XML 문서의 유효성 문제 해결에 관한 연구

홍성표* · 송기범* · 방극인* · 이 준**

*조선대학교 대학원 컴퓨터공학과 · **조선대학교 전자정보공과대학 컴퓨터공학부

A Study on Resolution of Validity in XML Document

Seong-pyo Hong* · Gi-beom Song* · Keug-in Bang* · Joon Lee**

*Dept. of Computer Engineering, Graduate School, Chosun University

**School of Computer Engineering, Chosun University

e-mail: hony1128@hanmail.net

요 약

XML은 문서의 데이터 포맷 표현을 향상시키는데 중점을 두고 만들어졌기 때문에 문서 변조 및 데이터 삭제 등의 공격에 취약한 문제점을 가지고 있다. 이에 대한 해결책으로 XML 전자 서명, XML 암호화 기법, XML 접근 제어와 같은 다양한 해결책이 제시되었지만 XML 암호화로 인한 구조적인 XML 유효성 위반 문제 및 DTD 공격에 대한 해결책 부재 등의 문제점이 해결되지 않고 있다.

본 연구에서는 유효성과 암호화를 동시에 만족시켜줄 수 있는 방법으로 XML 스키마를 이용하는 방법을 제안하였다. XML 스키마는 정형 XML 문서에 대해서도 지원이 가능하며, XML 문서에 대한 메타 정보를 담고 있으므로 따로 DTD를 필요로 하지 않는다. 또한 각각의 XML 문서에 대하여 동적인 생성이 가능하며 자체 유효성 검사 규칙을 가지고 있으므로 DTD 기반 XML 문서의 암호화에 대한 확장성이 뛰어난 장점을 가지고 있다.

ABSTRACT

XML has weakness problems on document modulation and elimination of data Because of the XML gives priority to present data format, XML electrical signature, XML cryptography, or XML access control is provided to overcome those weakness problems. However, structured XML efficiency contravention problem occurred from XML encryption and absence of protection from DTD attack are still remains unsolved.

In this paper, we provide XML scheme that satisfies both efficiency and encryption. DTD is unnecessary because XML scheme supports formatting(Well-Formed XML) XML documents and it also include meta information. Because of the XML scheme has possibility to generate each XML document dynamically and self efficiency investigator rule, it has an advantage on extendability of DTD based encryption of XML documents.

키워드

Security, Digital Signature, XML Schema, XML Security

I. 서 론

XML 문서는 적격문서와 유효한 문서로 나눌 수 있다. 적격 문서란 XML 문서에 사용된 모든 요소들이 시작 태그와 끝 태그를 가지고 있고, 중첩규칙을 위반하지 않는 문서를 의미하는 것으로 스키마가 없는 XML 문서를 비 검증용 파서로 파싱했을 때 오류가 없는 문서를 말한다. 따라서, 적격문서는 XML 사양에 명시되어 있는 XML 문서로서 간주되는데 필요한 최소한의 필수 조건을 만족해야 한다. 유효한 문서란 XML 문서가 스키마의 정의대로 올바로 작성되었는지 검사하는 유효성 검증 과정을 거치게 되는데, 스키마가 있는 XML 문서를 검증용 파서를 사용하여 파싱했을 때 오류가 없는 문서를 말한다.[1][3][4]

XML 기술의 급속한 성장과 서비스의 확산으로 XML 문서 보안의 중요성이 크게 대두되고 있고, 범용적인 네트워크 보안 기술과 함께 XML 기반

의 전자상거래 보안 또한 중요하게 여겨지고 있다. 현재 XML 보안 기술은 사용자 인증 및 데이터의 기밀성, 사용자 키관리, 접근제어 부분에 대해서 논의되고 있으며, 다른 정보보호 분야에 대한 연구도 계속 진행되고 있다. 그림 1은 Infrastructure 부분과 Application 부분으로 나뉜 계층별 XML 보안 기술을 나타낸다.[2][4][7]

