XML 문서 보안을 위한 효율적인 권한부여 방법

Efficient Authorization Method for XML Document Security

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

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

요약

XML은 스스로 의미 있는 정보를 기술할 수 있는 장점을 이용해 기업체의 데이터베이스나 응용 프로그램의 운영 중에 생기는 많은 데이터들에 대한 정보 교환 형태의 표준 데이터 타입을 제공해 줄 수가 있다. 따라서 대용량 XML 데이터의 효율적인 관리와 보안의 필요성이 점차 중요시되어 XML에 관한 안전한 접근 제어 기법을 개발하는 일은 새로운 방법을 필요로 하고 있다. 본 논문에서는 접근 권한부여 정책들을 정의하여 접근 제어 시스템을 설계한다. 실험 결과, 본 논문에서 제안하는 알고리즘이 기존의 접근 제어 기술에서 복잡한 권한의 평가 과정으로 시스템 성능이 저하되는 문제점이 개선되었다. 따라서 본 논문에서 제안한 안전한 XML 접근 제어 정책이 기존의 접근 제어 방법보다 향상되었음을 보여주었다.

■ 중심어 : | 보안 | 정책 | 권한부여 | XML |

Abstract

XML can supply the standard data type in information exchange format on a lot of data generated in running database or applied programs for a company by using the advantage that it can describe meaningful information directly. Therefore, as it becomes more and more necessary to manage and protect massive XML data in an efficient way, the development of safe XML access control techniques needs a new method. In this study access authorization policies are defined to design access control systems. The findings demonstrated that algorithm suggested in this study improved system performance which was low due to the complex authorization evaluation process in the existing access control techniques. It is consequently proved that the safe XML access control policy presented in this study is in an improved form as compared with the existing access control methods.

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

I. 서론

웹 상에서 표준 데이터 타입으로 XML이 제안된 이후 많은 데이터들이 XML 형태로 작성되고 변환됨에 따라 다량의 XML 데이터가 생성되고 있다. 따라서 대용량 XML 데이터의 효율적인 관리와 보안의 필요성이 점차 중요하게 인식되고 있다. 웹 환경에서 정보는 공용의 네트워크상에서 분산되고 공유된다. 그렇기 때문에 민감한 정보에 대한 권한이 없는 사용자의 접근, 침입, 정보 위조와 같은 공격에서 안전하지 못하다. XML

접수번호 : #090730-005
접수일자 : 2009년 07월 30일
심사완료일 : 2009년 08월 03일
교신저자 : 조선문, e-mail : sunmoon@pcu.ac.kr

문서의 보안에 관한 연구는 XML 문서에 대한 전자 서명 생성 및 검증 기능을 지원하는 XML 전자서명, XML 문서에 대한 암호화 및 복호화 기능을 제공하는 XML 암호화, 공개 키 등록과 위치와 검증을 위한 XML 키 관리 등 다양한 보안 방식을 정의하고 있다[2-4].

기존에 접근 제어는 대부분이 HTML 문서에 대한 것이었다. 이는 파일 단위의 접근 제어로써 XML의 주된 장점인 부분적 정보의 의미에 따른 접근 제어를 다룰 수 없었다. 이에 XML의 장점을 잘 적용한 접근 제어 방법 연구가 필요하게 되었다[5][8]. XML 문서로부터 HTML 페이지 생성을 가능하게 하는 도구를 사용할 수 있는데 이것은 XML에 대한 접근 제한의 명세를 훨씬 더 중요하게 만들어 접근 기법 개발에 새로운 요구사항을 필요로 한다[1][14].

첫 번째는 XML 문서는 민감도가 다양한 정보를 포함하여 다양한 보호 입도 수준을 지원해야 한다는 사실에서 나온다. 어떤 경우에는 여러 문서에 같은 접근 제어 정책이 적용될 수 있고, 어떤 경우에는 같은 문서의 부분마다 다른 접근 제어 정책이 적용될 수 있다. 다른 많은 중간 상황도 발생할 수 있다. 접근 제어는 모든 보호 입도 수준을 지원할 만큼 충분히 유연해야 한다.

