XML 표준전자문서의 Java기반 서명 시스템 설계

이대하° 김순자
경 북 대 학 교 전 자 공 학 과
bigsum@palgong.knu.ac.kr snjkim@ee.knu.ac.kr

Design of signature system based Java for XML/EDI Document

Dae-Ha Lee° Soon-Ja Kim
School of Electronics and Electrical Eng., Kyungpook National University

요 약

확장성 마크업 언어인 XML은 W3C에 의해 고안된 구조화된 표준이다. 이는 SGML의 부분집합으로서 문서의 내용에 관련된 태그를 사용자가 직접 정의할 수 있다. 이런 확장성이 뛰어난 XML을 이용하여 기업간에 사용되는 EDI 문서를 표현함으로써 이기종 시스템과 응용프로그램간의 원활한 문서유통을 이룰 수 있다. 그러나 그런 XML 표준전자문서의 인증이나 신뢰성이 보장되지 않는다면, 전송중 문서의 조작,도청 및 오용등과 같은 위협 때문에 그 사용이 제한될 것이다. 본 논문에서는 그러한 위협을 방지하고 XML 표준전자문서의 신뢰성 있는 전송을 보장하기 위하여 Java 기반 서명 시스템을 설계 하였다. 여기서 서명은 XML 기반으로 구성되었고, 시스템은 Java 애플릿과 애플리케이션을 이용한 두 가지 경우로 설계하였다.

1. 서 론

정보화 사회로 발전해감에 따라 시스템 및 응용에 독립적인 문서정보 처리가 요구되고, 문서 정보 교환,검색 등의 처리를 위한 표준이 필요하게 되었다. 이에 웹문서 양식인 XML이 등장하게 되었는데, 이 XML은 SGML의 부분집합으로서 W3C에 의해 고안되어진 확장성 마크업언어이다. '확장성'이라는 말에서 알 수 있듯이 문서의 내용에 관련된 태그를 사용자가 직접 정의할 수 있으며, 그 태그를 다른 사람들이 사용할 수도 있다. 그런 이유로 XML을 구조적인 자료로 구성된 문서를 위한 마크업 언어라고도 한다. 여기서 구조적인 자료란 자료의 내용이 가지는 의미와 역할이 정해진 규칙에 따라 표현된다는 것을 의미한다.

기존의 EDI(Electronic Data Interchange)는 기업과 기업이 서로 정의된 방식으로 문서를 전달하는 것이었므로 데이터 자체가 정형화되어 있어 확장시에 문제점과 구조적인 데이터 검색 및 처리에 어려움이 있었다. 그리고 이기종 시스템간의 호환성 문제도 남아있었다. 이러한 문제점을 극복하기 위해 웹기반하에서 확장성과 구조적인 특성을 가진 XML을 EDI에 적용하게 되었다[1,2].

그러나 XML화된 EDI 문서가 타인에 의해 쉽게 도청,조작 되거나 오용된다면 이러한 문서에 대한 신뢰성이 떨어져 그 이용이 제한될 것이다. 그러므로 문서에 대한 일단의 보안기술이 필요하게 된 것이다.

이에 본 논문 2장에서는 XML, 디지털 서명에 대한 기본개념과 서명에 대한 XML태그들에 관하여 살펴보고, 3장에서는 Java기반의 서명 시스템에 대해서 논하고, 4장에서는 결론 및 고찰, 향후과제를 제시한다.

2. 기본개념

2.1 XML(eXtensible Markup Language)

XML은 HTML과 SGML이 가지는 단점을 보완하도록 설계되었다. XML의 시초는 모든 플랫폼, 운영체제,환경에서 실행할 수 있는 마크업 언어로서 웹의 내용을 더 효율적으로 표현하도록 고안된 것이지만, SGML을 개량해 모든 데이터를 표현할 수 있도록 하는 효과적인 수단으로 개발된 것이기도 하다. SGML은 다양한 기능을 갖고 있음에도 불구하고 사용이 어렵고 DTD 생성이나 이해가 쉽지 않았다. 그리고 마크업 언어 자체가 아니라 마크업 언어를 생성하기 위한 간접적인 표준이라는 점 등으로 인해 널리 사용되지 못했다. 반면 HTML은 간단하고 사용하기 쉽지만 SGML에 의해 정의된 고정적인 마크업 언어이므로 웹이 지니고 있는 다양함이나 동적인 특성을 쉽게 반영하지 못하고 있다. 이런 문제를 극복한 것이 바로 XML이

