인터넷 환경에서 안전한 XML 문서에 관한 연구

홍성표*, 박영옥*, 조애리*, 유혁선*, 이 준**
* 조선대학교 대학원 컴퓨터공학과
** 조선대학교 전자정보공과대학 컴퓨터공학부

A Study on Secure XML Documents in Internet Environments

Seong-Pyo Hong*, Young-Ok Park*, Ai-Ri Cho*, Hyuk-Seon Yu*, Joon Lee**
* Dept. of Computer Engineering, Graduate School, Chosun University
** School of Computer Engineering, Chosun University

요 약

XML은 SGML(Standard Generalized Markup Language)의 간략화된 버전으로 SGML의 확장성, 구조, 검증의 특성을 계승하고 있다. 이런 장점으로 XML은 발표된 이래로 인터넷 상의 자료 표현의 표준으로 각광받고 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 제시되었다.

본 논문에서는 XML 문서에만 전자서명을 첨부하는 것이 아니라, DTD에 전자 서명을 첨부하는 방법을 제안하였다. 먼저 DTD파일을 끝까지 읽으면서 파싱을 하고 여기서 추출되는 엘리먼트나 속성, 엔티티들을 해쉬테이블에 저장한다. 파싱이 종료되면 해쉬 테이블을 읽어 들여서 메시지 다이제스트를 수행한다. 수행 후 이를 개인 키와 합성하여 전자 서명을 생성한다.

1. 서론

비즈니스 세계에서의 보안은 비즈니스 처리의 안전을 보장하고 사생활과 기밀성을 유지하기 위해 사용되고 있으며 비즈니스 보안은 비즈니스의 생명에 비견될 만큼 중요하게 인식되고 있다. 오늘날의 인터넷 웹 기반의 사업환경에서 비즈니스 안전을 제공하는 것에 대한 방법은 변화가 필요하다.

XML은 최신 응용분야와 인터넷 컨텐츠를 위해 폭넓게 채택되고 있다. XML은 SGML(Standard Generalized Markup Language)의 간략화된 버전으로 SGML의 확장성, 구조, 검증의 특성을 계승하고 있다. 이런 장점으로 XML은 발표된 이래로 인터넷 상의 자료 표현의 표준으로 각광받고 있다.

그러나 XML은 데이터에 대한 의미적 접근 가능성을 지원하는 반면, 중요 정보들에 대한 표현이 구조적으로 드러나게 되고 따라서 XML 문서상에 나타나는 많은 정보들은 해킹에 무방비 상태로 노출될 수 있다. XML 문서 보안에 대한 근본 문제는 XML 문서가 XML 문서 원본과 문서 내에서 표현하는 여러 정보를 포함하는 DTD의 쌍으로 구성되는 구조 자체에 있다.[1][3]

DTD는 XML을 표현하기 위한 메타 컨텐츠를 가지고 있는 파일로서, 문서내의 데이터에 대한 의미의 구별, 문서의 유효성 검증 목적으로 사용되기 때문에 DTD에 대해서도 XML 자체의 보안에 상응하는 보안 정책이 요구된다.

본 논문에서는 DTD 공격에 대한 해결책으로 XML 문서에만 전자서명을 첨부하는 것이 아니라, DTD 에도 전자 서명을 첨부함으로써 DTD 문서에 대하여 신뢰성을 부여하는 방법을 제안하였다.

2. XML 정보보호 기술

XML 기술의 급속한 성장과 서비스의 확산으로 XML 문서 보안의 중요성이 크게 대두되고 있고, 범용적인 네트워크 보안 기술과 함께 XML 기반의 전자상거래 보안 또한 중요하게 여겨지고 있다.