두 번째는 XML 문서가 항상 사전 정의된 문서 타입에 맞는 것은 아니다. 접근 제어 정책은 문서 타입과 관련하여 명세될 가능성이 매우 높으므로 문서가 기존의 접근 제어 정책에 의해 다루어지지 않는 상황을 적절하게 관리해야 한다. 웹에서는 문서 교환 및 수집 과정이 빈번할 수 있으므로 이러한 상황에 적절히 대처할 수 있어야 한다.

Samarati는 WWW 문서에 관한 접근 제어를 제안 하였다[7]. 이 모델은 HTML 문서를 고려하고 있으며 링크로 연결된 비구조화 페이지로 구성 되어있다. 권한부여는 문서 전체나 문서 내에서 선정된 부분에 대해 이루어질 수 있다. 주체에게 문서에 대한 접근을 선택적으로 허용한다는 생각을 토대로 했다.

그러나 본 연구는 기존의 제안과 실질적으로 차이가 있다. 차이는 HTML 문서와 관련하여 XML 문서의 보다 풍부한 구조와 그 구조를 설명하는 XML 문서에 DTD를 추가할 수 있는 가능성 때문이다. 이러한 측면에서 HTML 문서에 대해 고안된 것보다 미세한 접근 제어 정책을 정의하고 실행해야 한다. 제안된 접근 제어 모델은 XML에 대한 모델과 마찬가지로 데이터를 의미론적으로 구조화할 수 있는 언어를 기반으로 하지 않는다는 사실에서 유도되는 큰 한계가 있다. 따라서 권한부여를 관리하는 일이 매우 어렵다. 특히 문서의 부분에 접근하고자 할 경우 페이지를 수동으로 여러 부분으로 나누어 서로 다른 권한을 부여해야 한다.

본 논문의 구성은 2장에서는 관련 연구로서 DOM 설명과 기존 접근 제어 문제점을 기술하고 3장에서는 제안하는 XML 문서 보안 명세화 정책에 대해서 기술한다. 4장에서는 효율적인 XML 문서 관리 구조와 성능 평가를 기술한다. 끝으로 5장에서는 결론과 향후 연구에 대해서 기술한다.

II. 관련연구

DOM(Document Object Model)은 HTML 문서 및 XML 문서와 같이 인터넷 문서에 대하여 문서의 내용 및 구조를 객체로 표현하고 객체를 핸들링 할 수 있다. 즉, 동적으로 문서의 내용과 구조, 스타일을 바꿀 수 있게 하는 플랫폼에 독립적이고 언어 중립적인 인터페이스이다. DOM을 사용하는 목적은 어떠한 환경이나 애플리케이션에서도 사용할 수 있는 표준 프로그래밍 인터페이스를 제공하는 데 있다.

Hada는 조건부 액션 개념을 전통적 권한부여 의미론으로 통합함으로써 보다 유연성 있는 권한부여 결정을 하도록 하였다. 조건부의 권한 부여 규칙은 사용자에게 자신의 요청이 권한 부여를 받게 되거나 혹은 시스템이 특정 보안 액션을 취하게하는 규칙이다[6][13]. 그러나 이 논문에서는 개별적으로 속성 혹은 테스트 노드를 보호하는 가능성을 제공하지 않는다. 또한 완전하게 XPath 언어를 활용하지 않는다. 접근 제어 시스템에서는 권한의 평가 과정이 복잡하고 응답시간이 느린 단점이 있다.

XrML(eXtensible rights Markup Language)은 디지

털 자원 활용에 대한 권리와 조건을 표현하기 위한 XML 기반언어이다[9]. XrML은 직접적인 방식으로 다양한 작업 흐름 단계에서 단순한 권리와 복잡한 권리를 모두 표현할 수 있다는 점에서 포괄적이다. 또한 여러 다양한 분야로 확장되고 적용될 수 있다는 점에서 보편적이다. XrML은 본 논문의 XML 문서 접근을 위한 보안 요구사항을 고려하고자 할 때, 다음과 같은 문제가 있다.

