XML 문서의 접근제어를 위한 RBAC의 응용

반 용호, 심 효영, 김 종훈
동아대학교 컴퓨터공학과

Application of RBAC for Access Control of XML Document

YongHo Ban, HyoYoung Sim, JongHun Kim
Dept. of Computer engineering, Dong-A University

요 약

XML(eXtensible Markup Language)는 웹상에서 문서를 표현하고 교환하기 위한 표준으로 자리 잡았다. XML 문서는 자체적으로 민감성의 등급이 다른 정보를 포함할 수 있는 구조를 가지고 있으므로 XML 문서에 대한 특정 사용자 그룹의 선택적인 접근 및 공유를 위한 방법이 반드시 필요하다. 이를 위해서는 XML문서에 대한 접근제어 정책을 규정하고 수행하기 위한 방법과 메커니즘이 요구된다. 또한, XML 문서에 사용되는 접근제어 메커니즘은 사용자가 소유하고 있는 권한부여 정보에 의존하여 문서의 안전하고 선택적인 배포가 가능해야 한다. 본 논문에서는 XML 문서의 안전하고 선택적인 접근 문제를 해결하기 위하여 RBAC를 응용한 메커니즘을 제안한다.

I. 서론

XML(eXtensible Markup Language)[1]은 네트워크 상에서 교환되는 구조화된 문서 표현 방법이다. XML은 일반 사용자 또는 기업 환경에서의 애플리케이션 통합 및 웹 서비스 등에서 정보교환을 위한 중요한 수단으로 여겨지고 있다. 이런 추세에 대응하여 XML이 실제 응용 서비스에 적용되었을 때 적절한 안전성을 보장하기 위한 몇몇 방법들이 제시되고 있다. 특히 XML에 대한 암호화와 전자서명 등은 국제적인 표준이 제정되는 등 많은 연구가 이루어진 상황이다[2][3]. 그러나 HTML과는 달리 XML을 이용한 서비스는 B2B나 전자입찰 또는 XML/EDI와 같은 소수의 사용자가 한정된 서비스 환경에서 특정 정보를 공유하기 위해 사용되고 있는 것이 대부분이다. 즉, 구조화된 문서인 XML을 사용하는 것은 특정 그룹에서 특정 정보를 공유하기 위함이므로 XML 형태로 정의된 정보에 대한 권한별 관리를 위한 메커니즘이 반드시 정의 되어야 한다. 특히, 입찰정보나 계약금액, 계약자 이름 혹은 인사정보와 같은 민감한 데이터를 여러 그룹의 사용자가 공유하여 사용하는 환경에서는 핵심적으로 고려되어야 하는 사항이다.[5][6][7][8] 그러므로 인터넷과 인트라넷 환경에서 운용되는 많은 XML 응용 서비스들에서 사용되는 XML 자원에 대한 유연한 접근제어를 제공할 수 있는 메커니즘과 보안 정책이 필요하다. XML 자원에 대한 보안은 많은 수의 사용자와 XML 객체를 가진 기업 환경에서 특히 필수적으로 고려되어야 하는 매우 복잡하고 난해한 문제로 간주되고 있다. 이들에게 있어서 보안 문제는 시스템 설계에서 가장 먼저 고려해야 할 핵심요소이기 때문이다. 이 논문에서는 역할기반 접근제어(RBAC)[4] 모델을 바탕으로 XML 문서에 대한 접근 권한을 관리하기 위한 보안 정책을 제안한다. 본 논문의 나머지 부분은 다음과 같이 구성된다. 2절에서는 XML 문서에 대한 보안 요구사항과 접근제어를 위해 제안된 이전의 연구를 살펴본다. 3절에서는 RBAC의 특징을 살펴보고 이를 확장한다. 4절에서는 XML 문서의 보호를 위해 확장된 RBAC를 기반으로 추상화된 시스템의 설계를 정의하고, 최종적으로 결론 및 향후 연구방향을 제시한다.