다음은 주요 XML 정보보호 기술들이다.[2][5-6]

◎ XML Digital Signature

- 인증과 무결성, 전자서명, 부인봉쇄
- XML Encryption
  - 문서의 기밀성
- XML Key Management Specification(XKMS)
  - 효율적인 키 관리
- Security Assertion Markup Language(SAML)
  - 인터넷에서 인증과 인가정보 교환
- XML Access Control Markup Language(XACML)
  - 인가 규칙

2.1 XML 디지털 서명

디지털 서명은 전자화된 문서의 메시지 내용이 수정 및 변조되지 않았음을 보장하는 동시에 메시지의 주체인 사용자들의 신원을 제 3자에게 확인할 수 있게끔 하는 인증방식을 말한다. W3C에서 표준으로 권고된 XML 디지털 서명 기술은 기존의 디지털 서명을 XML을 이용하여 표현한 것으로, 기존의 디지털 서명과 같이 전체 문서에 대해서도 서명을 할 수 있고, 또 XML Transform 기술을 이용하여 서명이 필요한 문서의 일부분에 대해서도 서명 할 수 있어, 기존 문서의 재사용성을 높일 수 있다.[1][7-8]

2.1.1 서명문서 생성

XML 디지털 서명문서의 생성에는 크게 두 가지 생성 절차가 있는데, 첫 번째는 Reference 생성이고, 두 번째는 서명 생성이다. Reference 생성은 사용자 문서에 여러 가지 Transform을 적용하고, Transform 된 문서에 대해 해쉬 값을 계산한다.

서명 생성은 위에서 생성한 Reference 부분을 포함한 서명정보 부분, 즉 SignedInfo 엘리먼트 영역을 사용자의 개인키를 이용해 서명값을 계산하는 과정이다. 여기서 서명전에 서명할 부분에 대한 무결성을 위해 Canonicalization을 수행한다. 그림 1은 XML 서명문서 생성과정을 나타낸다.

생성된 서명 문서는 enveloped, enveloping, detached 세 가지 형태로 구분된다. 서명될 문서 안에 Signature 엘리먼트가 포함되어 있으면 enveloped 서명이고, 서명될 문서가 Signature 엘리먼트 안에 포함되어 있으면 enveloping 서명이며, Signature 엘리먼트와 서명될 문서가 한 XML 문서 안에 없으면, detached 서명 형태이다.

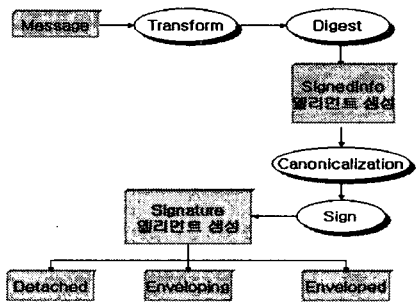


그림 1. XML 서명문서 생성

2.1.2 서명문서 검증

XML 디지털 서명 문서의 검증 역시 Reference 검증과 서명 검증 두 부분으로 나뉘어지는데 두 가지 검증 절차는 다음과 같다.

Reference 검증절차는 수신된 서명 문서의 SignInfo 엘리먼트를 추출하여 이 부분에 대한 무결성을 위해 Canonicalization을 수행하고, Reference 생성과정과 동일하게 사용자 문서에 여러 가지 Transform을 적용한 다음 해쉬 값을 계산한다. 여기서, 사용자 문서는 Reference 엘리먼트의 URL에서 얻을 수 있다. 이렇게 검증시 계산한 해쉬 값과 서명문에 포함된 해쉬 값을 비교하여 그 값들이 동일할 경우 Reference 검증은 성공하게 된다.

서명 검증 절차는 사용자의 공개키에 해당하는 정보를 얻는다. 이 공개키로 서명 값을 검증하게 된다. 검증 결과가 유효하다면 수신 받은 디지털 서명문서는 유효하다고 판단한다. 그림 2는 XML 서명문서 검증 과정을 나타낸 것이다.

