웹 서비스를 위한 데이터 접근 제어의 정책 시스템
Policy System of Data Access Control for Web Service

조선문*, 정경용**
배재대학교 IT*, 상지대학교 컴퓨터정보공학부**

Sun-Moon Jo(sunmoon@pcu.ac.kr)*, Kyung-Yong Chung(kyjung@sangji.ac.kr)**

요약

접근 제어 기법은 모든 보호 입도 수준을 지원할 만큼 충분히 유연해야 한다. 또한 접근 제어 정책은 문서 타입과 관련하여 명세 될 가능성이 매우 높으므로, 문서가 기존의 접근 제어 정책에 의해 다루어지지 않는 상황을 적절하게 관리해야 한다. XML 문서에 관해서도 단순한 권한부여를 넘어서 좀 더 유연하게 정책을 기술하고, 선택할 수 있는 접근 제어 방법을 고려해야 한다. 본 논문은 XML 문서 접근을 위한 권한부여와 효율적인 관리를 위한 접근 제어 정책 시스템을 기술하고 설계하여 XML 자체의 능력을 활용하는 방법을 제안한다. 본 논문 시스템의 주요 특징은 특정 XML 문서에서 누가 어떤 접근 특권을 행사할 것인가를 고려하면서 조직 전반에 걸친 정책 관리자와 단일 문서 작성자의 요구를 원만하게 조정하는 것이다.

■ 중심어 : | 권한부여 | 정책 | 보안 | XML | 접근 제어 |

Abstract

Access control techniques should be flexible enough to support all protection granularity levels. Since access control policies are very likely to be specified in relation to document types, it is necessary to properly manage a situation in which documents fail to be dealt with by the existing access control policies. In terms of XML documents, it is necessary to describe policies more flexibly beyond simple authorization and to consider access control methods which can be selected. This paper describes and designs the access control policy system for authorization for XML document access and for efficient management to suggest a way to use the capacity of XML itself. The system in this paper is primarily characterized by consideration of who would exercise what access privileges on a specific XML document and by good adjustment of organization-wide demands from a policy manager and a single document writer.

■ keyword : | Authorization | Policy | Security | XML | Access Control |

I. 서론

XML은 SGML(Standard Generalized Markup Language)에 기반을 둔 단순하고 매우 유연성 있는 텍스트 타입이다. 인터넷 상에서의 데이터 교환 및 표현의 표준으로 사용된 이후, 많은 새로운 데이터들이 XML 타입으로 작성되고, 기존의 데이터들이 XML 타입으로 변환되어 현재는 XML 형태의 데이터양이 크게

접수번호 : #081104-005
접수일자 : 2008년 11월 05일
심사완료일 : 2008년 11월 14일
교신저자 : 조선문, e-mail : sunmoon@pcu.ac.kr

증가하고 있다.

초창기 연구는 접근 제어 정책 권한의 관리자에 의해 객체에 대한 접근 권한이 정해지는 임의적 접근 제어, 정보의 보안 수준과 사용자의 보안 등급에 따라 접근을 제어하는 강제적 접근 제어, 역할과 해당 역할의 권한을 정의하는 역할 기반 접근 제어로 나눌 수 있다[3][13].

기존의 웹 기반의 접근 제어는 파일 단위나 파일의 부분에 권한을 기술하는 것이 가능하다. 그러나 이런 방법은 XML 문서의 정보 의미에 기반을 둔 접근이나 요소와 같이 아주 작은 단위의 접근이 불가능하다. 이러한 접근 제어들은 DOM(Document Object Model) 트리를 이용하여 XML 문서와 DTD(Document Type Definition)의 요소에 접근 권한을 설정한다[4][5][7]. 설정된 접근 권한 정보에 의해 사용자의 XML 데이터 접근을 제어한다.

EIT SHTTP 스키마[8] 같은 몇몇 접근법은 보안 관련 HTML 태그화를 사용하여 문서 내 권한부여를 명시적으로 나타낸다. 모든 문서에는 그 문서에 대한 권한부여를 나타내는 관련 보안 태그가 있다. 이것은 보다 강력한 접근 제어 메커니즘을 구성하기 위한 올바른 방향인 것 같지만, HTML의 근본적인 한계로 인해 정보 구조와 의미론을 충분히 고려할 수 없다.

