융통성 있는 XML 관리를 위한 접근제어 정책

조선문*

요약

XML은 SGML에서의 복잡성을 제거하고 HTML에서의 고정된 태그의 한계에서 벗어나 사용자가 문서 구조를 정의할 수 있게 한다. 기존의 접근제어는 XML 문서들을 파싱하여 트리로 생성하는 과정에서 DOM 트리가 메모리상에 적재되어야 한다. XML 문서 검색에 있어서 사용자가 XML 문서의 특정 항목에만 접근할 수 있도록 권한을 부여함으로써, 사용자의 권한 범위에 해당하는 데이터만을 제공한다. 이를 위해서는 XML 문서로부터 사용자의 권한에 따라 권한이 허용되지 않는 부분은 제거하고 접근이 허가된 부분만을 전달한다. 본 논문에서는 효율적인 접근제어 정책을 제안한다.

Access Control Policy for Flexible XML Management

Sun-Moon Jo*

ABSTRACT

XML as a simplified dialect of SGML overcomes the limitations of HTML with a fixed set of tags and thereby allows its users to define their own document structures. The existing access control requires DOM trees to be loaded on memory in the process of parsing XML documents to generate the trees. This provides data only corresponding to its users' authority levels by authorizing them to access only the specific items of XML documents when they're searching XML documents. In order to do this, the XML eliminates certain parts of documents which are inaccessible and transmits parts accessible depending on its users' authority levels. In this paper we propose an efficient access control policy.

Key Words : XML, Access Control, Authorization, Policy, Security

* 배재대학교 교양교육지원센터 (✉sunmoon@pcu.ac.kr)
· 제1저자(First Author) : 조선문 · 교신저자(Correspondent Author) : 조선문
· 접수일(2011년 5월 22일), 수정일(1차 : 2011년 6월 10일), 게재확정일(2011년 6월 14일)

Ⅰ. 서 론

XML 문서를 파싱하고 접근에서는 정규식을 표현하여 수행하는 파싱은 단순한 XML 문서의 경우 파싱을 쉽게할 수 있고, 파싱 속도가 빠른 이점이 있다. 이는 단일 DTD에 대한 응용프로그래밍이며 다른 DTD에 대해서는 재사용할 수 없다는 단점이 존재한다. 이 경우 정규식을 지원하지 않는 프로그래밍언어로는 이를 처리하기 어렵다는 등의 문제점이 존재한다. 따라서 XML 프로세서는 여러 DTD에 대한 XML 문서의 파싱을 위해 특별하고 일반화된 방법을 제공할 수 있어야 한다. API는 프로세서에 따라 자체 API를 제공하는 방법과 표준화된 API를 이용하는 방법이 있으며 표준화된 대표적인 API로 SAX와 DOM이 있다[1].

XML 기반 접근제어 기술의 목적은 인터넷상의 접근제어 서비스를 위한 다양한 제품과 제품들이 서로 다른 환경에서 일관되게 적용될 수 있는 권한부여 정책을 제공하고 정책을 통하여 다양한 환경 및 방식을 가진 접근제어 제품도 상호운영이 가능하도록 하는 것이다.

최근 웹 상에서 권한부여 기반 접근제어를 지원하는 몇 가지 프로젝트가 실행되고 있지만 이용할 수 있는 권한부여 및 접근제어는 준비 단계에 있다. 예를 들어, 아파치 서버는 사용자나 호스트, 호스트, 사용자 목록을 포함한 구성 파일을 통한 접근제어 목록의 명세를 허용하는데 이는 허용과 거부만 하는 서버 접속이다. 제안된 기법은 XML에 대한 모델과 마찬가지로 데이터를 의미론적으로 구조화할 수 있는 언어를 기반으로 하지 않는다는 사실에서 한계가 있다. 따라서 권한부여를 관리하는 일이 매우 어렵다.

본 논문은 XML 문서에서 단순한 권한(허용과 거부)을 넘어서 좀 더 유연하게 정책을 기술하고 선택할 수 있도록 접근제어 정책을 연구한다. 또한 쓰기 연산과 연산 수행에 뒤따르는 DTD의 변경에 관해서도 고려한다.