II. 관련 연구

2.1 XML 문서의 보안 요구 사항

XML 문서에 대한 보안 요구는 XML 문서의 특성에서 민감성의 정도가 다른 문서를 포함 할 수 있으므로 다양한 보안 계층이 지원되어야 한다는 사실로부터 발생한다. XML 문서를 위한 접근 제어 메커니즘은 경우에 따라 최소한의 단위로 보안 정책을 적용할 수 있도록 하는 충분한 유연성을 가져야 한다. 즉, XML 문서에 대한 접근제어 정책은 XML 문서차체 뿐만 아니라, 해당 문서가 포함하는 다양한 형태의 정보(loot, element, sub-element, attribute) 및 link로 연결된 외부 문서에 대하여 접근 제어 정책이 전파되어 원하는 요소에 대한 접근제어 정책이 즉시 적용될 수 있어야 한다.

XML 문서는 미리 정의된 문서 형태로 항상 구성될 수 없다는 사실로부터 발생한다. 일반적인 접근 제어 정책은 문서 유형의 관점에서 정의되는 경우가 대부분이므로, 어떤(위치에 있는)문서에 대한 현재의 접근제어 정책을 적절하게 운용함으로서 처리되지 않을 상황이 발생 할 수 있다. 문서의 교환과 획득 과정이 웹을 통해 빈번히 발생하므로, 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 요구 된다.[9][10]

2.2 RBAC

RBAC(Role-based Access Control)의 중심적인 개념은 사용자가 기업이나 조직의 정보 자원을 임의로 접근할 수 없도록 하는 것이다. 대신에 접근 권한이 역할(role)에 부여되고 사용자는 적절한 역할에 소속됨으로써 역할의 수행에 필요한 최소 자원만을 접근할 수 있도록 한다. 이러한 아이디어는 권한 관리를 매우 단순화 시켜주고 기업의 특정한 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. 사용자는 그들의 업무적 권한과 책임에 따라 특정 역할의 구성원이 되며 접근 구조의 변경이 없이도 역할의 변경을 쉽게 할 수 있다. 다음 그림은 RBAC의 기본 모델이다. 기본 모델은 사용자(U : user), 역할(R : role), 인가권한(P : permission), 세션(S : session)으로 구성되어 있다.

- 사용자(user)와 역할(role)

사용자는 컴퓨터 시스템을 통하여 시스템 내의 정보를 사용하는 객체로서 한 사용자는 한 명의 사람에 대응된다. 역할은 접근제어 정책을 구현하는 중요한 의미적 구조이다. RBAC 시스템에서는

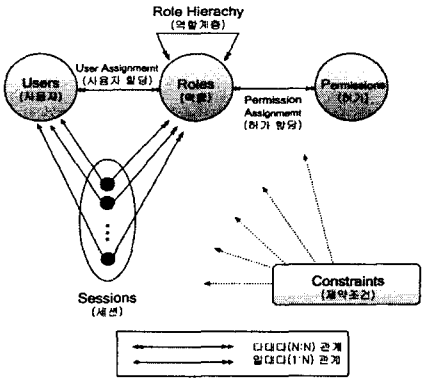

그림1. RBAC96 역할기반 접근제어 모델

시스템 관리자가 회사나 조직의 업무 기능에 따라 역할을 생성하고 역할에 권한을 부여한다. 역할 계층(RH : role hierarchy)은 관련성이 있는 역할들 간의 부분순서(partial order) 관계로서 정의되며 기업의 권한과 책임의 체계와 매우 유사하여 기업의 권한체계를 모델링 하는데 매우 적합하다.

- 인가권한(permission)

인가권한은 시스템의 하나 또는 그 이상의 객체에 대한 특정 접근 모드(예 : read, write, update)의 승인을 나타낸다. RBAC에서의 인가권한(permission)은 권한허가(authorization), 접근권리(access right), 권한(privilege)과 같은 의미를 갖는다. 여기서 객체는 기업 또는 조직의 정보 시스템을 구성하고 있는 자료(data)나 시스템 자원(system resource)을 말한다. 인가권한은 네트워크 수준으로부터 특정 레코드의 특정 필드에 대한 접근 단위에 이르기까지 다양한 레벨, 다양한 범주로 주어질 수 있다.

- 세션(session)

사용자는 시스템에 로그인 등을 통해 그들이 가진 역할의 부분집합을 활성화할 때 세션을 형성한다. 각 세션은 하나의 사용자와 여러 개의 권한을 사상시킨다. 이중 화살표는 다중 역할이 동시에 활성화한다는 것을 말한다.