XML 기반 접근 제어는 인터넷상의 접근 제어 서비스를 위해 서로 다른 환경에서 일관되게 적용될 수 있는 권한부여 정책을 제공하고 정책을 통하여 기존의 다양한 환경에 상호 운영이 가능하도록 해야 한다. 본 연구는 XML 문서를 위한 접근 제어 정책 시스템을 제안하면서, 문서에 대한 권한부여를 나타내는 DTD 및 각각의 XML 문서와 관련된 권한부여 시트를 사용한다. 각각의 권한부여가 바로 XML 문서이므로 접근 제어 정책은 XML 자체의 능력을 활용하고 있다.

현재 XML 문서로부터 HTML 페이지 생성을 가능하게 하는 도구를 사용할 수 있는데 이것은 XML에 대한 접근 제한의 명세를 훨씬 더 중요하게 만들어 접근 제어 개발에 새로운 요구사항을 필요로 한다[2].

첫째, XML 문서는 민감도가 다양한 정보를 포함하여 다양한 보호 입도 수준을 지원해야 한다.

둘째, XML 문서가 항상 사전 정의된 문서 타입에 맞는 것은 아니라는 사실이다. 접근 제어 정책은 문서 타입과 관련하여 명세될 가능성이 매우 높으므로 문서가 기존의 접근 제어 정책에 의해 다루어지지 않는 상황을 적절하게 관리해야 한다.

본 논문의 구성은 2장에서는 관련 연구로서 XML 문서의 기본 구성과 기존의 접근 제어 문제점을 서술한다. 3장에서는 기존의 접근 제어 문제점을 개선하기 위한 접근 제어 정책을 정의한다. 4장에서는 문서 관리를 위한 접근 권한부여를 기술한다. 5장에서는 접근 제어 정책 시스템의 구성도와 성능 평가를 기술한다. 끝으로 6장에서는 결론과 향후 연구에 대해서 기술한다.

II. 관련연구

논문[9]에서의 연구를 기반으로 XML 문서와 DTD를 형식에 맞게 기술한다. $\mathcal{L}_\varepsilon$는 요소 식별자 집합이고, 레이블은 요소 태그와 속성 이름, 값은 속성과 요소 값의 집합이다. XML 문서를 다음과 같이 정의하여 사용한다.

XML 문서는 튜플 $d = (V_d, \overline{v_d}, E_d, \emptyset E_d)$이다. 여기에서, $V_d = V_d^e \cup V_d^a$는 요소와 속성을 나타내는 노드의 집합이다. 각각의 $v \in V_d^e$는 연관된 요소 식별자 $id_v \in \mathcal{L}_\varepsilon$ 가진다. 여기에서 각각의 $V \in v_d^a$는 값 $val \in V$ 값을 연관하여 가진다.

$\overline{v_d}$는 문서 요소(문서 root에서 호출)를 나타내는 노드이다.

$E_d = E_d^e \cup E_d^a \subseteq V_d \times V_d$는 에지의 집합이다. 여기서 $e \in E_d^e$는 요소-하위 요소 관계 또는 IDREF(s) 속성으로 인하여 요소 사이에 링크이고 $e \in E_d^a$ 는 요소-속성 관계를 에지로 표현한다.

$\emptyset E_d : E_d \rightarrow$Label는 에지 레이블링 함수이다.

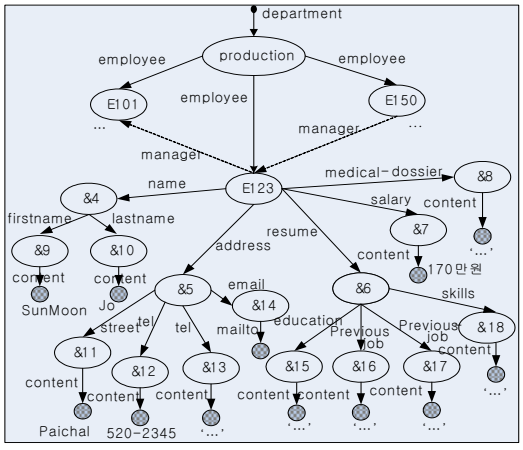

그림 1. XML 문서 그래프 표현

[그림 1]은 XML 문서를 그래프로 표현한 것이다. XML 문서는 요소와 속성 및 이들 사이의 에지 관계를 나타내는 레이블으로 구성된 그래프로 표현한다. ID 속성 값일 수 있는 요소 식별자가 들어 있는 요소를 나타내는 노드는 시스템에 의해 자동적으로 생성될 수 있다.