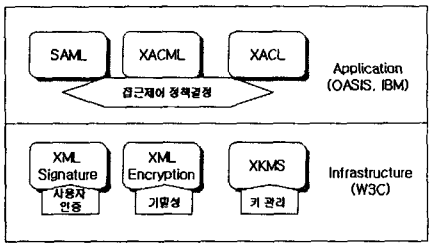

그림 1. 계층별 XML 보안 기술

XML Encryption에서 XML 문서를 효과적으로 암호화하기 위해서는 기본적으로 XML 문서의 유효성을 유지시켜 주어야 한다. 그러나, 어떤 XML 문서도 암호화 후에는 정형 XML 문서가 된다. 왜냐하면 암호화에 관련된 태그를 암호화 이전에 작성된 XML 문서에 기반이 되는 DTD에서 지원해주지 못하기 때문이다. 즉, XML 문서의 유효성을 유지시키면서 암호화를 수행하기 위해서는 DTD를 기반으로 한 XML 문서의 경우 DTD 자체가 새롭게 정의 되어야하는 해결되기 어려운 단점을 갖는다. 일부 태그가 암호화로 대체되면 전체 DTD가 새롭게 바뀌어야만 유효한 XML 문서로 검증될 수 있다. 뿐만 아니라, DTD는 확장성이 떨어지며, 데이터로서 XML을 제대로 기술하기 어렵고 네임스페이스를 제대로 지원하지 못하는 등 여러 가지 제한 사항을 가지고 있어 DTD에 기반한 XML 문서의 암호화가 유효성을 유지하는 데 많은 어려움이 따른다.

본 논문에서는 유효성과 암호화를 동시에 만족시켜줄 수 있는 방법으로 XML 스키마를 이용하는 방법을 제안한다. XML 스키마는 정형 XML 문서에 대해서도 지원이 가능하며, XML 문서에 대한 메타 정보를 담고 있으므로 따로 DTD를 필요로 하지 않는다.

II. XML DTD의 단점

XML 1.0에서는 XML 문서의 타입을 정의하기 위한 메커니즘(DTD)을 제공하지만 XML 개발자들은 DTD를 구현한 경험에 의해 DTD의 수많은 단점과 내용 모델을 정의하는 더욱 포괄적이고 엄격한 방법이 필요하다는 것을 알게 되었다. DTD의 대표적인 단점은 다음과 같다.[5][6][8]

① DTD를 기술하는 문법은 XML을 기술하는 문법과 다르다. DTD의 경우 EBNF(Extended Backus-Naur Form) 표기법을 사용함으로써 XML 문서와 달리 사람이 읽는데 어려움이 많다. 또한 개발자가 XML 문법과 DTD 문법 둘다 알아야하는 어려움과 DTD를 작성하는 S/W와 XML 문서를 작성하는 S/W를 분리시켜야할 뿐 아니라 DTD문서와 XML 문서를 구문 분석하는 파서 역시 분리시켜야 한다.

② 확장성을 갖추지 못한다. DTD를 갱신했을 때, 이전의 DTD로 검증된 모든 문서들을 갱신된 DTD로 다시 검증해야만 한다. DTD를 변경하지 않고서는 XML 문서를 확장시킬 수 없다.

③ XML 문서의 데이터 타입이 문자 타입인, #PCDATA(Parsed Character DATA)와 CDATA(Character DATA) 둘 뿐이므로 제대로 표현하지 못한다.

④ DTD는 상속과 하위 분류에 제한이 있어 객체지향 디자인에 부응하지 못한다.

⑤ DTD는 이름 공간을 지원하지 못한다. 컴퓨팅 환경이 XML 중심으로 변모하고 XML 정보간의 상호연동이 확산되면 될수록 이름 공간의 필요성이 높아지고 있는 반면에 DTD는 효용가치가 떨어지기 때문이다.

⑥ DTD 만으로는 기술 능력의 한계 때문에 강력한 설명을 지원할 수 없다.

⑦ 기본적인 요소 내용에 대한 기능이 없다.

⑧ DTD는 주어진 요소나 속성이 어떤 종류의 정보를 포함하고 있는가를 제어할 수 없다.

⑨ 제대로 된 DTD 작성이 어렵다.

⑩ DTD에 있는 모든 선언들은 포괄성을 가져 다른 문맥에 있다하더라도 같은 이름을 사용할 수 없다.