- 사용자 배정(user assignment)과 인가권한 배정(permission assignment)

사용자 배정과 인가권한 배정은 다대다 관계이며 RBAC 모델에서 매우 중요한 구성요소이다. RBAC의 특징 중의 하나는 사용자가 정보 객체들에 대해서 실행할 수 있는 연산들을 직접 사용자에게 부여하는 대신 조직의 업무 수행에 필요한 역할에 배정하고(인가권한 배정), 사용자는 해당

역할의 구성원이 됨으로써(사용자 배정) 정보 객체에 대해 지원하는 연산을 수행하도록 하는 것이다. 이러한 방법은 사용자와 정보 객체수가 많은 일반 기업 환경에서 권한의 관리를 매우 용이하게 수행할 수 있는 장점을 제공한다.

Ⅲ. XML을 위한 RBAC의 응용

3.1 XML 문서의 접근 권한 부여 유형

접근 권한을 정의 하고 관리하기 위해서 권한 유형과 계층 구조, 권한의 주체 계층에 대하여 정의 한다. XML 문서에 대한 권한 유형은 크게 두 가지로, XML 문서 자체에 대한 권한 부여 유형과 XML 문서 내의 element에 대한 권한 부여 유형으로 나눌 수 있다.

- XML Schema와 XML 문서에 대한 권한 부여
  - 스키마 생성(Schema Document Generate): XML 문서의 정의 부분인 스키마 문서를 읽을 수 있고, 스키마 문서를 생성 할 수 있음을 의미한다.
  - 스키마 읽기(Schema Document Read): XML 문서의 정의 부분을 읽을 수 있다.
  - 인스턴스 생성(Instance Document Generate): 스키마 문서를 읽을 수 있고, 이를 기반으로 인스턴스 문서를 생성 할 수 있다.
  - 인스턴스 읽기(Instance Document Read): 스키마 문서와 인스턴스 문서를 읽을 수 있다.
  - 인스턴스 수정(Instance Document reWrite): 스키마 문서와 인스턴스 문서를 읽을 수 있을 뿐만 아니라, 생성된 인스턴스 문서에 대한 수정이 가능하다.

스키마 문서의 권한 유형과 인스턴스 문서의 권한 유형에서는 스키마 문서의 권한이 인스턴스 문서의 권한을 지배하고 있으며, 이들의 관계를 계층 구조로 표현 할 수 있다. 이 경우 상위 권한 유형은 하위 권한 유형에 대하여 묵시적인 지배 권한을 가지게 된다. 스키마 문서와 인스턴스 유형을 계층 구조로 나타내면 다음과 같다. 권한 유형별 우선순위는 다음과 같이 정의 된다.

$$SDG > IDW > IDG = IDR > SDR$$

- XML 문서 내부의 각 노드(엘리먼트, 속성)에 대한 권한 부여 유형
  - 엘리먼트 읽기(Element Read): XML 문서에 정의된 각 노드의 엘리먼트 혹은 속성을 읽을 수 있다.
  - 엘리먼트 수정(Element reWrite): XML 문서의 엘리먼트나 속성을 수정할 수 있다.

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. 권한 유형별 우선순위는 다음과 같이 정리된다.

$$IDW > IDR = EW > ER$$

3.2 XML을 위한 RBAC의 확장

본 논문에서 제안된 접근제어정책은 RBAC를 다음과 같이 확장하여 정의한다. 사용자(user), 역할(role), 역할 계층(role hierarchy), 사용자-역할 배정(user-role assignment), 세션(session)은 RBAC 모델에서와 동일하다. 역할과 최종 권한 부여의 직접적인 배정 대신에, XML 문서를 정의하는 스키마(또는 DTD)에 대한 권한 부여(schema based permission)와 역할과 스키마 객체 사이에서의 명시적인 역할-권한 배정(explicit role-permission assignment)을 사용한다.

- SP $\subseteq$ OT x SO : Schema-based Permission
- IP $\subseteq$ OT x IO : Instance-based Permission
- PM : SO $\rightarrow$ IO : Permission mapping

정의 1. Schema Objects(SO)
*Schema Object*는 XML Schema 또는 Schema component(s) 이며, XPath 표현에 의해 생성된다.

정의 2. Instance Objects(IO)
*Instance Object*는 XML Instance 또는 Instance component(s)를 의미한다.

정의 3. Operation Type(OT)
*Operation Type(OT)*는 SO 또는 IO에 대한 권한부여 과정에 있어서 사용자 별로 해당 객체에 대한 다양한 작업을 수행할 수 있는데, 이들 작업에 대한 유형을 연산(Operation Type)이라고 한다.