첫째, XrML은 주체의 그룹 구성원 자격 같은 매우 단순한 특성을 투명하게 표현하는 데 어려움이 있으며, 타깃 문서의 요소가 요구 제기자의 특정 속성과 부합되어야 한다는 제약이 있다.

둘째, XrML은 정책 작성자로 하여금 디지털 서명을 통한 인가서 발행자 인증 같은 많은 세부 사항들을 다루게 한다. 의무는 지원되지만, 그 규약에 수반된 복잡성의 부분은 XrML 내에서 직접적으로 다루어야하는 문제가 있다.

셋째, XrML은 전자 서적, 오디오, 또는 비디오 파일 같은 정적 자원을 다루는데 더 적합한 것 같은 반면, 수정 가능한 XML 문서 같은 동적 자원을 다루는 데에는 많은 어려움이 있다.

Gabillon에서는 권한 부여의 의미론을 다양한 것으로 정의하였다[10]. 그러나 이 모델은 모든 종류의 노드를 보호할 수 있는 가능성을 제공하지 않는다. XML 문서의 접근 제어는 읽기 연산만을 제공하고, 접근 제어 시스템에서 권한의 평가 과정이 복잡하고 응답시간이 느린 단점이 있다.

III. XML 문서 보안 명세화 메커니즘

본 논문에서 제안한 접근 제어 알고리즘을 기술하기에 앞서 XML 접근 제어의 정책에 대해서 기술한다.

1. 접근 권한 관리 정책

전자상거래와 같은 특정 분야의 경우 타인에게 공개하지 않아야 하는 개인 정보 등에 대한 보안이 필요하며, 이에 따라 문서에 대한 접근 관리가 요구된다. 따라서 문서를 생성할 때 문서의 내용에 대해서 사용자들에게 접근 권한을 부여하고 접근 권한을 소유하는 사용자만이 문서의 특정 내용에 접근 가능하도록 하기 위한 접근 권한 관리가 필요하다. 즉, 기존의 데이터베이스에서 사용자에게 접근 권한을 부여하는 것과 마찬가지로 XML 문서를 대상으로 한 검색에 있어서도 사용자가 XML 문서의 특정 항목에만 접근할 수 있도록 권한을 부여함으로써 사용자의 권한 범위에 해당하는 데이터만을 제공할 수 있도록 하는 기능이 필요하다. 이를 위해서는 사용자의 접근 권한을 관리할 수 있어야 하며, 사용자가 XML 문서에 접근할 때 권한에 따라 제어할 수 있어야 한다. 쉬운 방법은 권한에 따라 사용자별로 별도의 문서를 보유하는 것이나, 이는 기억 장소의 낭비와 문서의 중복에 따른 문서 변경이 어려운 문제점이 있으므로 하나의 문서로부터 사용자의 권한에 따라 권한이 없는 부분은 제거하고 접근이 허용된 부분만을 전달하는 것이 필요하다.

보안을 필요로 하는 정보에 대한 접근 제어 정책은 권한의 소유자에 의해 객체에 대한 접근 권한이 정해지는 임의적 접근 제어(DAC: Discretionary Access Control), 정보의 보안 수준과 사용자의 보안 등급에 따라 접근을 제어하는 강제적 접근 제어(MAC: Mandatory Access Control), 그리고 역할과 해당 역할의 권한을 정의하는 역할 기반 접근 제어(RBAC: Role-Based Access Control)로 나눌 수 있다[12].

XML 문서의 권한 타입은 크게 두 가지로 XML 문서에 대한 권한 타입과 XML 문서내의 요소에 대한 권한 타입으로 정의한다. XML 문서에서 최상위 요소 하나만 가질 수 있지만 하위 요소는 여러 개의 계층 구조를 가질 수 있다. XML 문서에 대한 권한 타입은 [그림 1]과 같이 정의하여 사용한다.

