안전한 XML/EDI 메시징을 위한 XML Signature 설계 및 구현

전형득 (동국대학교 대학원 전자상거래학과, junhd@dongguk.ac.kr)
송유진 (동국대학교 정보산업학과, song@donggguk.ac.kr)
문태수 (동국대학교 정보산업학과, tsmoon@dongguk.ac.kr)

1. 서론

최근 기업에서 인터넷을 활용한 전자문서교환(electronic data interchange)이 급증함에 따라 교환되는 문서의 안전한 전달을 위한 보안 서비스 문제가 대두되고 있다. 기업간 B2B 전자문서 교환은 문서의 신속한 교환과 처리과정의 자동화를 통해 기업 업무 자동화에 큰 기여를 하고 있다. 그러나 현재의 전자문서 교환방식은 해당 소프트웨어 개발과 통신망에 대한 부대 비용으로 인해 광범위하게 채택되지 못하고 있다. 이러한 문제를 해결하기 위한 방안으로 현재 광범위하게 사용되고 있는 웹기반 표준문서인 XML(eXtensible Markup Language)을 이용한 전자문서교환이 새롭게 떠오르고 있다[1].

XML을 통한 데이터 교환은 서로 다른 어플리케이션간의 데이터 교환이 가능하다는 장점 때문에 기존 EDI를 통한 문서교환방식이 갖는 문제점을 효율적으로 해결할 수 있다. XML은 어플리케이션간 인터페이스가 필요한 분야들에서 실용적인 해결책으로 인식되기 시작했고, 서로 다른 시스템과 아플리케이션을 사용하면서 비즈니스 트랜잭션을 교환해야 하는 기업간 B2B나 e-비즈니스 시장의 필수 기술요소로서 등장하게 된다. 기업간 비즈니스 과정에서 발행하게 되는 발주, 납품, 계약서는 그 내용의 무결성 유지가 필수적이며, XML 문서의 무결성과 기밀성 등의 보안서비스가 제공되지 않는다면 XML의 편리함에도 불구하고 효율적인 활용이 어렵게 될 것이다.

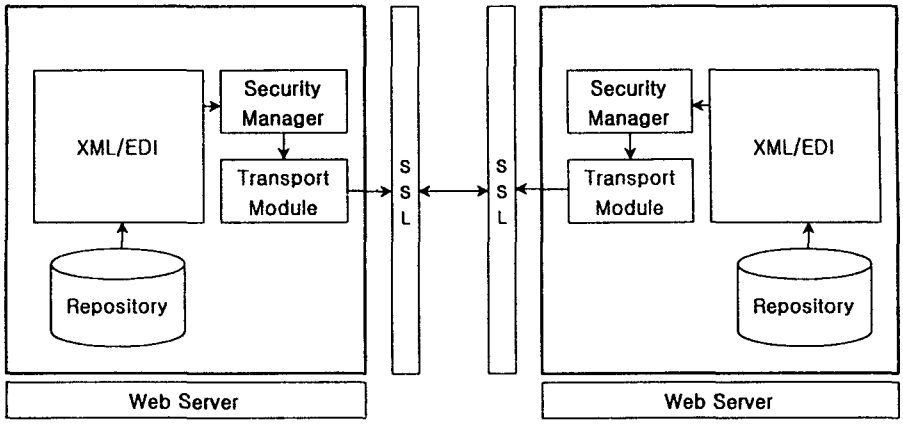
개방된 웹을 통해 전송되는 XML문서는 제 3자에 의해 도청되거나 변조될 위험성을 항상 갖고 있다. XML/EDI 시스템은 기존 시스템과의 통합이 용이하고, 인프라를 활용한 시스템 도입으로 비용과 시간을 절약할 수 있다. XML/EDI 시스템의 안전한 메시지 전송은 시스템의 신뢰성과 효율성을 높이고, 가치를 보유한 트랜잭션을 처리할 수 있어 시스템의 부가가치를 높이게 된다. XML 전자서명을 채택함으로써 전송되는 메시지의 무결성과 인증 그리고 부인방지의 보안요구사항을 충족시킬 수 있으며, XML/EDI 시스템에 적용하여 효율적인 메시지 전송과 발생되는 여러 보안 문제를 해결할 수 있다.

본 연구의 목적은 안전한 XML/EDI 메시징을 위하여 XML 전자서명시스템을 구현함으로써 B2B 전자상거래의 거래문서에 대한 메시지 인증과 무결성, 부인방지 기능을 향상하고 신뢰성 있는 전자상거래 환경을 조성하고자 하는 것이다.

2. 관련연구

2.1 XML/EDI 시스템

XML/EDI는 기존의 EDI를 통해 교환하던 거래정보에 대해 필요한 엘리먼트를 추출하여 XML DTD(Document Type Definitions)로 정의하고, 정의된 태그를 사용하여 구현하는 것이다. 기존의 EDI에서는 전송되는 메시지 중에서 데이터 항목을 분리하고 식별하기 위해서 독특한 세그먼트 식별자를 사용하여 왔는데, XML/EDI에서는 이러한 세그먼트 식별자를 XML DTD로 정의하여 데이터를 교환하도록 하는 것이다. XML/EDI는 전통적인 EDI에서 처리할 수 있는 업무의 한계를 벗어나 전자상거래 전반에서 필요한 프레임워크를 제공하고 있다. <그림 1>은 XML/EDI시스템의 구성도를 나타낸 것이다.