정의 4. Permission Mapping (PM)
*Permission Mapping(PM)*은 하나의 XML Schema는 자신의 Schema 구조를 따르는 여러 개의 Instance 문서를 생성할 수 있다. 따라서 특정 Schema와 이

들 Schema를 따르는 Permission 역시 관계를 가지는데, 이들 사이의 연관성을 Permission Mapping 이라고 한다.

정의 5. Schema Permission Assignment(SPA)
Schema Permission을 역할에 할당하는 과정을 말한다. 하나의 Schema Permission에 여러 개의 역할을 부여할 수 있으며, 한 개의 역할 역시 여러 개의 Schema Permission을 가질 수 있으므로 다대다(N:N)관계를 가진다.

정의 6. Instance Permission Assignment(IPA)
Instance Permission을 역할에 할당하는 과정이다. Instance Permission Assignment와 마찬가지로 역할 컴포넌트와 다대다(N:N)관계를 가진다.

SOH(Schema Object Hierarchy)는 보안 관리자에 의해 정의된 스키마 객체사이의 부분적인 순서로, 낮은 계층의 객체 상에 정의된 권한부여가 상위 계층 객체로 전달되게 된다. 일부 제약들이 SPA와 IPA에 대하여 정의된다. SPA는 역할과 SP사이에서 정의되는 명시적인 역할의 권한 배정을 의미 하며, IPA는 묵시적인 역할의 권한배정을 의미한다.

정의 7. Schema Object Hierarchy(SOH)
SOH는 스키마 객체간의 부분적인 순서 관계를 말한다 : $SOH \subseteq SOx\ SO$.

일반적으로 SOH는 스키마 객체사이의 재사용 관계를 기반으로 한다. 재사용 관계는 비순환적이며 재귀적인 관계를 가지는 부분적인 순서로서 간주될 수 있다. 여러 유형의 재사용 메커니즘들이 W3C XML Schema에 정의되어 있다. DataType은 기본적인 스키마 타입으로 구성된다.

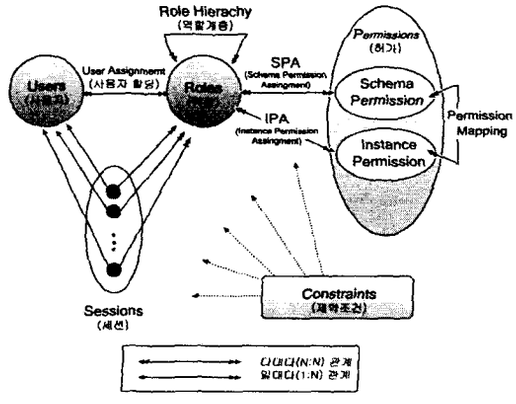

그림2. XML 문서를 위한 접근제어 모델

스키마의 엘리먼트와 속성이 그들의 "type" 이름을 규정하여 이들 기본적인 타입으로 생성되어 질 수 있다. *DataType derivation*은 새로운 데이터 타입이나 엘리먼트는 동일한 스키마 또는 다른 스키마에서 정의된 존재하고 있는 데이터 타입으로부터 유도될 수 있다. 동일한 스키마의 또 다른 엘리먼트 또는 다른 스키마로부터의 참조에 의하여 생성될 수 있다.

Ⅳ. 메커니즘의 추상적 설계

여기서는 앞에서 설명된 시스템에 대한 구체적인 제안이 아니라 추상적인 설명으로 시스템의 동작을 설계하고 이를 설명한다. 실제 환경에서 XML 객체는 이질적이고 다양한 서버와 조직으로부터 다른 XML 스키마를 기반으로 하고 있다. 그러므로 방대한 수의 스키마 컴포넌트와 권한 할당(permission)이 존재하게 될 것이다. 컴포넌트와 퍼미션 간의 복잡한 관계는 RPA(role permission assignment)를 어렵게 만들 수 도 있다. XML 요청과 응답은 XML 메시지로, 형식은 스키마로 정의되고, 보안 정보의 전송을 위해서 기존의 표준으로 정의된 SAML(Security Assertion Markup Language)을 사용하여 사용자 인증, 사용자-역할 할당 요청/응답이 이루어진다. 사용자 인증과, 역할 및 정책 결정을 위한 서버의 하부 메커니즘은 추상화하여 순수한 XML로 정의할 수 있도록 한다.