① Define Read(DR): XML 문서의 정의 부분을 읽기
② Instance Read(IR): 인스턴스 문서를 읽기
③ Instance Generate(IG): 인스턴스 문서를 생성
④ Instance Write(IW): 인스턴스 문서를 읽기, 생성, 수정
⑤ Element Read(ER): 요소의 데이터를 읽기 가능
⑥ Element Write(EW): 요소의 데이터를 읽기, 수정

그림 1. XML 문서와 요소에 대한 권한

인스턴스 문서에 대한 권한 타입은 상호 결합된 형태로 존재하고 이를 계층구조로 표현할 수 있다. [그림 2]는 본 논문에서 정의한 인스턴스 문서의 권한 타입을 계층 구조로 나타낸 것이다. [그림 2]에서 IW 권한의 소유자는 IG와 IR 권한도 묵시적으로 소유하는 것을 보여주고 있다. 또한 문서 내의 요소에 대한 권한 타입도 인스턴스 권한 타입과 상호 결합된 형태로 존재하고 이를 계층 구조로 표현할 수 있다.

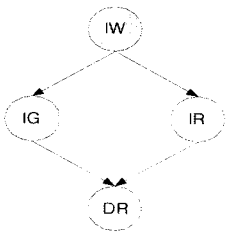

그림 2. 문서에 대한 권한 타입

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 권한 부여 사이에 충돌이 발생한다. 본 연구의 시스템에 의해 실행되는 충돌 해소 정책은 [그림 3]과 같은 원리를 토대로 권한의 우선순위를 결정하는 규칙들을 제안한다.

전파 규칙은 권한 충돌이 발생할 때 이를 조정하여 접근 권한을 설정하는 보안 정책이다. 권한은 전파와 오버라이딩을 고려하여 각 요소에 타입을 반영한 최종 부호를 결정한다. 동일 주체에 대해 허가와 거부가 모두 존재하는 경우 충돌 해결 원리에 따라 하나의 접근 권한만을 결정한다.

step1: subject 관계사이의 부분적 순서에 의해 기술된 가장 상세히 기술된 주체에 관한 권한에 우선순위가 높다.
step2: 전파되어 발생한 권한보다는 직접 기술된 권한이 우선순위가 높다.
step3: DTD에 기술된 권한보다는 XML 문서에 직접 기술된 권한이 우선순위를 갖게 된다.
step4: 노드상의 권한이 그의 조상의 권한 보다 우선 시 된다.

그림 3. 권한 충돌 우선순위

2. 안전한 접근 제어 알고리즘

각각의 접근 권한부여 상태는 주체가 요소/속성에 접근할 수 있는지 여부를 표시한다. 인스턴스나 스키마 수준에서 해당 객체에 대한 각각의 권한부여와 관련된 타입은 객체의 구조와 관련이 있다. 3.1절에서 논의한 원리에 따라 문서에 대한 권한부여를 시행하기 위해서는 문서의 요소/속성에 허가 권한부여가 적용되는지(+), 거부 권한부여가 적용되는지(-), 아니면 어떠한 권한부여도 적용되지 않는지를 표현해야 한다.

레이블링된 권한정보는 사용자의 질의를 거부할지 혹은 허가할지 결정하는데 사용된다. 연산자 단위로 권한 정보를 DOM 트리에 레이블링 한다면, 질의문에 포함된 연산자의 종류 수만큼 레이블링 과정을 반복하여 처리 한다. 이러한 반복적인 과정을 제거하기 위해 논 논문에서는 문서 트리 알고리즘을 제안한다.