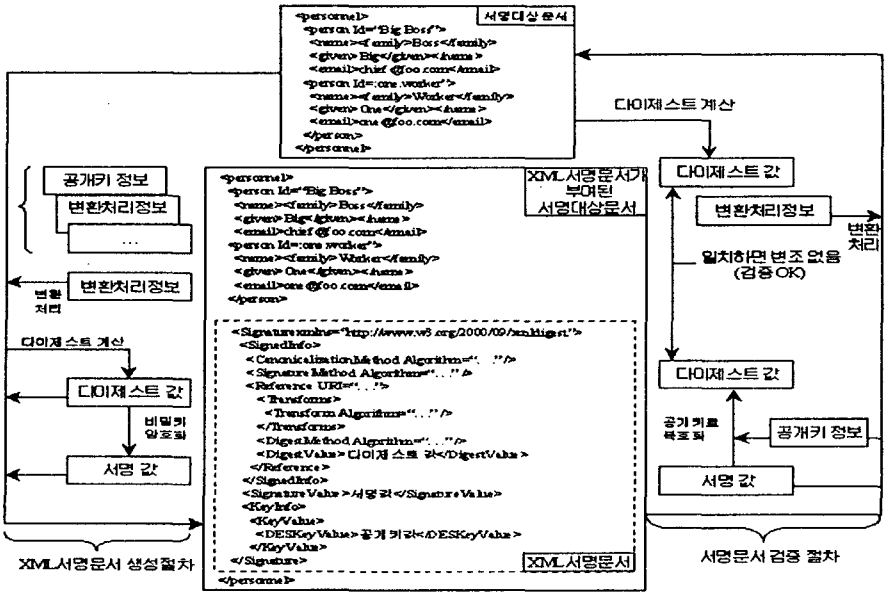


<그림 1> 인터넷 기반 XML/EDI시스템 구성도

일반적인 XML/EDI 시스템에서는 거래를 위해 필요한 문서의 DTD를 갖고 있으며, 문서의 형태를 정의하는 XSL과 전송데이터의 처리내용을 담고있는 템플릿을 참조하여 XML 문서를 작성하게 된다. 수신 측에서는 전용 브라우저나, HTML등을 활용하여 전송된 문서를 화면에 보여주게 된다. 이때 양측에서 참조하는 DTD, 스타일 시트, 템플릿 등은 수신자와 송신자 사이에 글로벌 저장소로서 존재하며, 송신측과 수신측에서 공유하여 사용하게 된다.

2.2 XML 전자서명

XML 전자서명이란 XML 전자문서를 작성한 사람의 신원과 문서의 변경여부를 확인할 수 있는 고유정보, 즉 XML 전자문서에 찍은 인감도장이나 사인이라고 할 수 있다[5]. 공개키 암호방식에 기반을 둔 전자서명은 비밀키로 문서에 전자서명을 하고 공개키를 이용하여 이를 검증한다. 전자서명은 인증(Authentication), 무결성(Integrity), 부인방지(Non-repudiation)의 보안서비스를 제공한다.



<그림 2> XML서명문서의 생성과 검증 절차

XML 전자서명은 해쉬 알고리즘으로 서명 대상문서에 대한 다이제스트를 생성하고, 전자서명 생성 알고리즘과 서명자의 서명 생성키를 사용하여 전자문서의 축약정보에 전자서명을 하고 서명문을 생성한다. 수신자는

서명 문서를 전자서명 검증 알고리즘과 서명자의 서명 검증키를 사용하여 수신된 전자서명으로부터 원본 문서의 다이제스트와 수신한 다이제스트를 비교하여 문서의 변조 여부를 확인한다[2].

XML 전자서명 문서는 DTD 및 XML 스키마를 이용해 정의된 XML 요소로 구성되며, 각 XML 요소는 계층구조를 갖는다. XML 전자서명은 W3C에서 정의하고 있는 XML-Signature Syntax and Processing 표준안을 따르고 있다[12]. XML 전자서명과 관련한 선행연구로는 NEC(Nippon Electronic Company), IBM(International Business Machines Corporation), ETRI(Electronic and Telecommunications Research Institute)의 XML전자서명 등의 구현사례가 있다[6, 10, 11].

NEC(2001)의 XML 전자서명 라이브러리는 XML 문서를 정규 XML문서로 변환하는 기능과 전자서명 생성 및 검증의 기능을 제공한다[10]. 자바 기반의 이 라이브러리는 Xerces Java Parser와 JCE 암호 프로바이더의 환경에서 동작한다. XML 서명 대상문서를 정규화 알고리즘을 적용해 문서를 정규화하고, RSAwithSHA1 이나 DSS알고리즘을 이용해 전자서명을 수행한다.

IBM(2001)의 XML Security suite(XSS)는 XML문서에 대한 전자서명을 생성 및 검증하는 기능, 정규 XML문서로 변환하는 기능, XML을 이용한 접근제어 기능을 제공한다. 또한 엘리먼트 단위의 데이터를 선택적으로 암호화하는 기능을 제공하고 있다[11]. XSS는 W3C의 XML 표준화 그룹에서 제공한 XML 전자서명, XML 암호화, XML 액세스 컨트롤 등의 표준을 구현한 참조구현 이다.