논문의 구성은 2장에서는 관련 연구로서 XML의 기본 구성과 기존의 접근제어의 문제점을 서술한다. 3장에서는 접근제어를 개선하기 위한 문서 접근제어 정책을 정의한다. 4장에서는 접근제어 정책 평가를 기술한다. 끝으로 5장에서는 결론과 향후 연구에 대해서 기술한다.

Ⅱ. 관련연구

XML 문서의 기본 구성은 요소이다. 요소는 깊이에 관계없이 내포될 수 있으며 다른 요소들을 포함할 수도 있다. 회사 부서에 관한 정보가 들어 있는 XML 예는 그림 1과 같다. 이 문서는 생산 부서 사원들의 이름, 주소, 이력서, 급여, 의료 기록을 제공한다. 이 그림에서 부서 요소는 문서 요소 즉 문서의 모든 요소를 담고 있는 가장 바깥쪽 요소이다. 사원, 주소, 이메일은 XML 문서의 계층 구조에서 깊이가 다른 요소들이다.

RBACM(Role-Based Access Control Models)에서 제안한 시스템은 역할을 부여 받는 사용자에 대한 정의와 역할 정보, 역할 계층 등 정보를 하나의 설정 파일에 모두 포함하여 사용자에 대한 변경이나 역할 정보에 대한 변경이 발생시 설정 파일을 일일이 재구성하여야 한다는 단점을 가지고 있다. 만약 이러한 방식에 의해 접근제어 규칙을 정의 한다면 사용자의 수와 인스턴스 개수에 비례하여 규칙의 수가 증가 한다[2].

XACL(XML Access Control Language)은 주체가 어떤 자원에 대해 액션을 수행할 권한이 있는가를 명세한다. 그러나 XACL은 여러 문제점을 가지고 있다. 확장이 불가능하므로 주체, 자원, 액션 등을 명세할 수 없다. 또한 한 정책은 항상 정확히 한 타깃 XML 문서와 관계가 있다. 정책 계층을 정의할 수는 없으며, 정책 조합은 언어에 의해 지원되지 않는다[7,8].

Cabillon은 XML 문서를 트리로 표현하였다. 사용자는 접속 이름으로 사용할 수 있는 식별자를 가지고 있

다. 주체 계층은 XML 주체 시트에 개별적으로 표시된다. 권한부여 규칙은 주체의 집합, 객체의 집합, 접근, 우선권으로 구성된다. 주체는 사용자이다. 객체는 XML 주체 시트의 요소 주체와 관련하여 주체의 위치 경로로 표현하고 있는 객체의 XPath 트리 노드이다. 접근에 대한 값은 허용과 거부 중 하나이다[3]. Cabillon 기법은 모든 종류의 노드를 보호할 수 없다. 또한 XML 접근제어는 읽기 연산만을 제공한다.

```
<?xml version="1.0" encoding="euc-kr" ?>
<department id="production">
  <employee id="E101">
      ...
  </employee>
  ...
  <employee id="E123" manager="E101">
    <name>
      <firstname> SunMoon </firstname>
      <lastname> Jo </lastname>
    </name>
      <address>
      <street> Paichai University 14 Yeonja-1-Gil, Seo-Gu,
Daejeon, Korea
      </street>
      <tel> 042 520 1234 </tel>
      <tel> 042 520 1235 </tel>
       <email mailto="sunmoon@pcu.ac.kr"/>
      </address>
      <resume>
        <education> ... </education>
        <previous-job> ...
        </previous-job>
      <previous-job> ... </previous-job>
      <skills> ... </skills>
      </resume>
      <salary> 100만원 </salary>
    <medical-dossier> .. . </medical-dossier>
  </employee>
  ...
  <employee id="E150" manager="E123">
  ...
  </employee>
</department>
```

그림 1. XML 문서
Fig. 1 XML Document

기존 연구는 XML 문서의 특징인 정보의 의미에 따라 처리하기 위한 정보의 기반한 접근과 요소와 같이 작은 단위의 접근이 불가능하다. 접근제어를 위한 요구사항은 다음과 같다[4,5,6].