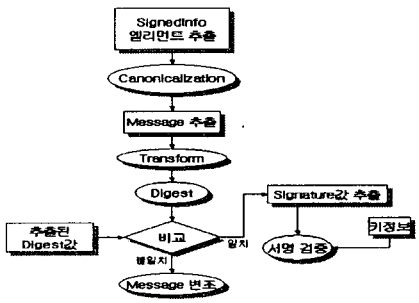


그림 2. XML 서명문서 검증

2.2 XML 암호화 기법

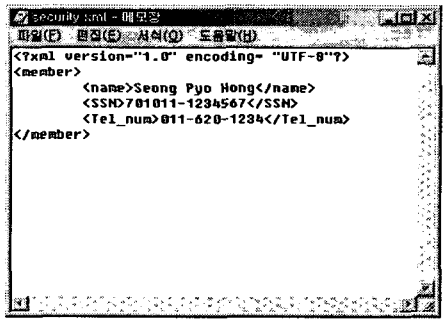
XSS1999[3]와 Imamura2000[4] 및 Brandt2000[5] 등에서 제시한 XML 암호화 기법은 기존의 암호화 기법을 XML에 적용시킨 방법으로, XML 문서 전체의 암호화 기법으로 시작하였으며, 속도 문제의 단점이 지적되어 현재는 엘리먼트 단위로 암호화를 수행하는 향상된 기법이 제안되었다.

Imamura2000은 엘리먼트 암호화를 위한 엘리먼트를 정의하고 있다. 이 정의된 엘리먼트들을 기반으로 하여 XMl 문서에서 암호화가 요구되는 엘리먼트에 대해 암호화를 수행한다. 이 방법은 불필요한 암호화 연산을 하지 않으므로 속도 및 비용면에서 효과적인 반면, DTD 선언(!DOCTYPE...>)이 포함된 XML 문서, 즉 유효한 문서를 처리해주지 못한다. 이는 암호화 과정에서 암호화를 위한 엘리먼트가 문서에 새롭게 추가되지만, 암호화 이전 문서의 DTD에서는 이를 지원하지 않으며, XML 문서는 하나의 DTD를 기반으로 작성되므로 기존 DTD와 새롭게 작성된 DTD를 동시에 인식할 수 없기 때문이다.

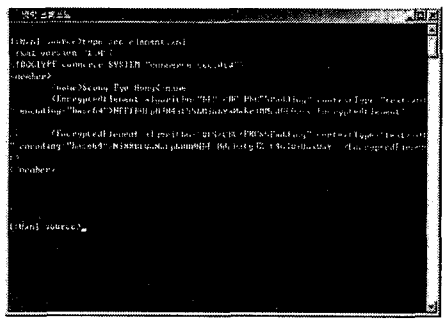
한편, Brandt2000은 여러 계층에 대한 암호화 방식을 제안하였다. 첫째, 암호화하고자 하는 엘리먼트를 secure 엘리먼트로 대체하고, 데이터도 암호화한다. 이 방법은 효과적인 암호화는 가능하지만 모든 암호화 정보가 <secure> 태그로만 설정됨으로써 동일한 태그로 설정된 데이터의 충돌 문제가 발생할 수 있다. 이를 위한 개선 방법으로 태그는 그대로 두고 데이터만 암호화하는 방법을 제안하였다. 둘째, 엘리먼트는 그대로 두고 데이터만 암호화하는 방법이다. 이것은 DTD 내의 엘리먼트 선언에서 불리언 타입의 보안 속성을 설정하여, "참"일 경우 데이터를 암호화하는 방법이다. 이 방법은 데이터만 암호화가 가능한 장점이 있으나, DTD 속성에는 불리언 타입이 정의되어 있지 않으므로 DTD 문법을 위반하는 문제점이 있다.

셋째, XML 스타일 시트를 이용하는 방법이다. 스타일 시트가 XML 문서와 분리되어 있는 특성을 이용한 것으로, 기존 문서를 보존하는 상태에서 스타일 시트 형식의 시큐리티 시트를 이용하여 암호화된 형태로 만드는 방법이다. 이 방법은 XML 문서 및 DTD와 XML 스키마까지도 암호화가 가능하다고 한다. 그러나 현재 검증된 결과가 없고 오히려 암호화를 위해 특정 내용을 분리시키는 컴포넌트(component)없이는 암호화가 불가능하며, 암호화 수행 후 DTD를 새로 작성해야 하지만 이를 지원하지 못하는 단점이 있다.