$\mathcal{L}_{\varepsilon t}$는 DTD 요소 식별자 집합이고, "Label*"을 {*, +, ?}의 부호와 레이블에서 이름의 연결을 통해 얻은 문자열의 집합이다. DTD는 다음과 같이 정의한다.

DTD는 튜플 t = (V_t, $\overline{v_t}$, E_t, $\emptyset E_t$)이다. 여기에서, $V_t = V_t^e \cup V_t^a$ 는 요소와 속성 타입을 나타내는 노드의 집합이고, 각각의 $v \in v_t^e$ 는 연관된 DTD 요소 식별자 $id_t \in \mathcal{L}_{\varepsilon t}$ 가진다. $v \in v_t^a$ 는 타입 $t \in$ type을 가진다.

$\overline{v_t}$는 모든 DTD 요소(DTD root에서 호출)를 나타내는 노드이다

$E_t \subseteq V_t \times V_t$ 는 에지의 집합이면, 여기서 $e \in E_t$ 는 요소-하위 요소 또는 요소-속성 관계를 나타낸다.

$\emptyset E_t : E_t \rightarrow$ Label* $\cup$ {union, content}는 에지 레이블링 함수이다.

Lim[10]에서는 갱신 연산을 지원하는 XML 모델을 제안하기 위해 XML 갱신 연산자를 정의했고, 이를 접근 제어에 포함시켰다. 그러나 이 논문에서는 환경을 가정하여 연구하였다. 예를 들면, XML의 엘리먼트 사이에 의미적 종속성은 없다. 또한 검색보다는 갱신 질의가 매우 빈번하게 발생한다. 이 논문에 문제점은 검색 질의에 대해서 오버헤드가 많이 발생한다.

Gabillon[6]에서는 권한 부여의 의미론을 다양한 것으로 정의하였다. 그러나 이 모델은 모든 종류의 노드를 보호할 수 있는 가능성을 제공하지 않는다. XML 문서의 접근 제어는 읽기 연산만을 제공하고, 접근 제어 시스템의 구현에서 권한의 평가 과정이 복잡하고 응답 시간이 느린 단점이 있다.

XrML[11]은 직접적인 방식으로 다양한 작업 흐름 단계에서 단순한 권리와 복잡한 권리를 모두 표현할 수 있다는 점에서 포괄적이다. 그러나 XrML은 전자 서적, 오디오, 비디오 파일 같은 정적 자원을 다루는데 더 적합한 반면, 수정 가능한 XML 문서와 같은 동적자원을 다루는 데에는 어려움이 있다.

III. XML 문서를 위한 접근 권한부여

문서 기반 정책에 따르면 권한부여는 XML 문서와 직접적인 관계가 있다. 각각의 문서는 보호에 가장 적절한 권한부여를 정의함으로써 따로 처리한다. 예를 들어, 문서에 있는 모든 정보가 동일한 보호 요구를 지니고 있다면, 종속 전파와 함께 문서 요소에는 한 가지 권한부여만 정의한다. 이와 반대로, 문서의 부분마다 보호 요구사항이 다를 경우, 문서 수준에서의 권한부여를 최소한도로 하고 전파 없이 정의한다.

DTD 기반 정책에 따르면 권한부여는 XML 소스의 DTD와 관계가 있으며, DTD 대 인스턴스 전파 관계로 인해 DTD 인스턴스에 전파된다. 권한부여는 DTD의 아주 작은 단위의 보호 수준에서 정의되어 DTD의 유효 문서 인스턴스 내용에 다양한 보호 단계를 제공할 수 있다. 전파를 이용하여 시행할 보호 요구사항에 따라 DTD에 대한 다양한 권한부여를 명세함으로써 문서 보호를 실행한다.