III. XML 문서의 유효성 검증

3.1 XML 스키마

XML은 컨텐츠와 표현을 분리시켰지만, DTD의 많은 문제점으로 인해 확장성이 떨어지는 단점을 안고 있다. 이러한 문제점을 해결하기 위한 방안으로 XML 스키마가 제안되었다.[2][5][7]

표 3은 임의의 문서에 대한 XML 스키마의 예이다. XML 스키마의 작성은 XML 스키마가 정의된 네임스페이스를 먼저 참조 한 후 문서를 처음부터 끝까지 읽어들인다. 그리고, 가장 하부에 있는 엘리먼트 및 속성에 대하여 데이터와 엘리먼트 등에 대한 정보를 이용하여 스키마를 작성한 후 루트 엘리먼트에 도달할 때까지 이 과정을 역으로 순환하게 된다. 표 1에서는 <name>과 <GPA> 태그가 가장 하부에 있으므로 이에 대한 엘리먼트 스키마를 작성한 후 이 태그를 포함하는 <student> 태그에 대한 스키마를 작성하고 있음을 알 수 있다.

표 1. XML 문서와 그에 대한 XML 스키마의 예

```
<?xml version="1.0"?>
<class xmlns="x-schema:schema_ex.xml">
<student studentID="10141102">
<name> Seong Pyo Hong <name>
GPA>4.5</GPA>
</student>
</class>
```

```
<?xml version="1.0"?>
<schema xmlns="urn:schemas-microsoft-co
m:xml-data" xmlns:dt="urn:schemas-micros
oft-com:datatypes">
<AttributeType name='studentID' dt:type='
string' required='yes'/>
<ElementType name='name' content='textO
nly'/>
<ElementType name='GPA' content=textOn
ly' dt:type='float'/>
<ElementType name='stuent' content='mix
ed'>
  <attribute type='stuedntID'/>
    <element type='name'/>
    <element type='GPA'/>
</ElementType>
<ElementType name='class' cotent='eltOnly'>
    <element type='student'/>
</ElementType>
<Schema>
```

3.2 유효성 검증

효과적인 XML 문서의 암호화를 위해서는 기본적으로 XML 문서의 유효성을 유지시켜 주어야 한다. 그러나, 어떤 XML 문서도 암호화 후에는 정형 XML 문서가 된다. 왜냐하면 암호화에 관련된 태그를 암호화 이전에 작성된 XML 문서에 기반이 되는 DTD에서 지원해주지 못하기 때문이다. 즉, XML 문서의 유효성을 유지시키면서 암호화를 수행하기 위해서는 DTD를 기반으로 한 XML 문서의 경우 DTD 자체가 새롭게 정의 되어야하는 해결되기 어려운 단점을 갖는다. 일부 태그가 암호화로 대체되면 전체 DTD가 새롭게 바뀌어야만 유효한 XML 문서로 검증될 수 있다. 뿐만 아니라, DTD는 첫째, 확장성이 떨어지며 둘째, 데이터로서 XML을 제대로 기술하기 어렵고 셋째, 네임스페이스(Name-space)를 제대로 지원하지 못하며 넷째, 기술 능력에 한계가 있는 등 여러 가지 제한 사항을 가지고 있어 DTD에 기반한 XML 문서의 암호화가 유효성을 유지하는데 있어 많은 어려움이 따른다.

따라서 본 논문에서는 유효성과 암호화를 동시에 만족시켜줄 수 있는 방법으로 XML 스키마를 이용하는 방법을 제안한다. XML 스키마는 정형 XML 문서에 대해서도 지원이 가능하며, XML 문서에 대한 메타 정보를 담고 있으므로 따로 DTD를 필요로 하지 않는다. 또한 각각의 XML 문서에 대하여 동적인 생성이 가능하며 자체 유효성 검사 규칙을 가지고 있으므로 DTD 기반 XML 문서의 암호화에 대한 확장성이 뛰어난 장점을 가지고 있다.

암호화된 XML 문서에 XML 스키마를 이용하여 XML 문서의 유효성을 유지하는 과정은 아래와 같다.