다. XML의 기술적인 주요한 특징으로는 HTML을 보완한 것이기에 인터넷 환경을 고려하여 만들어졌다는 것과 XML은 단순히 문서의 내용을 기술하는 표준 뿐 아니라 컨텐트를 포함시키고 있으며, 또한 특정 컨텐트를 표현하는 태그와 속성을 설명하는 DTD(Document Type Definition)를 정의할 수 있다. XML의 핵심은 바로 DTD에 있다고 할 수 있다[3].

2.2 디지털 서명

디지털 서명 생성 절차는 서명하고자 하는 원문(M)을 해쉬함수에 통과시켜 일정한 크기의 메시지 다이제스트를 생성한다. 이 메시지 다이제스트를 서명자의 비밀키로 암호화하면 이것이 원문에 대한 디지털 서명이 된다.

$$HASH(M) = Digest$$
$$E_{sk}(Digest) = S$$

원문과 원문에 대한 디지털 서명(암호화된 메시지 다이제스트)을 함께 수신자에게 전송하면 원문이 서명자에 의해 생성되었음을 증명할 수 있다.

디지털 서명된 문서(M+S)의 수신자는 디지털 서명 확인을 위해 디지털 서명 생성자와 동일한 해쉬함수를 이용하여 원문의 다이제스트(Digest')를 구하고, 서명 즉,암호화된 다이제스트(S)를 서명자의 공개키로 복호화하여 서명자가 생성한 다이제스트(Digest)를 구한다.

$$HASH(M) = Digest'$$
$$D_{pk}(S) = Digest$$

이 두 다이제스트 Digest'와 Digest를 비교하여 일치하면 서명이 확인된 것이다[4].

2.3 디지털 서명을 위한 XML 서명 요소

XML은 데이터나 표현에 대한 언어일 뿐이고 그 자체가 서명을 해주는 것은 아니다. 서명은 외부 애플리케이션이나 애플릿을 통하여 하게 된다. 서명될 문서와 그것의 서명된 값을 XML을 이용하여 표현하는 것이다. 앞에서 살펴본 것처럼 서명에는 몇 가지 함수가 필요한데 이 함수들은 여러 가지 알고리듬으로 구현될 수가 있다. 대표적으로 사용되는 알고리듬으로는 해쉬함수로는 SHA-1이고 공개키 암호화 알고리듬으로는 RSA를 사용하지만 항상 그런 것은 아니다. 어떤 환경이나 필요에 의해서 이 알고리듬은 바뀔 수가 있다. 그리고 이런 알고리듬뿐만 아니라 여러 가지 정보, 즉 서명자정보, 이 서명문을 받을 사람의 정보, 서명날짜, 서명시간, 서명할 문서의 종류 등등이 필요하게 된다. 이를 기존의 HTML로 표현하게 된다면 필요한 정보의 확장성 문제나 정확한 데이터 정보의 검색측면에서 문제점을 가지게 된다.

서명에 사용되는 XML 서명요소는 두 부분으로 나뉘는데, 첫 번째는 서명정보(SignedInfo)부분이고, 나머지 한 부분은 그 서명정보에 대한 서명값(Value)을 나타내는 부분이다[5,6].

표 1에서 XML 서명요소에 대한 태그들을 볼 수 있다.

표 1. 디지털 서명을 위한 XML 서명요소

```
<Signature>
 <SignedInfo>
  (resources information block)   : 서명해야할 문서의 위치 및 그
                                    문서의 종류, 그 문서의 해쉬값
                                    을 나타냄
  (other attributes)              : 서명정보에 관련된 다른 속성
  (originator information block)  : 작성자의 신원확인 정보
  (recipient information block)   : 수신자의 신원확인 정보
  (key agreement algorithm        : 마스터키로부터 세션키를 만드는
   information block)               알고리듬을 나타냄
  (signature algorithm            : 서명값 계산에 사용되는
   information block)               알고리듬을 나타냄
 </SignedInfo>
 <Value encoding='encoding scheme'>
  (encoded signature value)       : 서명된 값을 나타냄
 </Value>
</Signature>
```

3. Java 기반의 서명 시스템

웹 환경에서 Java기반 서명 시스템을 구현하는데 있어 두가지 방법이 있는데, 하나는 애플릿을 이용하는 것이고 나머지는 애플리케이션을 이용하는 것이다.