ETRI(2001)의 ESES(ETRI Secure E-Commerce Service)는 XML 전자서명 표준안에 기반하여 개발되었으며, XML 문서를 포함한 임의의 디지털 컨텐츠에 대한 무결성을 보장하고 인증을 제공하는 것을 목적으로 한다. XML 문서의 전자서명을 위한 정규화 기능, 메시지 다이제스트 기능과 필요한 여러 API를 제공한다. 또한 검증되어진 국내 표준알고리즘과 AES표준 알고리즘을 지원한다[6].

XML/EDI 시스템에 적용될 XML 전자서명을 수행하기 위해 사용되는 알고리즘은 W3C에서 제안된 알고리즘을 사용하게 되며, [표 1]과 같다[3].

[표 1] XML 전자서명 지원 알고리즘

알고리즘	관 련 요 소	알 고 리 즘	규약요건
다이제스트 계산	Digest Method	SHA1	REQUIRED
부호화	Transform	Base64	REQUIRED
MAC	Signature Method	Hmac-SHA1	REQUIRED
공개키 방식 서명	Signature Method	DSAwithSHA1(DSS)	REQUIRED
		RSAwithSHA1	RECOMMAND
정규화	Canonicalization Method	정규 XML	RECOMMAND
		정규 XML	REQUIRED
변환	Transform	XSLT	OPTIONAL
		Xpath	RECOMMAND
		Enveloped Signature	REQUIRED

3. XML 전자서명 설계

XML 전자서명은 XML 문서에서 디지털 콘텐트를 포함하고있는 부분이나 문서 전체를 대상으로 전자서명을 수행한다. Network Working Group Request for Comments에서는 전자서명 데이터모델과 구문에 대한 요구사항을 정의하고 있으며, 이러한 요구사항을 기반으로 설계한다[4]. 데이터 모델에 대한 요구사항은 (1) XML 서명의 자료 구조는 반드시 RDF 데이터 모델에 근거한 것이어야 한다. 하지만 RDF 직렬화 구문을 사용할 필요는 없다. (2) XML 서명은 주소로 지정될 수 있는 자원이면 모두 적용될 수 있다. XML 서명에서의 참조 대상은 재료 목록 안의 URL이나 프래그먼트 같은 XML 주소들에 의해 식별되는데, 이러한 주소들은 네트워크로 접근 가능한 외부 자원이나 같은 문서나 패키지 안의 내부자원을 가리킨다. (3) XML 서명은 XML 문서 전체 또는 일부분에 적용될 수 있어야 한다. (4) 고정된 웹 자원에 대해 여러 개의 키, 콘텐트 변환, 알고리

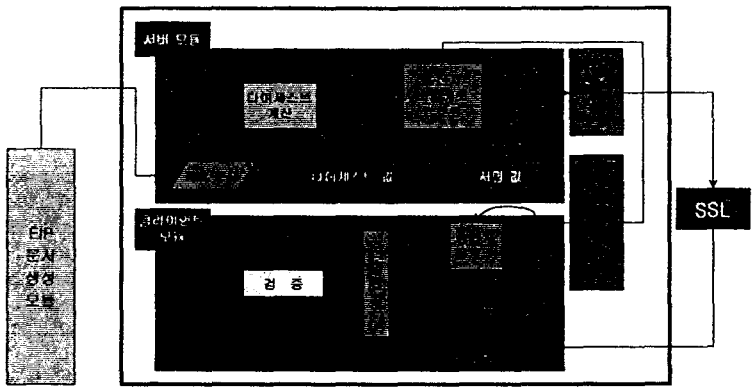
즘에 의한 다중 XML 서명이 존재할 수 있어야 한다. (5) XML 서명은 그 자체로 훌륭한 객체이며, 따라서 참조되고 서명도 될 수 있어야 한다. (6) 대칭 또는 비대칭 인증 체계나 동적인 키 재료의 합의 등과 같은 여러 가지 디지털 서명이나 메시지 인증 코드들의 사용이 명세서에서 허용되어야 한다. 자원이나 알고리즘 ID도 훌륭한 객체이며, URI에 의해 지칭될 수 있어야 한다. (7) XML 서명은 다른 문서에 포함되는 문서나 인코드 된 문서의 원래 버전에도 적용될 수 있어야 한다.

또한 Network Working Group에서는 서명 문서의 양식에 대한 요구사항을 규정하고 있으며, 내용은 다음과 같다. (1) XML 서명은 반드시 XML 원소이어야 한다. (2) XML 서명문이 문서 안에 배치될 때, 문서의 루트 원소 태그에 변화가 있어서는 안되며, 문서의 콘텐트 모델이 허용하는 위치에 서명을 추가하는 것 이외에 루트에서 시작된 세습 트리에 변화가 있어서도 안 된다. 예를 들면 , XML 폼은 서명된 후에도 여전히 그 응용 프로그램에 XML 폼으로 인식될 수 있어야 한다. (3) XML 서명은 구성 부분들의 서명 특성을 유지하면서 추가나 삭제에 의해 복합문서의 제작을 가능하게 하는 메커니즘을 반드시 제공해야 한다. (4) 분리된 웹 서명은 XML 서명의 중요한 용도 중의 하나가 될 것이다. 그러나, 서명은 XML이나 다른 방법으로 인코드 된 콘텐트를 캡슐화하거나 그 안에 끼워 들어 갈 수도 있다.