예를 들어, 문서와 관련된 DTD에 대해 DTD 수준에서 한 가지 권한부여만 명세하고, 권한부여가 요소 대 하위 요소 및 요소 대 속성 혹은 링크 관계로 인해

DTD의 모든 하위 요소, 속성, 링크로 전파되고 DTD 대 인스턴스 관계로 인해 DTD의 모든 문서 인스턴스로 전파되도록 함으로써, 주체에게 보호 요구사항이 같은 문서에 접근할 권한을 부여할 수 있다. 포함된 요소와 속성에 대한 보호 요구사항이 같은 DTD는 전파 없이 DTD 전체에 대한 권한부여를 명세하고, 하위 요소와 속성 혹은 링크에 대한 추가 권한을 다수 부여하여 주어지는 특정 권한을 명세함으로써 적절하게 보호한다. 보안 정책을 포함하는 정책 기반은 [그림 2]와 같이 설명한다.

```
〈Policys〉
 〈policy〉
  〈subject〉〈credentialtype="member"/〉
  〈/subject〉
  〈object href="/paper.xml" path="issues"/〉
  〈altg value="rlg"/〉
  〈action name="read" sign="+" propagation/〉
 〈/policy〉
 〈policy〉
  〈subject〉〈credentialtype="nomember"/〉
  〈/subject〉
  〈object="paper.xml" path="/issues"/〉
  〈altg value="rlg"/〉
  〈action name="read" sign="+" propagation/〉
 〈/policy〉
 〈policy〉
  〈subject〉〈credentialtype="nomember"/〉
  〈/subject〉
  〈object href="/paper.xml"
          path="/issues/issuesTuple/
          articles/articlesTuple/abstract"/〉
  〈altg value="rlg"/〉
  〈action name="read" sign="-" propagation =
                no_ propagation"/〉
 〈/policy〉
 〈policy〉
  〈subject〉〈user userid=sunmoon@pcu.ac.kr/〉
  〈/subject〉
  〈object href="/paper.xml"
          path="/issues/issuesTuple/
          articles/articlesTuple[@id=' JSM' ]"/〉
  〈altg value="rlg"/〉
  〈action name="read" sign="+" propagation/〉
 〈/policy〉
〈/Policys〉
```

그림 2. 정책 기반

예를 들어, member는 paper.xml 소스에서 모든 것을 볼 권한이 있고, nomember는 논문 요약을 제외하고 paper.xml 소스에 포함된 문제에 대한 모든 정보를 읽을 권한이 있다고 가정하자. 첫 번째 요구사항은 전파로 전체 소스에서 member에 대해 한 가지 읽기 정책을 명세함으로써 시행된다. 두 번째 요구사항은 nomember에 대해 두 가지 정책을 명세함으로써 충족된다. 첫 번째 정책은 이슈 요소에서 전파가 있는 읽기 권한에 대한 허가 정책이고, 두 번째는 이슈의 추상적 하위 요소에 대한 거부 읽기이다.

그러나 사용자 sunmoon@pcu.ac.kr이 JSM에 의해 확인된 논문에 대한 모든 정보를 읽을 권한이 있다고 가정한다. 보안 관리자는 이 논문에서 전파가 있는 읽기 권한에 대한 식별 기반 허가 정책을 명세함으로써 요구사항을 시행한다.

XML은 계층적인 트리 구조이기 때문에 상위 노드의 권한이 하위 노드의 권한에 영향을 미칠 수 있다. 동일한 사용자라 하더라도 속해있는 그룹, IP 주소, 컴퓨터 이름에 따라서 권한이 달라질 수 있다. 주체가 부호는 다르지만 같은 보호 객체에 대한 같은 권한을 놓고 두 가지 권한이 부여된다는 점에서 권한부여 사이에 충돌이 발생한다. 본 연구의 시스템에 의해 실행되는 충돌 해소 정책과 알고리즘은 기존의 논문을 바탕으로 이용한다[1].

1. 문서 권한부여 주체와 객체 정책

일반적으로 주체는 식별 번호나 요청이 나온 위치를 토대로 언급할 수 있다. 위치는 숫자로 된 IP 주소(예: 203.250.100.87)나 심볼릭 이름(예: lab.pcu.ac.kr)과 관련하여 나타낼 수 있다. 본 논문에서는 IP 주소, 심볼릭 이름을 사용한다. 따라서 접근을 요청하는 주체는 사용자 ID, IP 주소, 심볼릭 주소로 구성된다.