첫째, XML 문서는 웹서버에서 접근제어 시스템을 확장할 수 있어야 한다.

둘째, XML 문서는 보안 민감성의 수준이 각각 다른 요소를 포함할 수 있으므로 이를 만족하는 보안 계층이 지원되어야 한다.

셋째, XML 문서를 위한 접근제어는 미세 단위로 정책을 지원을 해야 한다.

III. 문서 접근제어 정책

3.1 접근 권한부여

그림 2는 인스턴스와 요소 권한 타입의 계층 구조를 나타낸 것이다. IW는 인스턴스 문서를 읽고 생성할 수 있을 뿐만 아니라 생성한 인스턴스 문서에 대한 수정이 가능하다. IR은 문서를 읽을 수 있다. EW는 요소의 데이터를 읽고 수정할 수 있다. ER은 요소의 데이터를 읽고 수정도 가능하다.

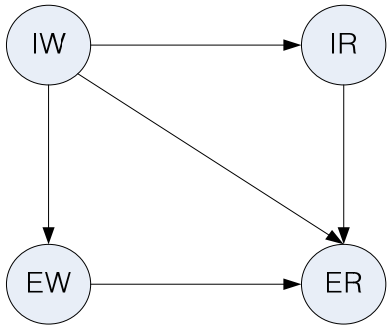

그림 2. XML 요소 권한 타입
Fig. 2 XML Element Authority Type

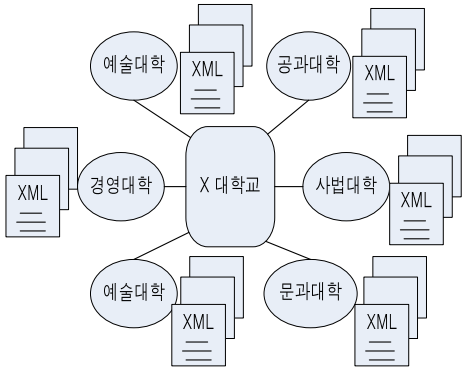

그림 3. X 대학교
Fig. 3 X University

요소에 명세된 권한부여는 요소의 속성에만 적용하거나, 재귀 접근에서는 하위 요소와 속성에 적용할 수 있는 것으로 정의한다. 보완 측면에서 재귀 권한부여는 트리에 있는 노드에서 자손으로 허가와 거부를 전파함으로써 요소의 전체 구조화된 내용에 해당하는 권한부여를 명세화 한다.

권한부여는 단일 XML 문서나 DTD에 명세될 수 있다. 명세된 권한부여는 DTD의 인스턴스인 모든 XML 문서에 적용된다. 조직은 DTD와 관련하여 권한부여를 명세하고 특정 사이트는 DTD 뿐 아니라 개별 문서와 관련하여 권한부여를 명세할 수 있다. 예를 들어, 그림 3에서 X 대학교가 다양한 부서로 구성되어 있고 각각의 부서는 특정 XML 문서를 관리할 책임이 있다고 가정한다. 여기서 X 대학교의 모든 부서에서 충족시켜야 할 일반적 보호 요구사항은 X 대학교 상태에 DTD 수준 권한부여를 통해 표현한다. 부서 내에서만 적용되는 보호 요구사항은 DTD 수준 권한부여를 통해 표현될 수 있다. 이와 비슷하게, 특정 문서에만 적용되는 요구사항은 문서와 관련된 인스턴스 권한부여를 통해 한다. 접근제어 처리에서 조직 DTD 수준 권한부여와 사이트 DTD 수준 권한부여가 각각의 DTD와 관련하여 동일 수준을 수행함으로써 통합한다.

본 논문에서는 권한 충돌 시 우선순위를 결정하는 DTD 수준의 권한부여를 하드로 명세하여 사용한다. 권한부여 타입 간의 우선순위를 지정할 때, 우선순위를 높은 것부터 제안하여 설명하면 다음과 같다: L(로컬), R(재귀), LDH(로컬 하드), RDH(재귀 하드), LD (로컬 스키마 수준에서 상세), RD(재귀는 스키마 수준에서 상세).