[그림 4]는 문서 트리 알고리즘이다. 전체 접근 제어 정보에서 XML 문서에 해당하는 접근 권한을 A.xml에 하고, DTD에 해당하는 접근 권한을 A.dtd로 분리한다. 보안 관리자가 특정 요소에 명시적으로 권한을 설정하지 않았다면, 미리 정해진 기본 접근 권한이 할당된다. 의뢰자(rq) rq와 XML 문서 URI가 있을 때, 알고리즘은 먼저 문서를 나타내는 트리로 변수 T를 초기화하고 T의 루트에 r을 초기화한다. 따라서 Initial_Label의 첫 번째 단계는 의뢰자 rq에게 적용되고 인스턴스 및 스키마 수준에서 문서 URI에 대해 정의된 권한부여 집합 A의 결정으로 구성되어 있다. Default() 함수는 명시적인 권한 설정이 없을 때 할당되는 권한을 반환한다. Propagation_rule() 함수는 동일한 모드에서 충돌이 발생한 경우 미리 정해진 전파규칙에 의해 우선순위가 가장 높은 권한을 반환한다. 이러한 과정은 전순위 순서로 트리를 방문한다. L1r과 L2r의 합집합 값이 공집합이라면, 즉 명시적 접근 권한 정보가 없다면, Lr에 미리 정해진 기본 접근 권한 값을 설정한다. 그렇지 않다면 명시적 권한들 중 우선순위가 가장 높은 권한을 설정한다. 루트 노드의 경우는 부모 노드가 없지만 루트 노드를 제외한 모든 노드는 부모 노드가 존재한다.

```
Input: A requester rq and an XML document URI
Output: The view of the requester rq on the document URI
/* L1r = action(r) in A.dtd */
/* L2r = action(r) in A.xml */
/* A.xml A = {a= <subject, object, action, altg, sign, type>
| a ∈ authorization} */
/* Type() : returns the type specified in the authorization
rule */
void main()
/* r is the root node of URI document,
   n is a nodes other than r,
   p is the parent node of n */
{ initial_label(r);
   label(n, p);
   prune();    }
void initial_label(r)
{ if L1r ∪ L2r = Ø   then Lr=default(r);
   else Lr = propagation_rule(L1r, L2r);  }
void label(n, p)
/* L is local, R is recursive, LDH is Local DTD Hard, RDH
is Recursive DTD Hard */
{ if type (p) in [L, R, LDH, RDH] then
        if L1n & L2n = Ø   then Ln = Lp;
        else Ln = propagation_rule(Lp, L1n, L2n);
   else if L1n & L2n = Ø   then Ln = default(n);
          else Ln = propagation_rule(L1n, L2n);   }
void prune()
/* Tree representing the document, Determines if n has to
be removed from r */
{ For each c ∈ children(n) do prune()
   if children(n) == Ø and Ln ≠ '+' then
      remove the current node from r;    }
```

그림 4. 문서 트리 알고리즘

레이블링의 결과로 각 노드 n은 접근이 가능(+)하거나 접근이 불가능(-)하다. 각 노드의 값 ε 권한이 기술되지 않은 것으로 닫힌 정책과 열린 정책 시행에 따라 각각 거부와 허용으로 해석할 수 있다. 문서의 구조를 유지하기 위해 요청자에게 보여지는 문서의 부분은 접근 가능한 부분의 자식노드이고 접근이 불가능하거나 권한이 기술되지 않았더라도 시작 태그와 끝 태그는 포함한다. 원래의 문서의 트리는 요청자에게 권한이 없는 부분도 모두 포함하고 있으므로 각 노드의 부호에 따라 트리의 제거 과정을 거쳐 요청자에게 허용 가능한 부분만을 보여주는 뷰를 생성하게 된다.

IV. 효율적인 XML 문서 관리 시스템 구조

본 논문에서 시스템 도구로는 CPU Pentium IV 3.4GHz, 하드디스크 300GB, 메모리 1024MB, Windows XP 운영체제, Apache Tomcat 6.0, DOM, 인터넷 익스플로러 8.0, Java 5.0을 이용하였다.

제안하는 XML 문서 관리 시스템의 구조는 [그림 5]와 같다. 사용자는 원격 사이트에서 HTTP 요청이나 질의로 XML 문서를 요청할 수 있다. 이 프로세서는 사용자에 의해 요청한 유효한 XML 문서와 인스턴스 레벨에서 권한의 기술된 접근 제어 리스트를 입력으로 한다. 프로세서의 연산에서 또한 문서의 DTD와 스키마 레벨에 기술된 접근 제어 리스트도 포함한다. 이 프로세서의 출력은 사용자에게 접근이 허용된 정보만을 포함하는 유효한 문서이다. 이 시스템에서 문서와 DTD는 DOM 기술에 따라 내부적으로 표현된다.