사용자들과 머신에 적용할 수 있는 권한부여 명세를 허가하기 위해 본 논문에서는 사용자 그룹과 위치 패턴을 지원한다. 사용자 그룹은 서버에 정의된 사용자들의 집합이다. 위치 패턴은 기호나 숫자로 된 식별자와 관련하여 물리적 위치를 식별하여 표현한다. 패턴은 특정 명칭이나 숫자 대신 와일드카드(*) 문자를 이용하여 명

세한다.

XML 문서의 내부 구성요소 식별을 위해 W3C에서 제안한 XPath 언어를 사용한다[12]. 표준 언어를 도입하므로 다음의 장점이 있다. 첫째, 언어의 구문과 의미론을 사용자가 잘 알고 있다. 둘째, 함수 시스템을 만들기 위해 쉽게 재사용할 수 있다.

본 논문은 브라우징 권한과 쓰기 권한 두 종류의 권한도 지원한다. 브라우징 권한은 주체가 요소에서 정보를 읽거나 링크를 따라 검색할 수 있게 한다. 읽기 권한은 주체에 요소와 구성요소를 볼 수 있는 권한을 부여한다. 검색 권한은 주체에 특정 링크와 해당 요소에 있는 모든 링크의 존재를 알아보고 그것을 따라 검색할 권한을 부여한다. 주체는 링크의 목적지 요소에 접근할 권한이 있을 경우에만 링크에 대한 검색 권한을 실행할 수 있다.

쓰기 권한은 주체가 요소 내용을 수정하거나 요소에 새로운 정보를 추가할 수 있게 한다. 추가 권한은 기존의 정보를 삭제하지 않고 주체가 요소에 정보를 쓰거나 요소에 링크를 포함한다. 이와 반대로 쓰기 권한은 주체가 요소 내용을 수정하고 요소에 링크를 포함시킬 수 있게 한다. 주체가 어떤 요소에 대한 쓰기 권한이 있으면 요소를 삭제할 수도 있다.

예를 들면, [그림 1]의 XML 문서를 살펴보자. 권한부여 Al=(Sun, production, read, altg, +, L)는 Sun이 생산으로 식별되는 문서에 포함된 모든 요소, 속성, 링크를 읽을 수 있게 하지만 문서 링크를 통한 검색은 허가하지 않는다. 권한부여 A2 =(Sea, production. {E123}.{manager}, Read, Altg, -, L)는 Sea으로 하여금 사원 E123의 관리자가 누구인지 알지 못하게 한다. 권한의 주체는 id나 접근을 요청한 위치로 기술 된다. 객체는 접근을 보호하려는 자원을 의미한다. 액션은 주체가 수행할 수 있는 연산이고, 부호는 권한의 허가 혹은 거부에 대한 표현이면, Type은 권한의 속성 값을 의미한다. Altg는 문서를 어느 보안 수준까지 허용하는지를 결정할 수 있다.

2. 권한부여와 접근 제어 정책 비교

본 논문의 시스템은 인스턴스 수준에서의 관련 접근 권한부여를 열거하는 XAS(XML Authorization Sheet)와 함께 사용자가 요청하는 유효한 XML 문서를 입력으로 받아들인다. 프로세서 연산은 스키마 수준의 권한부여를 명세하는 관련 XAS와 문서의 DTD가 포함된다. 프로세서 출력은 사용자가 접근할 수 있는 정보만을 포함하는 유효한 XML 문서이다. XAS 및 기타 XML 기반 정보를 일정하게 표현하기 위해 [그림 3]에 제시된 XML DTD를 통해 XAS를 제안한다. XML 문서 및 DTD와 관련된 XAS는 연결하는 문서 밖에 있으면서 링크 자체를 실행 및 관리가 가능한 자원으로 만드는 라인 외 링크를 정의하기 위해 XML XLink 명세의 추상적 성격에 의존하여 위치를 지정한다.

```
〈?xml version="1.0" encoding="euc-kr"?〉
〈!DOCTYPE authorizations [
〈!ELEMENT authorizations (authorization)+〉
〈!ELEMENT authorization (subject, object,
                altg, action, sign, type)〉
〈!ELEMENT subject (#PCDATA)〉
〈!ELEMENT object (#PCDATA)〉
〈!ELEMENT altg EMPTY〉
〈!ELEMENT action EMPTY〉
〈!ELEMENT sign EMPTY〉
〈!ELEMENT type EMPTY〉
〈!ATTLIST authorizations about CDATA
  #REQUIRED〉
〈!ATTLIST altg value (rlg|dsalg))
  #REQUIRED〉
〈!ATTLIST actionvalue
  (read|write|create|delete) #REQUIRED〉
〈!ATTLIST sign value (+|-) #REQUIRED〉
〈!ATTLIST typevalue (L|R|LDH|RDH|LD|RD)
 #REQUIRED〉
]〉
```