3.1 Java 애플릿을 이용한 시스템 구성

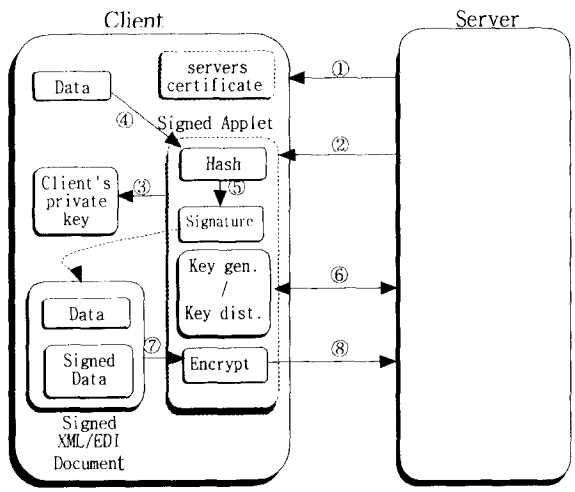

그림 1. 애플릿으로 구현한 서명시스템

① : Server가 인증기관(CA)에서 받은 서명용 전자인증서 파일을 다운로드 받는다.

② : Server로부터 서명 알고리듬이 내장된 애플릿을 다운로드 받는다. 이 때 애플릿은 서명된 애플릿(여기서의 서명은 애플릿에 대한 서명임)으로서 일반 애플릿과는 달리 Client의 파일을 읽거나 또는 Client측에 파일을 쓸 수 있다. 애플릿을 받은다음 인증서와의 확인과정을 거친다음 Client의 영역을 접근할 수 있게 된다[7,8].

③ : 서명된 애플릿은 Client에서 개인키 파일을 읽는다. 여기에서 개인키는 Client가 서명을 하기 위해 만든

것이다. 이 개인키에 대한 공개키는 인증기관을 통해 인증서 형태로 제공된다.

④ : Client가 서명하려고 하는 Data를 Hash한다.

⑤ :Client의 개인키로 Data를 서명한다.(Client의서명)

⑥ : 세션키를 만들고, 키교환 프로토콜을 이용하여 세션키를 분배한다.

⑦ : ⑤번 이후에 그대로 서명화된 XML문서를 전송하게 되면 제 3자가 그 문서를 불법으로 획득한 후 그 문서에 첨부되어 있는 원문의 내용을 읽거나 또는 더 나아가 원문과 서명된 부분을 분리하여 그 원문을 가지고 자신(제3자)의 개인키를 이용해 서명을 한뒤 Server에 보낼 수도 있게 된다. 이를 막기 위해 서명화된 XML문서를 세션키로 암호화한다.(암호화 속도를 고려하여 대칭키 알고리듬을 사용한다)

⑧ : 암호화된 서명문서를 Server로 전송한다.

3.2 애플리케이션을 이용한 시스템 구성

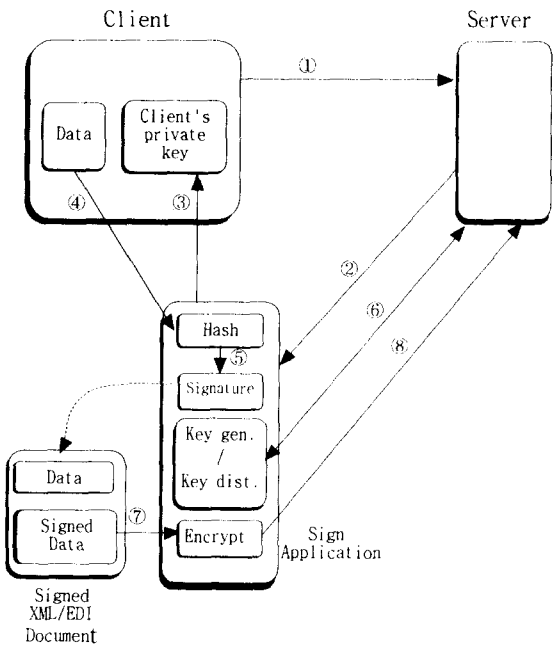

그림 2. 애플리케이션으로 구현한 서명시스템

① : Client가 서명을 하기위해 Server에게 서명요구를 보낸다.