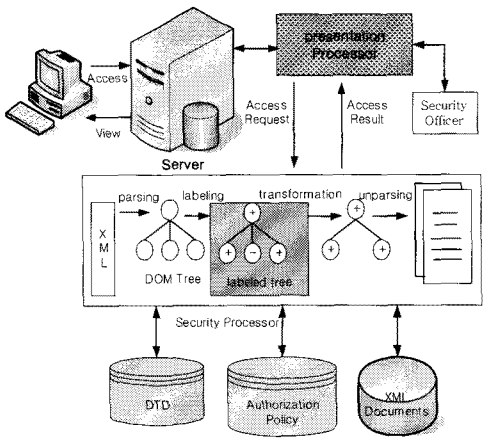

그림 5. XML 문서 관리 시스템 구조

접근 제어는 인증을 거친 사용자가 원하는 자원을 요청하게 되면 시스템은 사용자의 권한을 확인하고 요청한 자료를 제공할 것이지 아닌지를 결정하게 된다. 문서의 접근 제어 역시 인증과정을 거친 사용자가 원하는 XML 문서를 시스템에 요청하게 된다. 보안 프로세서는 요청한 사용자의 권한을 권한 데이터베이스로부터

불러와 체크하게 된다. 만약 사용자의 권한이 요청한 문서의 전체 혹은 일부에 권한이 주어졌다면 XML 문서의 변환 과정을 거친 후 사용자에게 알맞은 뷰 형태로 전달되게 된다.

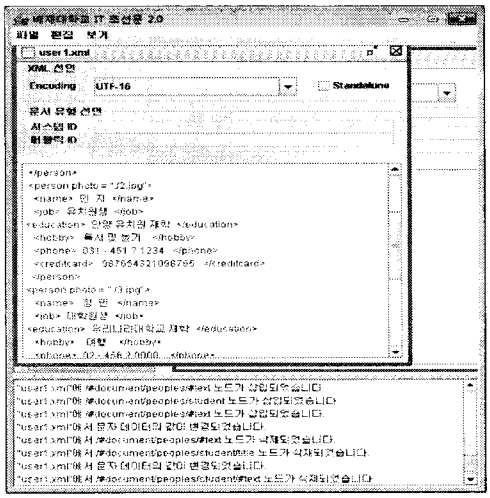

그림 6. 문서 처리 예

XML 문서에 대한 접근 제어 시스템이 최근 제시되었다. 그러한 시스템은 객체 지향 데이터베이스에 대한 이전 시스템과 매우 유사하며, 실제로 XML의 몇몇 특성을 고려하지 않는다. 특히 이 시스템은 두 가지 큰 단점이 있다. 첫 번째는 DTD에 부합되지 않거나 부분적으로 부합되는 경우를 고려하지 않는 다는 점이다. 따라서 기존 모델은 그러한 문서를 다루는 보안 관리자는 지원하지 않는다. 이와 반대로, 본 연구의 시스템은 매우 많은 옵션을 제공하여 이러한 문서를 다룰 수 있게 하였다. 두 번째는 기존의 제시된 기법은 XML 문서 고유의 접근 제어 모드를 제공하지 않는다. 단지 읽기 접근 모드만 제공할 뿐이다. 이와 반대로, 본 논문은 브라우징과 저자를 위한 많은 접근 모드를 제공하는데, 이를 통해 보안 관리자는 사용자가 요소에 있는 정보를 읽거나, 링크를 따라 검색하거나, 요소 링크를 추가/수정, 또는 삭제할 권한을 부여할 수 있다.

[그림 6]과 같이 본 논문에서 XML 접근 제어 처리에 관한 그림이다. 문서의 모든 객체에 대한 권한부여를 정의하고, 사용자가 XML 구조의 다양한 노드에서 다양한 방식으로 접근하게 할 수 있다. 자신에게 정의된 권한 부여에 따라 HTTP를 통해 먼 곳으로부터 XML 자원을 입수할 수 있다.

성능 평가는 제안한 접근 기법과 기존 접근 기법[5]에 대하여 접근 제어 시간을 비교하였다. XML 문서와 접근 시간을 측정하기 위한 매개변수는 벤치마크로부터 XML 데이터와 문서를 이용하였다[1][11].