3.1 XML 전자서명 기능설계

3.1.1 XML 전자서명 기능설계 흐름도

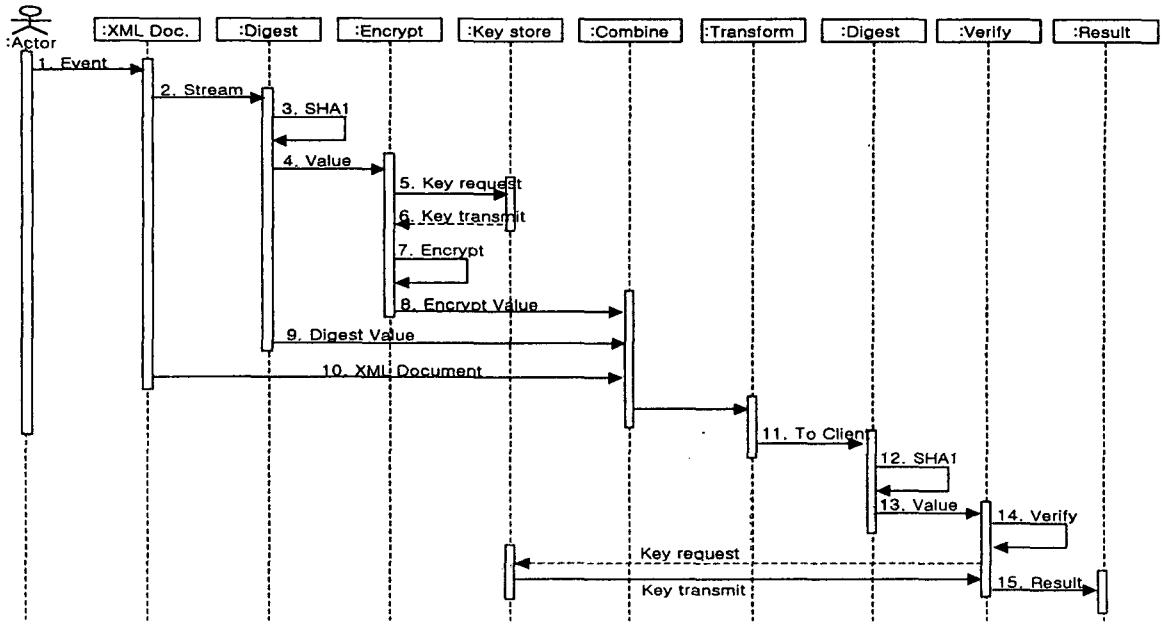
XML문서를 전자서명 처리하기 위해 업무응용시스템이나 XML/EDI 시스템으로부터 XML 문서를 읽어온 후 인증기관으로부터 수신자의 공개키를 받아 다이제스트 계산을 수행한다. 다이제스트 값을 송신자의 비밀키로 암호화하고, 서명 값을 생성한 후 원본 문서와 서명 문서를 합쳐서 클라이언트로 전송되고, 클라이언트는 인증기관으로부터 송신자의 공개키를 받아 서명 문서를 검증하는 절차를 거쳐 문서의 무결성을 확인한다.



<그림 3> XML Security Manager의 기능 흐름도

3.1.2 XML 전자서명 Sequence Diagram

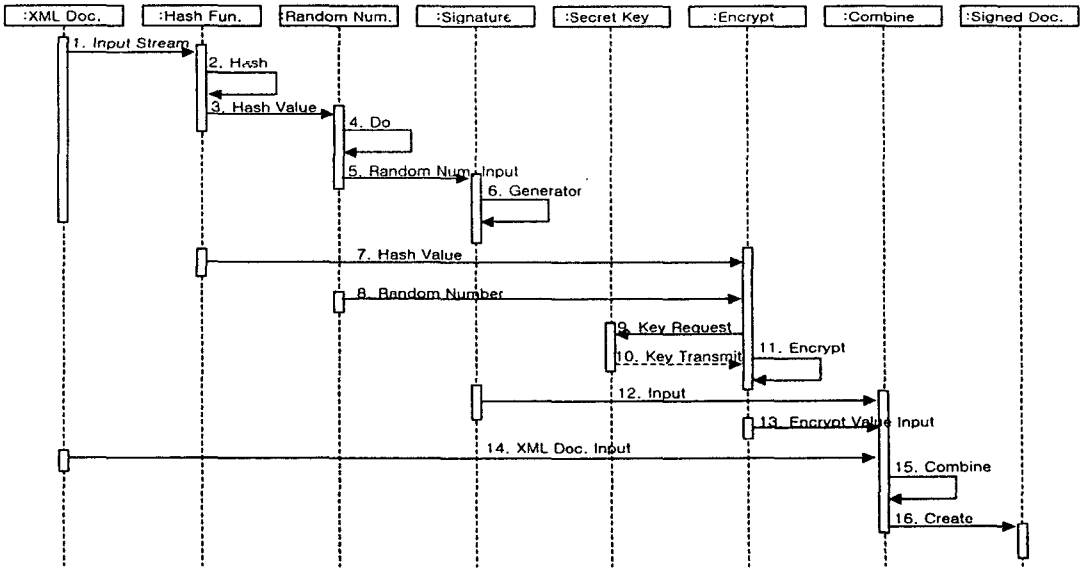
사용자는 XML 형식의 발주서 또는 납품서를 전자서명 하기 위해 서명 이벤트를 발생시킨다. XML 전자서명을 위해 키 저장소로부터 송신자의 비밀키를 요청하고 서명을 수행한다. 완료된 서명 문서는 전송모듈을 통해 클라이언트로 전송된다. 클라이언트는 수신한 XML 전자서명 문서의 무결성을 검증하고, 문서를 관리하게 된다.