그림 3. XML 권한부여 시트 구문

본 논문에서는 XML 데이터에 대한 접근 제어 정책에 대해서 몇 가지 제안을 하였다. 제안된 시스템의 가지는 주요 특징을 기존 시스템과 비교하여 관찰한다. 비교의 기준은 XML 문서의 보안 요구사항의 만족여부와 접근 제어의 주체, 주체 작은 단위의 보호 수준, 접근 제어 객체, 문서 레이블링 과정, 전파, 접근 권한, 일시적인 억제 적용 방식이다(제안된 논문들은 접근 제어를 위한 개념적 모델이므로 수행 속도등과 같은 수치적 비

교는 고려하지 않았다). 기존의 제안이 모두 1장에서 제시한 보호 요구사항을 충족시키는 것은 아니다. 해당 요구사항에 대한 제안들 간의 주요 차이는 [표 1]과 같다.

XML 문서에 대한 접근 제어를 수행하기 위한 여러 가지 접근 방법들이 제안되었다. 그러나 기존에 진행된 연구들이 가지는 공통적인 단점은 XML 문서에 대한 접근 제어 수행 주체를 개별 사용자로 정의하고 있다는 점이다. 사용자와 접근 제어 대상 객체의 관계를 1대1로 지정하는 형태로 접근 제어 규칙을 정의하고 적용함으로 사용자와 접근 제어 대상 객체 사이에서 1대1의 접근 제어는 가능하지만 대규모의 사용자 또는 대규모의 스키마 문서와 이를 따르는 인스턴스 문서를 가지는 환경에서는 적용이 불가능하다는 한계를 가진다.

표 1. 접근 제어 시스템 비교

요구 조건	제안 시스템	Damiani	Hada
접근 주체	주체, IP그룹	주체, IP그룹	주체, ID 역할과 그룹
주체 작은 단위의 보호	XPath에 요소와 속성 명세	XPath에 요소와 속성 명세	XPath에 요소 명세
접근 제어 객체	인스턴스 문서 요소	인스턴스 문서	요소
레이블링	필요	필요	필요
전파	예	예	예
접근 권한	검색, 브라우저, 삽입, 삭제	읽기	읽기, 생성, 삭제
일시적인 억제	예	아니오	아니오

제안하는 시스템은 접근하고자 하는 XML 문서 구조를 분석하여 각 요소 별로 접근 제어를 수행할 수 있도록 함으로써 문서 전체 문서뿐만 아니라 문서의 일부분에 대한 접근 제어를 수행할 수 있다.

Damiani[4]와 Hada[7]에서 제안하는 시스템은 DOM 트리를 이용하여 XML 문서와 DTD에 접근 권한을 설정하고, 설정된 접근 권한 정보에 의해 사용자의 XML 문서의 접근을 제어한다. 그러나 이 방법들은 사용자의 요구에 대해 DOM 트리를 생성하므로 다수의 사용자가 동시에 동일한 데이터에 접근할 때 사용자가 요구하는 데이터에 대해 매번 DOM 트리를 생성해야 하는 단점이 존재한다.

IV. 접근 제어 정책 시스템

본 논문에서는 현재 아파치의 Xalan 툴로 발전한 DOM API를 이용하여 자바로 프로토타입을 설계했다. 구현하기 위한 도구는 Intel Core 2 CPU 1.83GHz, 하드디스크 110GB, 메모리 1GB, Windows XP 운영체제에서 아파치 XML 파서, Tomcat 5.0, 인터넷 익스플로러 7.0, Java 5.0이다.