② : Server는 Client측의 서명 애플리케이션을 Wakeup 시킨다.

③ ~⑧ : 애플릿에서와 같다.

3.3 비교

Java 애플릿을 기반으로 시스템을 구성하게 되면 별도의 프로그램을 설치할 필요가 없이 웹 사이트 접속을 통해서 원하는 서비스를 받을 수 있고, 애플릿 자체가 일반 윈도우즈 사용 환경에 맞게 개발되었으므로 일반 사용자로 하여금 새로운 프로그램 사용법을 배워야 한다는 부담을 덜어줌으로써 업무의 효율성을 증대시켜 줄 것이다. 다만 애플릿은 접속할 때마다 다운로드받아야 한다는 점과 그 때문에 발생하는 속도문제로 그 크기에 있어 제한을 받게 된다. 그러므로 다양한 기능을 기대할 수가 없다.

애플리케이션의 경우는 일단 다운로드 받으면 그 애플리케이션을 계속 사용할 수 있고, 더 이상의 다운로드는 필요없다. 그리고 다양한 기능을 구현할 수 있다. 다만 Client측에서 애플리케이션을 구동하려고 할 때마다 Server측에 Wakeup 신호를 요구하게 된다(웹브라우저와 애플리케이션을 플러그인 형태로 사용할 경우).

어떤 방법으로 구현하는가는 필요한 기능과 사용환경에 따라 달라진다. 서명기능뿐만 아니라 다른 기능들을 많이 사용하고자 할 경우는 애플리케이션이 유리할 것이고 그렇지 않고 몇 가지 특수한 기능만 필요로 할 경우는 애플릿 구현도 좋은 방법이 될 것이다.

4. 결론 및 고찰

XML의 확장성과 구조적인 특성은 기존의 EDI 문서를 더 효율적으로 작성할 수 있게 한다. 하지만 이러한 문서가 전송중에 제 3자에 의해 조작,도청되거나 오용된다면 그 문서에 대한 신뢰성이 떨어져 그 사용이 제한될 것이다.

이에 본 논문에서는 전송중에 발생할 수 있는 위협으로부터 XML/EDI 문서를 보호하기 위하여 그 문서를 디지털 서명하고 또 세션키로 암호화하는 Java기반의 서명시스템을 Java 애플릿과 애플리케이션을 이용하여 설계하였다.

향후에는 이 시스템의 구현 및 이 시스템 기능을 확장하여 이중서명 및 SSL을 이용하여 안전한 지불 프로토콜에 대한 연구가 이루어져야 할 것이다.

5. 참고문헌

[1] 방정환, "XML을 이용한 EDI문서 처리 시스템 설계 및 구현" 1999년도 한국정보과학회 가을 학술발표논문집 Vol. 26. No. 2

[2] The XML/EDI Group, "White Paper on Global XML Repositories for XML/EDI", February 1999, http://www.xmledi.com/repository/xml-repWP.htm

[3] 안향준, "XML 개요", http://my.netian.com/~aphise/xml/xmlmain.html

[4] 김춘길, 전자상거래, KRNET'98 6th Computer Networking Conference 특강자료집, pp173-200, 1998.

[5] Richard D. Brown, "Digital Signatures for XML", July 1998, http://www.ietf.org/internet-drafts/draft-ietf-xmldsig-signature-00.txt

[6] IETF W3C, "XML-Signature Core Syntax and Processing, 04-January-2000, http://www.w3.org/TR/xmldsig-core/

[7] "Signed-Applet Example(JDK1.1x)", http://java.sun.com/security/signExample

[8] "애플릿 전자서명", http://cs.chungnam.ac.kr/~dsyoon/java/applet