근제어가 가능하다. 개별 인스턴스 문서에 대한 접근은 할당된 역할과 사용자의 개별 정보에 따라 결정되도록 함으로써 같은 역할 내에서도 권한 정보에 따라 접근 할 수 있는 인스턴스와 인스턴스 내부의 특정 엘리먼트(노드)에 대한 접근이 제한되도록 하였다. 이와 같은 방식을 적용함으로써 얻어지는 장점은 정책 결정에 필요한 권한의 수가 감소하게 된다. 사용자 (U명), 전체 문서 (D개), 각 문서의 노드 수 (N개) 중 역할 (R개)에 부여된 권한의 수(P개)는

최대 정책 수(Max_P) = 지정 가능한 역할 U개 * 문서 D * 노드 수 N이 되고(=기존 방식),

최소 정책 수(Min_P) = 지정 가능한 역할 1개 * 문서 1개 * 노드 수 N이 되므로, 제안된 방식에서 정의되는 정책 수가 줄어 들 수 있음을 알 수 있다. 즉, 제안된 방식은 다수의 사용자 또는 문서의 양이 급속도로 증가되는 환경에서 효율적으로 적용 가능하다는 장점을 가진다.

■ 동일한 스키마를 따르는 서로 다른 인스턴스 문서에 대한 접근제어

Xinwen Zhang등이 [6]에서 제안한 모델은 RBAC를 XML 문서의 접근제어에 적용한 또 다른 모델이다. 이 모델에서는 권한부여 서버가 역할 서버(Role Server)로부터 역할정보와 요구되는 속성을 획득 후, 권한 부여 서버는 목표 XML 문서와 그 문서가 정의된 전체 스키마를 요구하는 방식으로, 각 XML 인스턴스는 유효한 스키마를 가진다고 가정하고 접근 제어를 수행한다. 이러한 방식은 본 논문에서 제안된 방식과 접근 제어를 수행하는 기법이 매우 유사하다. 그러나 본 논문에서 제안된 모델은 [6]에서는 고려되지 않았던 엘리먼트의 내용에 의해 사용자별로 개별 인스턴스에 접근할 수 있도록 함으로써 동일한 스키마를 따르는 별도의 인스턴스 문서에 대한 접근제어

표 2. 제안된 모델과 기존 모델과의 비교

비교 항목	RBAC [1]	E. Damiani [3]	He [5]	X. Zhang [6]	Jingzhu [16]	제안된 모델
기반 모델	RBAC	DAC	RBAC	RBAC	RGM	RBAC
접근제어 단위	문서단위	엘리먼트	엘리먼트	스키마 엘리먼트	엘리먼트	스키마 문서 인스턴스 문서 엘리먼트
접근제어 수행 주체	역할	개별사용자	역할	역할	역할	역할
스키마 접근제어	×	×	×	○	×	○
접근제어 대상 객체	개별 문서	인스턴스 문서	인스턴스 문서	스키마 엘리먼트	엘리먼트	스키마 문서 인스턴스 문서 엘리먼트
기본 연산	읽기/쓰기	읽기	읽기	읽기/쓰기	읽기	읽기/ 쓰기
문서에 대한 레이블링 과정	불필요	필요	불필요	불필요	필요	불필요
엘리먼트 표현	없음	XPath 사용	Xpath 사용	XPath XQuery	엘리먼트 이름을 이용	XPath 사용
사용되는 규칙의 수	가장 작음	가장 많음	많음	작음	많음	작음
XML DB지원 여부	×	×	○	○	×	○
광역 접근제어 지원여부	○	×	×	○	×	○
인스턴스 단위의 접근제어	×	×	×	×	×	○

까지도 고려하였다. Jingzhu Wang 등은 [16]에서 Role Graph Model과 OODB(Object Oriented database)를 결합하여 XML DB에 대한 접근제어 모델을 제안했다. 그러나 이 방식은 기존의 RBAC의 역할관계를 그래프로 표현한 것으로 기존 접근제어 방식에 비해 접근제어 수행 면에서 많은 제약을 가지고 있다. 표 2에서 기존에 제안된 방식과 논 논문에서 제안된 방식의 주요 특징을 비교 분석한 결과를 제시한다.

7. 결 론

본 논문에서는 XML 문서가 웹 환경에서 안전하게 보호될 수 있도록 하는 접근제어 모델과 메커니즘을 제안하였다. 본 논문에서 제안된 방식은 XML 문서에 접근하고자 하는 사용자의 접근권한을 미리 설정된 보안정책에 따라 판단하여 해당 사용자가 가진 권한에 따라 해당 문서에 대한 접근 범위를 결정하여 전체 문서에 대한 접근 허용 또는 거부뿐만 아니라, 문서를 구성하는 각각의 요소에 대한 수준별 접근제어를 수행하는 방법에 대한 연구 결과를 제시하였다. 본 논문에서 제안된 접근제어 모델과 메커니즘은 이전에 제안된 RBAC를 확장하여 사용자를 역할에 따른 그룹으로 하여 구성하고, 그에 따라 XML 문서에 대한 접근권한을 부여하는 방법이다. 본 논문에서 제안된 방식은 일반적인 자원 또는 HTML 문서에 적용되는 접근 방법과 달리 문서의 최소 단위인 엘리먼트 계층에서 문서 소유자의 보호 수준을 만족시키면서 적절한 접근권한을 가진 사용자에게 해당 XML 문서에 대한 접근과 변경을 허용하도록 하였다. 향후에 추가적으로 연구되어야 할 부분은 다음과 같이 요약된다. 먼저 제안된 모델을 기반으로 구성되는 시스템을 위하여 클라이언트의 요청에 대한 권한 부여를 자동화하기 위한 기법 연구가 필요하다. 또한 XML 문서에 대한 접근제어 수행 시 발생할 수 있는 예외처리 및 제약조건에 대한 정의와 규정에 대한 연구가 수행되어야 한다.