본 논문은 브라우징 특권과 저작 특권 두 종류의 특권도 지원한다. 브라우징 특권은 주체가 요소에서 정보를 읽거나 링크를 따라 검색할 수 있게 한다. 읽기 특권은 주체에 요소와 구성요소를 볼 수 있는 권한을 부여한다. 검색 특권은 주체에 특정 링크와 해당 요소에 있는 모든 링크의 존재를 알아보고 그것을 따라 검색할 권한을 부여한다.

허가와 거부 권한부여를 모두 명시할 수 있으며, 주체가 부호는 다르지만 같은 보호 객체에 대한 같은 권한을 놓고 두 가지 권한이 부여된다는 점에서 권한부여 사이에 충돌이 발생할 수 있다. 본 연구에서의 충돌 해소 정책은 다음과 같은 원리를 토대로 한다.

그림 1의 XML에서 권한부여 A1=(Sun, production, read, altg, +, L)는 Sun이 생산으로 식별되는 문서에 포함된 요소, 속성, 링크를 읽을 수 있게 하지만 문서 링크를 통한 검색은 허가하지 않는다. A2=(Minji, production.{E123}, read, altg, +, RDH)가 추가로 명시되었다고 가정한다. 본 연구 정책에 따르면 문서 수준에서 명시된 권한부여는 DTD 수준에서 명시된 권한부여에 우선하므로, E123으로 식별되는 요소에 적용될 때 권한부여가 우선된다. 그 결과, 민지는 E123 요소, 의료 기록 하위 요소를 비롯한 모든 하위 요소를 있는 모든 정보를 읽을 수 있다. 이와 반대로, A3=(jungmin, production, employee, salary, read, altg, -, L)가 명시되었다. 급여 요소와 관련하여 A3는 권한부여 A1보다 우위에 있는데, 이는 문서 계층에서 높은 수준에서 이어지는 것이 아니라 급여 요소 위에 바로 명시되기 때문이다. 그 결과 junmin은 급여 요소

에 있는 것을 제외하고 생산 문서에 있는 모든 정보를 알 수 있다.

본 논문에서는 접근 권한부여 a ∈ Auth는 권한을 6-튜플로 표시한다. XML의 접근 권한은 <subject, object, action, action label type group, sign, type> 타입으로 구성된다.

[정의 1] 접근제어 정책 규칙

Subject ∈ 역할(권한부여 주체)은 사용자 이름, IP 주소, 컴퓨터 이름

Object ∈ 권한이 있는 객체 XPath 1.0 (XML 요소)

Action: read, write, create, delete (주체)

Action Label Type Group(읽기 그룹)

Sign ∈ {+, -}(권한의 부호)

Type ∈ {L, R, LDH, RDH, LD, RD}(권한의 타입)

문서의 수가 많고 보호 요구사항이 다양한 문서인 XML 문서를 보호하면 많은 수의 권한부여를 정의할 수 있다. 소스에서 문서에 대해 정의하고 유지할 권한부여의 수를 제한하기 위해 XML 문서의 트리 구조를 이용하여 권한부여 전파를 시행할 수 있다.

3.2 접근제어 비교

본 논문에서는 XML 문서 데이터에 대한 접근제어에 대해서 기존 연구와 비교하였다. 표1과 같이 비교의 기준은 접근제어 주체, 주체 단위 보호 수준, 접근제어 객체, 내용 접근제어, 전파, 접근 권한 일시적 억제 등으로 구성하였다.

ACP(Access Control Policy)는 XML 문서에 대한 접근 가능 영역을 요소 단위로 접근제어를 지원한다. Gabillon에서 문제점은 문서에 대한 접근제어 주체를 개별 사용자로 이용하고 있다는 점이다. 이것은 1대1의 접근제어는 가능하나 많은 사용자 또는 대규모의 XML 문서와 이를 따르는 인스턴스 문서를 가지는 환경에서는 문제가 발생한다. 또한 Gabillon은 다양한 데이터 전달 전략에 따른 접근제어는 일시적 억제를 고려하지 않는다.