① XML 문서를 파싱
② XML 문서내의 데이터를 읽어들여서 암호화를 수행
③ 암호화된 XML 문서를 파싱해서 새로운 XML스키마를 작성
④ XML 스키마와 XML 문서에 전자 서명을 첨부
⑤ XML 스키마의 특정 데이터 암호화도 필요에 따라 수행

작성된 XML 스키마는 XMLSchema2001의 규칙을 따르게 작성되었다. 이 XML 스키마는 암호화된 XML의 유효성을 유지시키는 기능을 위해 만들어진 것이며 따라서, XML 문서가 수신되고 이에 대한 유효성 검증 작업을 마친 후에는 필요가 없다.

유효성 검사 결과 XML 문서가 암호화된 이후에도 지속적으로 유효성을 유지하고 있음을 확인할 수 있었다. 이 결과를 통해서 문서의 브라우징이 가능하며, 네트워크 상의 송수신 후에도 문서에 대해 무결성의 수준이 높아져 사용자와 제공자에게 동시에 신뢰성을 증대시킬 수 있다.

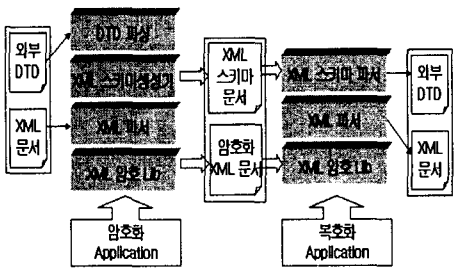

그림 2. XML 스키마를 이용한 유효성 검증 과정

IV. 결 론

XML은 전자상거래에 관련된 데이터 교환이 인터넷 상에서 쉽고 원활하게 이루어질 수 있도록 하는 어플리케이션에 적합한 언어로 평가받고 있다. 그러나, XML은 문서의 데이터 포맷 표현을 향상시키는데 중점을 두고 만들어졌기 때문에 문서 변조 및 데이터 삭제 등의 공격에 취약한 문제점을 가지고 있다. 이러한 문제점에 대한 해결책으로 XML 전자 서명, XML 암호화 기법, XML 접근 제어와 같은 다양한 해결책이 제시되었지만 XML 암호화로 인한 구조적인 XML 유효성 위반 문제 및 DTD 공격에 대한 해결책 부재 등의 문제점이 해결되지 않고 있다.

본 연구에서는 유효성과 암호화를 동시에 만족시켜줄 수 있는 방법으로 XML 스키마를 이용하는 방법을 제안하였다. XML 스키마는 정형 XML 문서에 대해서도 지원이 가능하며, XML 문서에 대한 메타 정보를 담고 있으므로 따로 DTD를 필요로 하지 않는다. 또한 각각의 XML 문서에 대하여 동적인 생성이 가능하며 자체 유효성 검사 규칙을 가지고 있으므로 DTD 기반 XML 문서의 암호화에 대한 확장성이 뛰어난 장점을 가진다.

향후 연구과제로는 문서를 분석할 때마다 문서의 유효성 확인으로 인한 속도저하 문제를 극복할 수 있는 방안에 대한 것이다.

참고문헌

[1] E. Bertino, M, Braun , S. Castano, E. Ferrari, M. Mesiti, "*Aurhor - x: a Java - Bas ed System f or XML Data Protection* ", Proceeding of th e 14th IFIP WG 11.3 Working Conference on Database Security , Schoorl. Netherlands , August . 2000.

[2] H. Maruyama, K.Tamur a, N. Uramoto, "*XML and Java, Developing Web Applications* ", Addison Wesley , May , 1999

[3] William J .Pardi, "*XML in Action, Web Technology* ", Microsoft Press , 1999.

[4] Jonathan Knudsen , "*Java Cryptography* ", O'REILLY, 1998.

[5] ST I- SECURITY Technologies Inc, "*J/LOCK - Java Cryptography Package*", March , 2000.

[6] Takeshi Imamura, Hiroshi Maruyama, "*Specification of Element - wis e XML Encryption* ", W3C XML-Encryption Workshop, November , 2000.

[7] Michiharu Kudo, Satoshi Hada, "*XML Document Security based on Provisional Authorization*", Conference on Computer an d Communication Society , Athens . Greece, Nov ember . 2000.

[8] E. Damiani, S Vimercati, S. Paraboschi, P. Samarati, "*Design and Implementation of an Access Control Process or for XML Documents* ", Proceedings of 9th International World Wide Web Conference, Amsterdam, May , 2000.