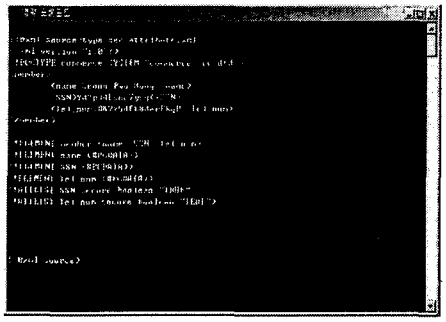
결론적으로, Imamura2000과 Brandt2000 모두 정형(well-formed) XML 문서에 대해서는 암호 기능을 지원하나, XML 문서의 유효성 유지 및 암호 기능을 동시에 만족시키기 위해서는 해결해야할 문제점을 많이 가지고 있다. 그림 3은 암호화에 대한 예를 보여주고 있다. (가)는 암호화 이전의 XML 문서이며, (나)에서는 <SSN>과 <Tel_num> 태그의 데이터가 암호화된 Imamura2000의 예이다. (다)는 Brandt2000에서 제시된 방법에 기반하여 수행된 XML 엘리먼트 암호화를 수행한 예로, DTD내에 정의된 속성에 기반하여 태그의 변경 없이 데이터만 암호화된 결과를 보이고 있다.



(가) 원본 XML 문서



(나) 엘리먼트 단위의 XML 암호화 기법



(다) 엘리먼트 속성에 기반한 XML 데이터 암호화 기법

그림 3. XML 암호화 기법

3. DTD 보안의 문제점

XML 문서는 DTD 또는 XML 스키마에 기반을 두어 작성된다. XML 문서의 데이터를 표현하는 메타 데이터 집합인 DTD는 여러 계층에서 공유되고 있다. 그런데 이러한 DTD의 공유 및 메타 컨텐츠 관리 측면에서 DTD의 보안 기법은 매우 중요함에도 불구하고, 현재의 연구는 XML 문서의 데이터 암호화에 초점이 맞추어져 있다.

DTD에 대한 가장 기본적인 공격은 DTD 파일을 삭제하거나 임의로 파괴하여 XML 문서에 대해 유효성 여부의 검증을 어렵게 하는 것이다. XML 문서는 DTD에 기반을 두어 작성되며 이 규칙을 지킨 문서만이 브라우징이 가능하게 되어있다. 정보 교환 측면에서 볼 때 DTD가 없는 정형 XML 문서는 정상적인 데이터의 의미를 인지하기 어렵기 때문에 애플리케이션 상에서 데이터 처리가 어렵다. 즉 DTD 선언을 포함하고 선언된 DTD 기반에서 작성된 XML 문서는 유효성이 검증되어야 브라우징을 비롯한 데이터 처리가 가능하다.[2][6][8]

그림 4는 정상적인 DTD 선언을 포함한 XML 문서인 경우이고, 그림 5는 공격에 의해 DTD가 삭제되어 정상적인 실행을 하지 못한 경우이다.

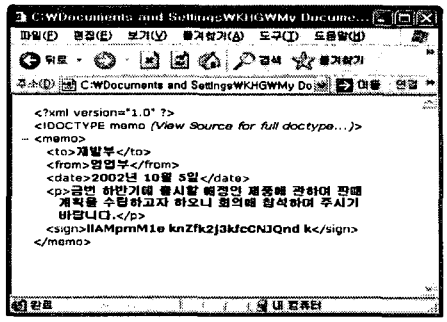


그림 4. 정상적으로 DTD 선언을 포함한 XML 문서

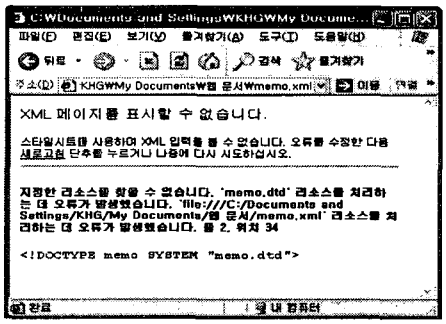


그림 5. 공격에 의해 DTD가 삭제된 XML 문서

4. 설계 및 구현

XML 문서의 데이터를 표현하는 메타 데이터 집합인 DTD는 여러 계층에서 공유되고 있기 때문에 DTD의 공유 및 메타 컨텐츠 관리 측면에서 DTD의 보안이 매우 중요함에도 불구하고, 현재의 연구는 XML 문서의 데이터 암호화에만 초점이 맞추어져 있기 때문에 DTD 파일을 삭제하거나 임의로 파괴하여 XML 문서에 대해 유효성 여부의 검증을 어렵게 하고있다.