<그림 4> XML 전자서명/검증 Sequence Diagram

3.1.3 XML 전자서명 생성 Sequence Diagram

XML 문서는 전자서명 수행 이벤트와 함께 해쉬 함수를 통과해 메시지 다이제스트를 생성하게 되고 송신자의 비밀키를 키 저장소로부터 요청하여 암호화한 다음 서명 문서를 생성한다. 생성된 XML 저자서명 문서는 XML HTTP를 이용해 클라이언트의 특정 폴더로 전송된다. 클라이언트에서는 수신한 문서를 검증하기 위해 검증 이벤트를 발생시킨다.

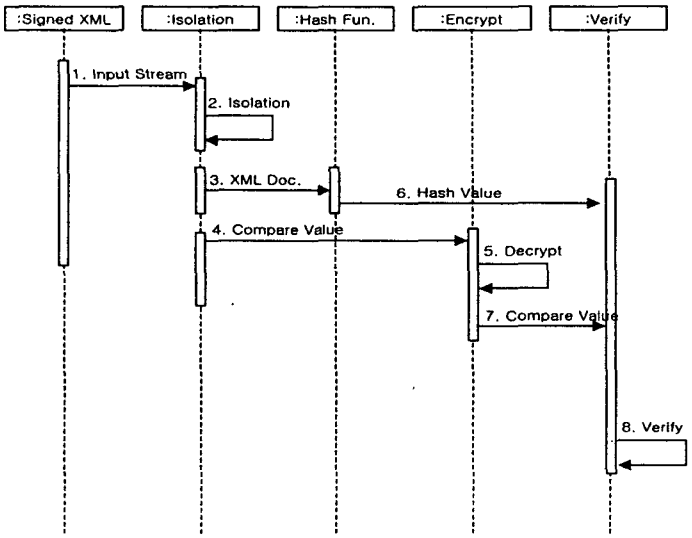


<그림 5> XML 전자서명 생성 Sequence Diagram

3.1.4 XML 전자서명 검증 Sequence Diagram

서버로부터 수신한 XML 전자서명 문서를 검증하기 위해 클라이언트는 메시지 다이제스트와 XML 문서 그리고 암호화된 비교 값을 분리한다. 수신한 다이제스트와 클라이언트 측에서 생성한

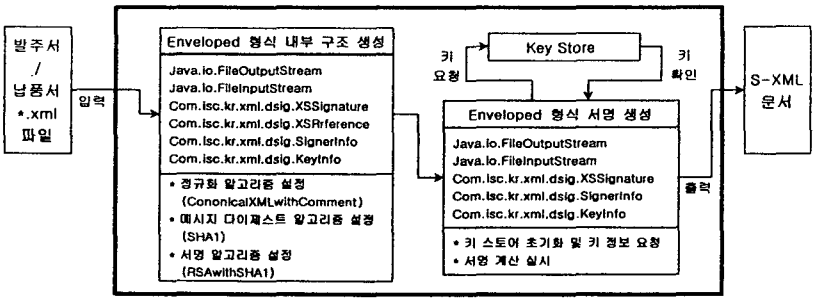
다이제스트를 비교하여 XML 전자서명 문서를 검증한다. 문서의 무결성이 확인되면 이 문서를 관리한다.



<그림 6> XML 전자서명 검증 Sequence Diagram

3.2 XML 전자서명 기능 명세도

XML 전자서명은 정규화를 위해 Canonical XML 알고리즘, 서명을 위한 DSAwithSHA1 알고리즘을 사용하고, 다이제스트 생성을 위해 SHA1 해쉬 알고리즘을 사용한다. 서명 생성에 필요한 키 정보를 위해 KeyInfo요소를 생성하고, 이를 기반으로 RSAKeyValue요소와 RSA 공개키 값을 설정한다. 생성된 XML 전자서명 문서의 무결성 검증을 위해 KeyInfo 요소의 X509Data 공개키에 관한 정보를 통해 검증하게 된다. <그림 7>은 전자서명의 생성과 검증 과정에서 사용되는 자바 패키지들의 관계를 나타낸다.