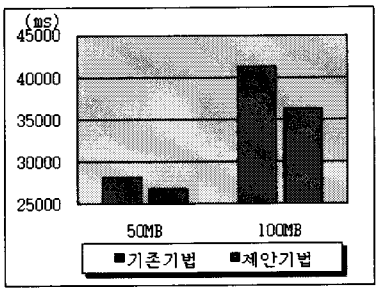

그림 7. 접근 제어 시간 비교

XML 문서는 50MB와 100MB를 평가하고 매개 변수의 시간 단위는 밀리세컨드이다. [그림 7]과 같이 접근 제어 시간을 제안한 기법과 기존 기법과 비교한 결과이다. 본 논문에서 제안하는 검색 시간을 나열하면 접근 권한부여 정책 파싱, 권한부여 정책 검색, 문서 파싱, 접근 권한부여 정책 검색, XML 문서 검색, 접근 권한부여 시간이 소요된다.

V. 결론

웹 환경에서의 정보는 공용의 네트워크상에서 분산되고 공유된다. 그렇기 때문에 민감한 정보에 대한 권한이 없는 사용자의 접근, 침입, 정보 위조와 같은 공격에서 안전하지 못하다. 정보에 대한 접근 제어는 대부분이 HTML문서에 대한 것이다. 이는 파일 단위의 접근 제어로써 XML의 주된 장점인 부분적 정보의 의미에 따른 접근 제어를 다룰 수 없었다. 이에 XML의 장점을 잘 적용한 접근 제어 방법 연구가 필요했다.

본 논문에서 권한부여 정책은 일반적인 사용자나 객체에 대한 권한을 부여할 때뿐만이 아니라 동일한 객체

에 대한 여러 주체들의 권한 충돌이 발생할 때, 또는 동일한 주체에 대한 객체에 관해서도 여러 권한들이 중첩될 때에도 활용된다. 권한부여를 정의함으로써 사용자와 접근 권한 정보의 관리가 용이해졌다. 또한 DOM 트리의 반복적인 방문으로 발생했던 시스템의 성능저하를 개선하였다.

향후 연구로는 XML 구조의 특성을 고려하는 보안 서비스 제공 모델로 전자서명, 암호화 등과 확장하여 통합된 프레임웍을 제시하는 연구가 필요하다.

참 고 문 헌

[1] 조선문, 정경용, "웹 서비스를 위한 데이터 접근 제어의 정책 시스템", 한국콘텐츠학회논문지, 제8권, 제11호, pp.25-32, 2008.

[2] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon, "XML Signature Syntax and Processing," http://www.w3.org/TR/xmldsi g-core/, 2002.

[3] T. Imamura, B. Dillaway, and E. Simon, "XML Encryption Syntax and Processing," http://www.w3. org /TR/xmlenc-core/, 2002.

[4] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon, "XML Key Management Specification(XKMS 2.0)," http://www.w3. org/TR/xmldsig-core/, 2002.

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

[6] M. Kudo and S. Hada, "XML Document Security based on Provisional Authorization," Athens, Greece, 2000.

[7] P. Samarati, E. Bertino, and S. Jajodia, "An authorization model for a distributed hypertext system," IEEE Trans. Knowl. Data Eng. 8, pp.555-562, 1996.

[8] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, "A Query Language for XML," In International Conference on World Wide Web, http://www8.org/, 1999.

[9] Content Guard. "eXtenible Rights Markup Language (XrML) 2.0," Available at http://www.xrml.org, 2001.

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

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

[12] R. 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.

[13] N. Qi, M. Kudo, J. Myllymaki, and H. Pirahesh. "A Function-Based Access Control Model for Xml Databases," In proc. CIKM, ACM, pp.115-122, 2005.

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

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 보안, 임베디드 시스템, 지능시스템, 프로그래밍 언어, 데이터마이닝

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> : 지능시스템, 데이터마이닝, 웨어러블 컴퓨팅, HCI, 상황인식, XML