참 고 문 헌

[1] Ravi S. Sandhu, E,J.Coyne, and H.L.Feinstein, "Role-based Access Control Modesl," *IEEE COmputer*, Vol. 29, No. 2, pp. 33-47, 1996.

[2] Elisa Bertino, Silvana Castano, Elena Ferrari, and Marco Mesiti, "Specifying and Enforcing Access Control Policies for XML Document Sourecs," *World wide Web Journal*, Vol. 3, No. 3, pp. 139-151, 2000.

[3] E. Damiani, S. D. C di Vimercati, S. Paraboschi, and P. Samarati, "A fine-grained access control system for XML documents," *ACM Transaction on Information and System Security*, Vol. 05, No. 02, pp. 169-202, 2002.

[4] E. Damiani, S. D. C di Vimercati, S. Paraboschi, and P. Samarati, "Controlling access to xml documents," *IEEE Internet Computing* Vol. 5, No. 6, pp. 18-28, November 2001.

[5] Hao He and Raymond K. Wong, "A Role-Based Access Control Model For XML Repositories," *First International Conference on Web Information Systems Engineering*, pp. 138-145, 2000.

[6] Xinwen Zhang, Jaehong Park, and Ravi Sandhu, "Schema based XML Security: RBAC Approach," *IFIP WG 11.3 Working Conference on Data and Applications Security*, 2003.

[7] R.Chandramouli, "Specification and Validation of Enterprise Access Control Data for Conformance to Model and Policy Constraints," *7th World Multi-conference on Systemics, Cybernetics and Informatics*, 2003.

[8] D.Ferraiolo, R.Sandhu, S.Gavrila, D.R.Kuhn, and R.Chandramouli, "Proposed NIST Standard for Rolebased Access Control," *ACM Trans. Inf. Syst. Security*, Vol. 4, pp. 224-274, Aug. 2001.

[9] Elisa Bertino and Elena Ferrari, "Secure and Selective Dissemination of XML Documents," *ACM Transactions on Information and System Security(TISSEC)*, Vol. 5, No. 3, pp. 290-331, Aug. 2002.

[10] Sabrina De Capitani di Vimercati, "Web and e-business application: An authorization model for temporal XML documents," *Proceedings of the 2002 ACM Symposium on Applied*

computing(SAC'02), pp. 1088-1093, March 2002.

[11] M. Kudo and S. Hada, "XML Document Security based on Provisional Authorization," *In Proc. of the Seventh ACM Conference on Computer and Communication Security (CCS'00)*, pp. 87-96, November 2000.

[12] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. "Design and implementation of an access control processor for XML documents," *Computer Network*, pp. 59-75, June 2000.

[13] R. Chandramouli, "Specification and Validation of Enterprise Access Control Data for Conformance to Model and Policy Constraints," *Cybernetics and Informatics*, SCI 2003.

[14] E. Bertino, M. Braun, S. Castano, E. Ferrari, and M.Mesiti. "Author-X: A Java-based system for XML data protection," *Proc. of 14th IFIP WG11.3 Working Conference on Database Security*, pp. 15-26 August 2000.

[15] Vaibhav Gowadia and Csilla Farkas, "RDF Metadata for XML Access Control," *Proceedings of the 2003 ACM workshop on XML security(XMLSEC'03)*, pp. 39-48, 2003.

[16] Jingzhu Wang and Sylvia L. Osborn, "A Role-Based Approach to Access Control for XML Databases," *SACMAT'04*, pp. 70-77, 2004.

[17] James Clark, Steve DeRose, "XML Path Language (XPath) Version 1.0," http://www.w3.org/TR/xpath, 1999.

[18] www.w3c.org, "eXtensible Markup Language (XML) 1.0," W3C Recommendation, 04 February 2004.

[19] www.w3c.org, "XML-Signature Syntax and Processing," W3C Recommendation, 12 February 2002.

[20] www.w3c.org, "XML Encryption Syntax and Processing," W3C Recommendation, 10 December 2002.

[21] OASIS, "eXtensible Access Control Markup Language Version 1.1," 24 July 2003.

[22] www.w3c.org, "XML Schema," http://xml.coverpages.org/schemas.html.

반 용 호

1998년 동서대학교 전자공학과 졸업(공학사)
2000년 동아대학교 컴퓨터공학과 졸업(공학석사)
2003년 동아대학교 컴퓨터공학과 박사수료

관심 분야 : 암호이론, 접근 제어, XML 보안, 홈 네트워킹 등

김 종 훈

1974년 동아대학교 전자공학과 졸업(공학사)
1977년 동아대학교 전자공학과 졸업(공학석사)
1986년 경북대학교 전자공학과 졸업(공학박사)
1986년~현재 동아대학교 컴퓨터공학과 교수

관심 분야 : 암호이론, 접근제어, HW/SW 통합설계, 홈 네트워킹 등