본 논문에서는 DTD 공격에 대한 해결책으로 XML 문서에만 전자서명을 첨부하는 것이 아니라, DTD 에도 전자 서명을 첨부함으로써 DTD 문서에 대하여 신뢰성을 부여한다. 애플리케이션에서 XML 문서 처리 전에 서명 값을 검증함으로써 정보 유출 등의 문제를 극복할 수 있다. 문제점은 DTD 내에 존재하는 엘리먼트 선언들의 순서문제이다. 전자 서명 시, 메시지 다이제스트 과정에서 바뀐 순서에 대해서는 검사하지 못하기 때문에 논리적으로 같더라도 전혀 다른 다이제스트 값을 생성하기 때문이다. 이 문제는 XML 전자 서명에서 나타난 것과 동일한 것으로 XML 정규화를 DTD에 적용시키는 정규

DTD 생성 등이 해결책으로 제시될 수 있다. 그러나 이는 DTD 파서가 따로 요구되며 DTD의 정규화를 위해 또 다른 구문법의 정의가 요구되는 등 많은 시간과 노력이 소요된다.

따라서 본 논문에서는 DOM을 이용하여 DTD의 전자 서명을 생성하는 방법을 제안한다. DOM은 구조에 대하여 표준화가 되어 있으며 문서 전체에 대한 트리구조를 구현할 수 있다는 점에서 문서 구조의 정규화에 유리한 장점을 가지고 있다.

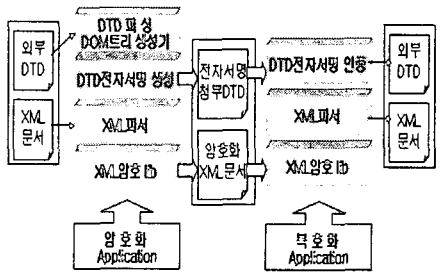


그림 6. DTD 전자서명을 이용한 XML 암호화 과정

DTD 파일에 대한 전자 서명 생성은 먼저 DTD파일을 읽어 들이고 DTD 파일의 끝까지 읽으면서 파싱을 하고 여기서 추출되는 엘리먼트나 속성, 엔티티들을 해쉬테이블에 저장한다. 파싱이 종료되면 해쉬 테이블을 읽어 들여서 메시지 다이제스트를 수행한다. 수행 후 이를 개인 키와 합성하여 전자 서명을 생성한다.

5. 결 론

XML은 인터넷 상에서 데이터 교환이 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 이러한 XML 보안의 취약점을 파악하여 DTD 전자 서명을 이용한 XML 보안 기능을 제안하였다. 기존의 XML 엘리먼트 암호화 기법과 DTD 전자 서명의 관점에 중점을 두었으며 XML 접근 제어 관점에서는 DTD 접근 제어의 적용 가능성을 제시하였다. 따라서 기존의 시스템에서 발생할 수 있는 DTD의 파괴와 같은 문제점을 접근 권한 부여기법을 이용하여 보완함으로써 보다 향상된 보안 기능의 지원이 가능해졌다. DTD 전자서명을 이용한 XML 문서의 암호화를 통해 얻을 수 있는 가장 큰 효과로는 XML이 데이터의 내용과 표현의 분리에만 치중함으로써 발생되었던 보안상의 문제를 극복할 수 있게 되었다는 점이다.

[참고문헌]

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

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

[3] Michiharu Kudo, Satoshi Hada, *"XML Document Security based on Provisional Authorization"*, Conference on Computer and Communication Society, Athens. Greece, November, 2000.

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

[5] 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.

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

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

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