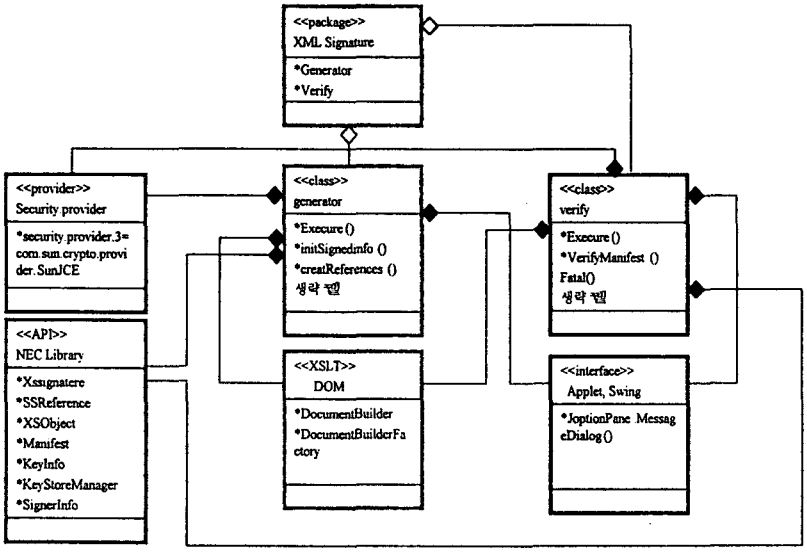
<그림 7> Security Manager 내의 자바 클래스 기능명세

서명을 처리하는 과정은 서명 대상문서를 입력받아 XML 문서로 타당한지 체크하고, 서명을 위한 비밀키를 요청한다. KeyInfo요소를 생성하여 DefaultRSA알고리즘을 설정한다. 이를 기반으로 RSAKeyValue요소와 RSA 공개키 값을 설정한다. 생성된 KeyInfo 요소는 Signature 요소에 포함되어 서명계산을 수행한 후 생성한다.

서명구조는 루트 요소인 <Signature>로 시작하여 SignedInfo와 KeyInfo에 서명 생성과 검증에 필요한 정규화 알고리즘, 서명 알고리즘, 변환 알고리즘, 다이제스트 알고리즘, 서명값 그리고 키 정보가 포함된다. 서명을 검증하는 단계는 생성된 XML 전자서명 문서가 제 3자에 의해 변조되지

않았는지의 무결성을 검증하는 단계이다. XML 전자 서명된 문서를 스트림으로 입력받아 KeyInfo 요소의 RSA공개키를 이용해 검증한다.

X.509 인증서 공개키 검증 단계는 XML 전자 서명된 문서에 존재하는 KeyInfo의 X509Data 요소에 공개키에 관한 정보가 포함되어있어 그 유효성을 검증할 수 있다. XML문서를 스트림으로 입력받아 XML문서의 유효성을 확인하고, 인증서로부터 읽어들인 키를 이용해 KeyInfo에 포함되는 공개키를 검증한다. <그림 8>은 XML 전자서명 기능명세도를 나타낸 것이다.



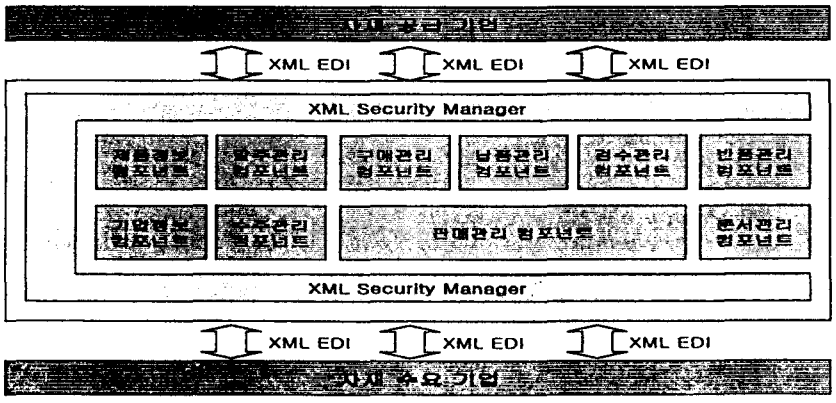
<그림 8> XML 전자서명 기능명세도

4. XML 전자서명 시스템의 구현

4.1. 시스템 적용대상

XML 기반 전자상거래 환경에서 트랜잭션들을 신뢰성 있게 지원하기 위해서는 XML/EDI 시스템 환경 구축에 있어서 온라인 인증을 위한 디지털 키 관리와 전자서명 및 데이터 암호화 등의 통합 처리가 용이해야 하며, 광범위한 애플리케이션들과의 상호 호환성 또한 확보되어야 한다.

기업 간 자재구매 활동은 자재 수요업체가 주문제품에 대한 발주서를 제공하면 법적 효력을 발생하고, 이에 대해 납기일을 고려한 자재 공급업체의 납품서를 전달하게 되며, 제조기업의 자재 검수 실적에 따라 검수서를 보내게 된다.



<그림 9> 구매시스템에서의 XML Security Manager 구성

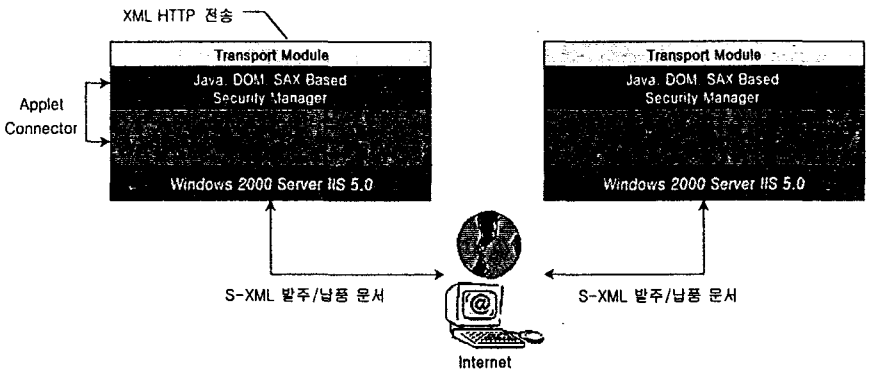
<그림 9>는 XML 전자서명 기능이 내장된 XML Security Manager를 구매시스템에 응용한 시

스템 구조도이다. 구매시스템이나 XML/EDI시스템에서 처리된 XML 문서는 최종적으로 XML Security Manager를 거쳐 전자서명 된 후 거래상대방에게 전송된다.