표. 1 접근제어 비교

Table 1. Access Control Comparison

요구 조건	ACP (접근제어 정책)	Gabillon
접근제어 주체	주체, IP 그룹 자격	주체 그룹
주체 단위 보호 수준	XPath 요소와 속성	XPath 요소와 속성
접근제어 객체	인스턴스 문서 요소	인스턴스 문서 요소
내용 접근제어	예	아니오
전파	예	서브 트리 정책
접근 권한	검색, 삽입, 삭제, 수정	읽기
일시적 억제	예	아니오

Ⅳ. 접근제어 정책 평가

본 논문에서는 DOM API의 자바 구현 서비스를 이용하여 자바로 ACP의 프로토타입을 설계했다. 도구로는 인텔 펜티엄 3.2GHz, 하드디스크 320GB, 메모리 2GB, Windows XP 운영체제에서 인터넷 익스플로러 8.0, Java 6.0을 이용하였다.

인증을 거친 사용자가 원하는 자원을 요청하게 되면 ACP에서 사용자의 권한을 확인하고 요청한 자료를 제공할 것이지 아닌지를 결정한다. 사용자의 권한이 XML 문서의 전체 혹은 일부에 권한이 주어졌다면 XML 문서의 변환 과정을 거친 후 사용자에게 전달한다.

그림 4는 minji가 로그인하여 권한부여 설정에 따라 결과를 보여주는 예이다. minji는 전화번호를 볼 수 없다(권한부여 sign value="-"/). 또한 카드 정보도 볼 수 없다.

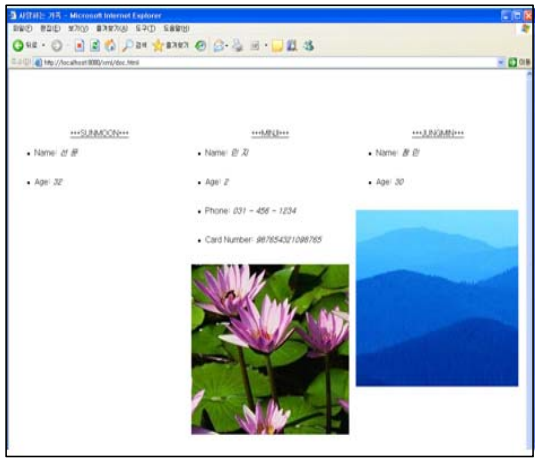

그림 4. 권한부여 결과
Fig. 4 Authorization Result

그러나 minji와 관계되는 권한부여가 아래와 같다면,

```
<subject>minji|*|*</subject>,
<object>/people/person[./name="minji"]</object>,
 <altg value="rlg" />,
 <action  value="read"/>,
  <sign value="+"/>,
  <type value="RDH"/>
```

minji는 본인과 관계되는 정보를 모두 볼 수 있는 권한이 설정되므로 전체의 정보를 볼 수 있다.

다중 스레드를 확보하기 위해서는 클래스를 스레드에 안전한 타입으로 구현하고 각 요청의 매개변수를 모두 특정 방법 환경에 격리시켜야 한다. 성능을 강화하는 구조적 솔루션은 사용자 계층을 관리하는 스레드를 분리하는 것이다. 해당 사용자에게 권한부여가 적용되는지 여부를 계산하는 서비스는 사용자가 직접, 또는 간접적으로 권한부여 주체에 명세된 그룹에 속하는지 평가해야 한다.

성능 평가는 XML 벤치마크로부터 XML 문서와 DTD를 다양한 접근 통제로 사용되는 약 17,000개의 노드에 대하여 기존에 연구와 ACP에 대하여 XML 문서 접근제어를 비교하였다[10].