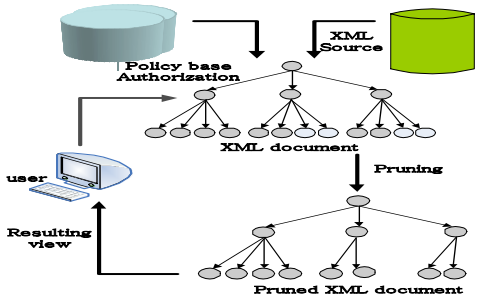

그림 4. 데이터 접근 제어 정책 시스템 구성도

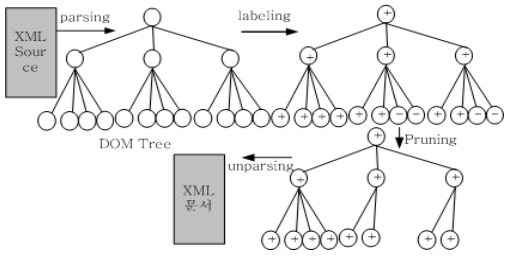

그림 5. 접근 제어 과정

[그림 4]와 같이 본 논문에서 데이터 접근 제어 정책 시스템 구성도의 그림이다. 문서의 모든 객체에 대한 권한부여를 정의하고, 사용자가 XML 구조의 다양한 노드에서 다양한 방식으로 접근하게 할 수 있다. 자신에게 정의된 권한부여에 따라 HTTP를 통해 먼 곳으로부터 XML 자원을 입수할 수 있다. 사용자가 XML 문서의 전체에 권한이 있다면 전체 문서를 사용자에게 보여 준다. 권한이 문서 일부에만 주어졌다면 권한이 없

는 부분을 제거하고 사용자에게 보여 진다.

[그림 5]는 접근 제어 처리 과정이다. 우선 XML 문서를 파싱한다. 접근 권한 정보를 이용하여 DOM 트리에 권한을 설정한다. DTD를 검증 단계에서 구조변경을 하는지 검사한다. 새로운 문서를 생성하기 위해 언파싱을 한다.

성능 평가의 대상은 [4]에서 제안한 XML 접근 제어 기법과 접근 제어 정책 시스템에 대하여 접근 제어 시간을 비교하였다. XML 문서와 DTD는 XML 벤치마크로부터 데이터를 이용하였다[12].

Aut.xas는 XML 문서로 작성되어 있으므로, 접근 제어 정보를 참조하기 위해서는 Aut.xas를 파싱하고 트리를 검색해야 한다. Aut.xas의 파싱 작업은 레이블링 과정에서 접근 권한 정보를 참조하기 위해 필요한 작업이다. XML 문서는 113.8MB(2,035,122 요소 노드)와 DTD인 Test.dtd(5KB), 권한 정보인 Aut.xas(173KB)에 해당하는 값들로 구성된다. 접근 제어를 측정하기 위한 매개 변수는 문서 파싱, 문서 검색, 권한부여 정책, 권한부여 검색으로 구성된다.

문서의 크기가 커짐에 따라서 XML 문서의 파싱과 트리의 검색은 증가한다. 그리고 매개 변수들 중에서 XML 문서 파싱이 시스템의 성능에 가장 큰 영향을 미치는 변수가 된다. 아래와 같이 질의를 사용하였다.

/people/person/phone

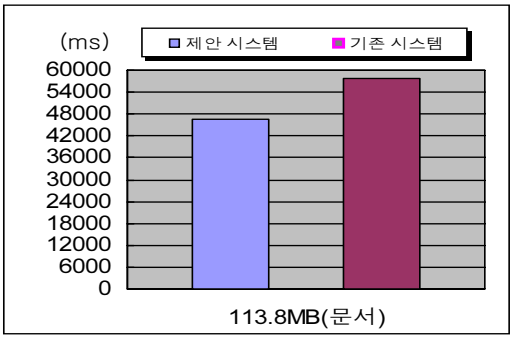

그림 6. 접근 제어 평가

[그림 6]과 같이 접근 제어 시간을 접근 제어 정책 시스템과 기존의 기법과 비교한 결과이다. 질의 경우에서는 기존에서 57,830(ms)의 접근 제어 시간이 요구된다. 그러나 본 논문에서 제안하는 접근 제어 정책 시스템은 46,410(ms)의 시간이 소요된다. XML 문서에서 갱신이 포함된 질의 문에서는 문서와 구조의 변경이 되는지를 검사하기 위해 DTD 검증 과정이 필요하다. 기존에서는 갱신을 수행하여 변경된 XML 문서를 파싱하고 새로운 트리의 모든 노드를 DTD와 비교하면서 검색한다.