4.2 개발환경 및 시스템 구성

XML 전자서명기능은 자바 기반으로 구현되었기 때문에 개방형 플랫폼을 지원한다. JavaTM 2 Runtime Environment, Standard Editon Version1. 3 이후 버전에서 동작하며, XML 파서(Xerces Java Parser Version 1.2. 1)와 XSL 엔진(Xalan-Java : Version 1.2. 1), Java Cryptography Extension(JCE) Version 1.2이후 환경하에서 동작한다.

Windows 기반의 IIS를 웹서버로 사용하게 되며, 기본 인증은 사용자 이름과 암호를 Base64 인코딩에 의해 웹을 통해 전달하는 보안 방식이다. 이러한 방식은 구현하기 쉬운 반면 중간에 IP 패킷을 가로채서 디코드하면 사용자 이름과 암호가 다른 사람에게 노출될 가능성이 있다. 이것은 사용자가 자신의 자적인 정보를 서버에 보낼 때 목적지 서버가 올바른 서버인지에 대한 어떠한 확인 절차도 이루어지지 않는다는 의미를 내포한다. 따라서 좀 더 안전한 보안 전송 방식인 SSL(Secure Socket Layer)이라는 표준 보안방식을 함께 사용한다.



<그림 10> XML 전자서명 시스템의 Block Diagram

4.3 XML 전자서명 모듈 구현

서명된 XML문서는 원본 XML문서의 컨텐츠를 그대로 포함하면서, XML 전자서명문서 양식의 요구사항을 잘 만족하고 있다. 또한 RDF 데이터모델에 근거하고 있다. <그림 11>은 전자서명 이전의 XML 문서를 보여주고 있으며, <그림 12>는 전자서명 이후의 XML 문서를 보여주고 있다.

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!-- edited with XML Spy v2.5 NT - http://www.xmlspy.com -->
- <report>
    <title>XML siggnature</title>
    <author>limjaeyong</author>
    <date>2001.2. 20</date>
    <body>XML signature W3C...</body>
    <Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#" />
  </report>
```

<그림 11> 서명 전 XML 문서

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!-- edited with XML Spy v2.5 NT    http://www.xmlspy.com    -->
- <report>
    <title>XML siggnature</title>
    <author>limjaeyong</author>
    <date>2001.2. 20</date>
    <body>XML signature W3C...</body>
  - <Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
    - <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      - <Reference Id="REF_01" URI="">
        - <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        </Reference>
      </SignedInfo>
    </Signature>
  </report>
```

그림 12 서명 후 XML문서

4.4 XML 전자서명 구현 화면

전자서명 모듈의 구현은 웹기반 구매시스템에의 응용을 위해 실행파일의 형태로 구현되었으며, ActiveX를 통해 브라우저 상에서 구현이 가능하다. <그림 13>은 Security Manager의 메인화면을 보여주고 있다. XML 문서를 전자서명하기 위해 문서 서명 탭을 누르면, <그림 14>와 같이 서명할 파일을 선택한다.

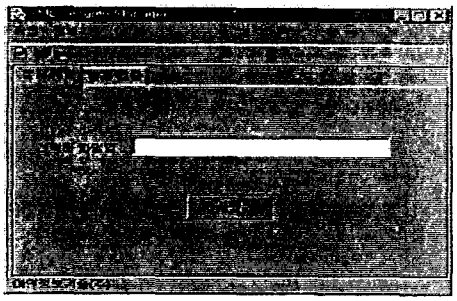


그림 13 Security Manager 메인 화면

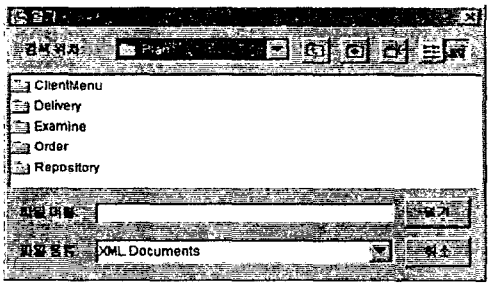


그림 14 서명 또는 검증파일 선택

그림 15 서명 대상파일 확인

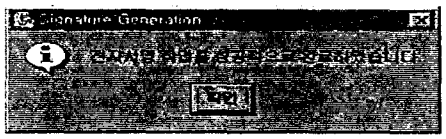


그림 16 서명처리 완료

<그림 14>에서 서명 버튼을 누르면 선택된 서명 대상파일을 확인하고, 서명 대상파일이 올바르게 선택되었는지 확인하면 <그림 15>와 같이 서명 작업이 수행된다. 서명이 성공적으로 완료되면 <그림 16>과 같이 서명완료 정보 창이 뜨게된다. 만일 서명이 어떠한 문제로 완료되지 못하였다면 원인과 함께 서명실패 경고 창이 뜨게 된다.