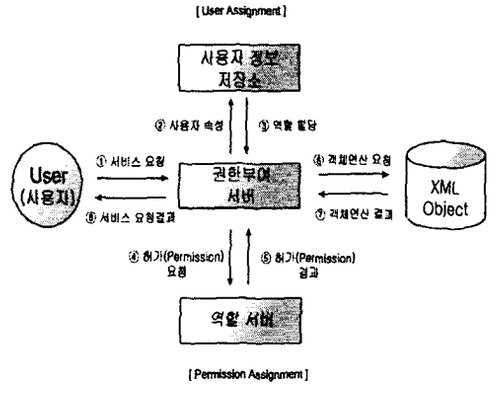

그림 3. 접근제어 메커니즘

사용자의 접근요청을 위해 요구되는 정보는 사용자 등록 정보, 사용자에 의해 활성화되는 역할의 집합, 접근의 유형, 요청문서로 이루어진다. 경우에 따라서 접근제어 결정의 출력이 예상되는 스

키마를 만족하도록 요구된다. 이를 위해 접근하고자하는 타겟 문서가 해당 스키마 문서의 보안 권한 설정에 만족하는지 여부를 검사한다.

권한 부여 프로세스는 입력 객체의 일부 노드를 제거 할 수 있으므로, 출력은 어떤 특정 스키마를 만족시키지 못 할 수 있는데, 이 경우에 문서에 대한 접근이 거부된다.

역할은 직접적으로 할당되거나 역할 계층을 가지는 상속을 포함하여, 사용자에게 할당된 역할의 집합을 반환한다. 목표문서를 트리 형태로 반환하여 각 서버 트리에 대하여, 먼저 root 노드에 대한 permission을 검사한다. 만약, 접근이 허용되어 있다면, 모든 하위트리가 동일한 메커니즘에 의해 검사된다. 아니라면, 전체 엘리먼트와 하위-엘리먼트에 대한 접근이 거부될 것이다.

V. 결론 및 연구방향

본 논문에서는 최근 그 필요성이 크게 인식되고 있는 XML 문서에 대한 보호 방안을 접근제어 기법을 적용하여 해결하고자 기존에 제안된 RBAC를 확장하여 사용자를 역할에 따른 그룹으로 하고 구성하고, 그에 따른 접근권한을 부여하는 메커니즘을 제안하였다. 본 논문에서 제안된 방식은 일반적인 자원 또는 HTML 문서에 적용되는 접근 방법과 달리 XML 문서가 가지는 구조적 특성을 충분히 활용하여 XML 문서의 각 element 레벨까지 소유주의 보호 권한을 만족하면서, 적절한 사용권한을 가진 사용자에게 해당 XML 문서에 대한 접근과 변경을 수행하는 방식으로 처리되도록 하였다.

향후 추가적으로 연구되어야 할 부분은 다음과 같이 요약된다. 먼저 권한부여 규칙들에 대한 보다 세밀한 정의가 이루어져야 한다. 그리고 실제 환경에서는 많은 수의 스키마 객체와 역할이 존재하기 때문에, 퍼미션-역할 배정을 수행하기 위한 시각적 도구의 개발이 필요하다.

참고문헌

[1] W3C. "Extensible Markup Language (XML) 1.0". World Wide Web Consortium(W3C). http://www.w3c.org/TR/REC-xml (October 2000).

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

[3] W3C. "XML Encryption Syntax and Processing". W3C Recommendation. http://www.w3.org/TR/xmlenc-core(December 2002)

[4] http://csrc.nist.gov/rbac/

[5] Specifying and Enforcing Access Control Policies for XML Document Sourecs, Elisa Bertino, Silvana Castano, 2000

[6] C. Ilioudis, G. Pangalos and A. Vakal-i, "Security Model or XML data".Proceedings of the 2nd International Conference on Internet Computing, 2001

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

[8] E. Damiani, S. D. C di Vimercati, S. Paraboschi, P. Samarati, "Controlling access to xml documents" IEEE Internet Computing 5(6):18-28, November 2001.

[9] Xinwen Zhang, Jaehong Park, Ravi Sandhu, "Schema based XML Security: RBAC Approach", 2003

[10] 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 (SCI 2003)