V. 결론

본 논문은 XML 문서 접근을 위한 권한부여와 효율적인 관리를 위한 접근 제어 정책 시스템을 설계하여 XML 자체의 능력을 활용하는 방법을 제안하였다.

보안 마크업은 XML 요소의 아주 작은 단위의 보호에서 인스턴스와 스키마 수준의 허가를 제공하는데 사용한다. 접속 허가 및 거부에 대한 지원뿐 아니라 사용자 ID와 관련 그룹 멤버십을 종합해보면, 본 논문의 보안 마크업은 예외를 지원하면서 다양한 보호 요구사항을 쉽게 표현할 수 있게 한다. 권한부여에서 서술한 요구사항을 실행하면 각 의뢰자에게 문서를 보여주는데, 의뢰자가 봐도 되는 정보만을 포함한다. 본 논문의 주체 개념은 식별 번호와 위치로 구성되어 있다. 식별 번호에는 그룹이나 조직 멤버십에 대한 정보가 포함될 수 있다. 객체의 아주 작은 단위의 보호 수준은 XML 문서 내의 단일요소나 속성만큼 정교할 수 있다. 본 논문은 데이터 종속 조건을 포함하고 있으며 제한적 조건 같은 다른 실행 조건을 쉽게 추가할 수 있도록 개방적이고 확장이 가능하다.

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

참 고 문 헌

[1] 조선문, 주형석, 유원희, “안전한 XML 접근 제어

의 정책 설계에 관한 연구", 한국콘텐츠학회논문지, 제7권, 제11호, pp.43-51, 2007.

[2] http://www.w3.org/TR/xmldsig-core/, 2002.

[3] 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.

[4] E. Damiani, S. D. Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Securing XML documents," in Proceedings of the 2000 International Conference on Extending Database Technology(EDBT2000), Konstanz, Germany, pp.27-31, 2000.

[5] 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.

[6] 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.

[7]http://www.trl.ibm.com/projects/xml/xss4j/docs/xacl-spec, pp.1-28, 2002.

[8] http://www.ietf.org/rfc/rfc2660.txt, 1999.

[9] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, "A Query Language for XML," In International Conference on World Wide Web, 1999.

[10] C. H. Lim, S. Park, and S. H. Son, "Access Control of XML Documents Considering Update Operations," In Proceedings of the 10th ACM workshop on XML security, Fairfax, 2003.

[11] http://www.xrml.org, 2001.

[12] A. R. Schmidt, F. Waas, M. L. Kersten, D. Florescu, I. Manolescu, M. J. Carey, and R. Busse, "The XML Benchmark Project," Technical Report INS-R0103, CWI, 2001.

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

저 자 소 개

조 선 문(Sun-Moon Jo) 정회원

- 2007년 : 인하대학교 컴퓨터정보공학과(공학박사)
- 2003년 ~ 2005년 : 세븐시스템 연구기획팀 팀장
- 2006년 3월 ~ 현재 : 배재대학교 IT 교수

<관심분야> : XML, 정보보안, 임베디드 시스템, 프로그래밍 언어

정 경 용(Kyung-Yong Chung) 정회원

- 2000년 2월 : 인하대학교 전자계산공학과(공학사)
- 2002년 2월 : 인하대학교 컴퓨터정보공학과(공학석사)
- 2005년 8월 : 인하대학교 컴퓨터정보공학과(공학박사)
- 2005년 8월 : 한국소프트웨어진흥원 책임
- 2005년 9월 ~ 2006년 2월 : 한세대학교 IT학부 교수
- 2006년 3월 ~ 현재 : 상지대학교 컴퓨터정보공학부 교수

<관심분야> : 유비쿼터스 컴퓨팅, 인공지능시스템, 데이터마이닝, U-CRM, HCI