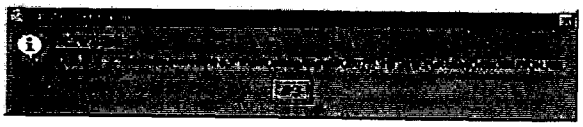


그림 17 검증 대상파일 확인

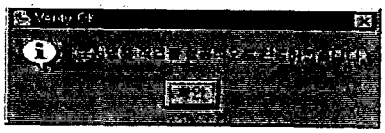


그림 18 검증 완료

<그림 17>은 전송된 XML문서를 검증하기 위한 화면을 보여주고 있다. 서명된 전자서명 파일을 검증하기 위해 Security Manager를 실행하고, 검증을 위해 문서 검증 탭을 누르고 검증대상 파일을 선택한다. 검증 버튼을 누르면 선택된 검증 대상파일을 확인하고, 검증 대상파일이 올바르게 선택되었는지 확인하면 검증작업이 수행된다. 검증이 성공적으로 완료되면 <그림 18>과 같이 검증성공 정보 창이 뜨게된다. 만일 서명된 문서가 변조되었다면 <그림 19>와 같이 검증실패 경고 창이 뜨게된다.

그림 19 검증 실패

4.5 XML 전자서명 요구사항 분석

본 논문에서 구현한 XML 전자서명 시스템은 W3C의 Network Working Group에서 제안한 XML 전자서명 요구사항을 다음 [표 2]와 같이 만족하고 있다.

[표 2] XML 전자서명 보안요구사항 만족 정도

요 구 사 항		만족함	만족하지 않음
구조	1. RDF 데이터모델	○	
	2. XML 주소에대한 식별 가능성	○	
	3. XML 문서 전체 또는 일부 적용성	○	
	4. 다중 XML 서명기능		○
	5. XML서명문서 참조 가능성	○	
	6. 디지털 서명이나 MAC 사용의 다양성	○	
	7. 포함 또는 인코드된 문서의 원본에 적용가능성		○
양식	1. 서명은 XML의 원소로 존재	○	
	2. 원본 문서 루트요소의 불변	○	
	3. 서명의 특정유지하면서 복합문서 제작 가능		○
	4. 서명은 인코드된 컨텐츠를 캡슐화 가능해야 함		○

5. 결론

B2B 전자상거래가 활성화되면서 XML/EDI 시스템은 기존 EDI를 통한 어플리케이션간 데이터 교환방식이 갖는 문제점을 효율적으로 해결할 수 있는 방안을 제시해 주고, 어플리케이션간 인터페이스가 필요한 분야들에서 실용적인 해결책으로 인식되고 있다. 이에 본 연구는 인터넷을 통한 XML 문서의 안전한 전달을 위해 XML 전자서명을 구현하고 이를 XML/EDI시스템에 적용하였다.

본 논문에서 구현한 XML 전자서명 시스템은 XML 전자서명을 채택함으로써 전송되는 메시지의 무결성과

인증 그리고 부인방지의 보안요구사항을 충족시킬 수 있으며, XML/EDI 시스템에 적용하여 효율적인 전자조달과 발생되는 여러 보안 문제를 해결할 수 있다.

본 연구의 한계점은 인증기관으로부터 인증서를 받아오는 기능이 구현되지 못하였으며, 또한 암호화와 전자서명을 동시에 구현하지 못하였다. 이를 해결하기 위한 향후 연구 과제로는 CA( )와의 연동을 위한 연구가 진행되어야 할 것이며, 또한 기업간 비즈니스에서 보다 강력한 보안서비스를 제공하기 위해 전자서명과 함께 암호화가 이루어지는 보안서비스에 대한 연구가 필요하다.

참고문헌

[1] 장우영, 유승범, 장인걸, 차일석, 신동일, 신동규, "XML/EDI 와 XML 전자서명 통합시스템의 설계", 한국 정보처리학회 논문지 제 8권 제 1호, 2001, p407

[2] 송유진, 이희권, 전형득, 한승현, 권헌숙, "전자상거래를 위한 차세대 정보보호서비스에 관한 연구", 한국전자통신연구원, 연구보고서, 2001, p11

[3] W3C, "Extensible Markup Language", http://www.w3c.org/xml

[4] Network Working Group Request for Comments 28087, "XML Signature Requirement", http://www.ietf.org/rfc/rfc2807.txt

[5] 한국후지쯔, "XML과 활용분야" http://kr.fujitsu.com/webzine/tech/issue/xml_use/feature/

[6] 이주영, 김주한, 이재승, 문기영, "안전한 전사상거래 플랫폼 개발을 위한 ESES의 구현", 한국전자통신연구원, 정보처리학회 논문지, 제 8-C권 제 5호, 2001, p293

[7] 원덕재, 이형석, 송준홍, 신동규, 신동인, "XML Signature에 기반한 XML/EDI System의 설계 및 구현", 한국정보처리학회 논문지 제 9권 1호, 2002년

[8] 문기영, 이주영, 박치항, "XML 기반 정보보호 서비스", 한국전자통신 연구원

[9] 송세봉, 장의진, 고훈, 신용태, "Secure XML 메시지 전송시스템 설계", 한국정보처리 학회 논문지 제 8권 1호

[10] NEC, "NEC XML Signature Software Library", http://www.sw.nec.co.jp/soft/xml_s/

[11] IBM, "IBM XML Security Suite", http://alphaworks.ibm.com

[12] W3C, "XML signature syntax and processing", http://www.w3.org/TR/xmldsig-core/