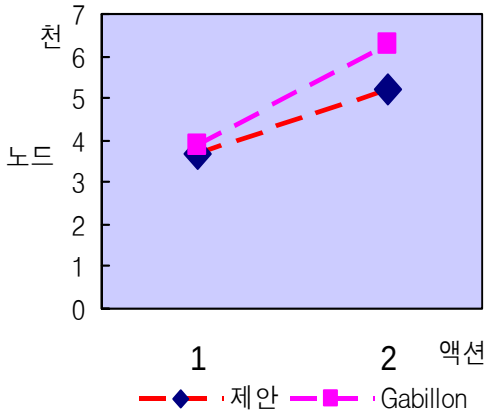

그림 5. XML 노드 평가
Fig. 5 XML Node Evaluation

그림 5에서와 같이 두개 실행 모드에 관하여 사용자에 대해 ACP와 Gabillon으로 구성하였다. 사용자에 대한 ACP 레이블 수와 Gabillon 노드 수의 비를 비교하였다. Gabillon은 접근 권한을 데이터에 따라서 접근제어 정보뿐 아니라 문서 노드에 있는 정보를 포함하기 있기 때문에 본 논문에서 제안하는 ACP에서는 XML 문서 부호화에 접근제어 정책을 이용하여 문서의 변화 노드 접근제어 정보를 하나만 저장한다. 그러므로 Gabillon의 노드수 보다는 적게 발생한다.

Ⅴ. 결론

본 논문은 XML 문서를 위한 융통성 있는 문서 관리를 위한 접근제어 정책을 정의하고 설계하였다. 접속허가, 거부에 대한 지원뿐 아니라 사용자 ID와 관련하여, 보안 마크업은 예외를 지원하면서 다양한 보호 요

구사항을 표현할 수 있다. 객체의 입도는 XML 문서 내의 단일 요소나 속성만큼 정교할 수 있다. 또한 정책은 일반적인 사용자나 객체에 대한 권한을 부여할 때뿐만이 아니라 동일한 객체에 대한 여러 주체들의 권한 충돌이 발생할 때도 활용된다.

향후 연구로는 기존의 연구를 바탕으로 앞으로 수행해야 할 일은 접근제어 정책 시스템을 좀 더 효율적으로 향상시키는 것이다. 또한 XML 문서를 이용하는 다른 응용에 관한 연구가 요구된다.

참고문헌

[1] David Megginson, "Simple API for XML(SAX)", http://www.megginson.com/SAX/index.html, 1999.

[2] S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," IEEE Computer, Vol.29, No.2, pp. 38-47, 1996.

[3] A. Gabillon, and E. Bruno, "Regulating access to XML documents", InProceedings of the Fifteenth Annual IFIP WG 11.3 Working Conference on Database Security, 2001.

[4] X. Zhang, J. Park, and R. Sandhu, "Schema based XML Security: RBAC Approach", IFIP WG 11.3, pp. 300-343, 2003.

[5] S. De Capitani di Vimercati, “An authorization model for temporal XML documents”, Proceedings of the 2002 ACM Symposium on Applied computing, pp. 1088-1093, 2002.

[6] C. A. Ardagna, E. Damiani, S. D. Capitani di Vimercati, and P. Samarati, "A Web Service Architecture for Enforcing Access Control Policies", Elsevier B.V, 2005.

[7] S. M. Jo, K. T. Kim, H. J. Kouh, and W. H. Yoo, "Access Authorization Policy for XML Document Security", Proceedings of International Symposium on Parallel and Distributed Processing and ApplicationsISPA Workshops 2005, pp. 589-598, 2005.

[8] S. Mohan, A. Sengupta, and Y. Wu, "A Framework for Access Control for XML", ACM Transactions on System and Information Security, pp.1-38, 2006.

[9] M. Murata, A. Tozawa, M. Kudo, and S. Hada, "Xml Access Control using Static Analysis", ACM Transactions on Information and System Security, 2006.

[10] The XML benchmark project, Available form: <http://www.xml-benchmark.org>.

저자소개

조선문(Sun-Moon Jo)

2001~2005년 세븐시스템 연구기획팀 팀장

2007년 인하대학교 컴퓨터정보공학과 (공학박사)

2006년~현재 배재대학교 IT교육 교수

※ 관심분야 : XML, 정보보안, 프로그래밍 언어