XML 전자서명을 이용한 XML/EDI 보안에 관한 연구

- A Study on XML/EDI Security using XML Signature -

고규준 *
Go Gyu Joon
정경찬 *
Jung Kyung Chan
박정선 *
Park Jeong Sun

Abstract

As the Internet spreads rapidly, the industrial structure is changing to a new paradigm. The previous EDI system was asked to be changed and WEB EDI, E-Mail EDI, FTP EDI etc. based on the internet have appeared. Recently, the XML/EDI which is based on XML documents has appeared.

EDI system has to assure a safe exchange between senders and receivers. But, the internet has security problems because it uses a open TCP/IP protocol. In this paper, we will propose an XML/EDI system with XML Signature.

1. 서 론

오늘날 많은 주문서, 송장과 같은 B2B(Business to Business) 문서들은 EDI(Electronic Data Interchange) 메시지들로서 교환된다. 이러한 EDI가 인터넷의 보급이 확산됨에 따라 문서전달을 위해 굳이 고비용의 사설망을 사용하기보다는 인터넷망을 기반으로 하려는 움직임이 시작되었고, 다양한 형태의 EDI들이 나타나게 되었다.

그러나, 인터넷은 TCP/IP라는 개방적인 프로토콜을 사용함으로 인해 사설망에서의 네트워크 구조보다 보안상의 위험에 많은 부분 노출되어지게 된다.

따라서, 인터넷을 기반으로 하는 시스템에서는 보안을 반드시 고려해야 한다. 현재 SSL, S/MIME, PGP 등 인터넷 보안 서비스를 위해 많은 표준들이 제공되어 지고 있으며, XML 문서에 적합한 보안 서비스로 XML-Encryption, XML- Signature가 논의되고 있다. 본 논문은 이러한 사회적 변화에 대응할 수 있는 XML/EDI 시스템과 이 시스템에 XML-Signature를 사용하여 얻을 수 있는 장점에 대하여 알아보고, 이를 활용하여 XML-Signature를 통해 서명을 거친 문서를 전달하는 과정을 구현하는 것이 연구목적이다.

* 명지대학교 산업공학과

2. EDI

2.1 EDI 시스템

EDI시스템은 크게 응용 소프트웨어, 변환처리 소프트웨어(Translator), 통신 소프트웨어, 메시지 처리 시스템(Message Handling System: MHS), 그리고 사용자단말기 등으로 구성된다[5].

응용소프트웨어는 EDI 사용자가 문서를 작성할 때 EDI 표준에서 정한 항목을 만족시킬 수 있도록 하며, 거래상대방에게 송신하기 위하여 문서를 작성, 수정, 조회, 삭제하는데 활용되고 수신된 전자문서를 조회, 출력하는 데에도 사용된다. 변환처리 소프트웨어는 EDI표준과 연결되어, 사용자 응용시스템에서 작성한 데이터 파일을 EDI표준 양식으로 변환하는 소프트웨어이다. 사용자 고유양식에서 표준양식으로 표준형식에서 사용자의 고유양식으로 변환해주는 기능을 한다. 통신소프트웨어는 시스템과 시스템간에 데이터를 주고받을 수 있도록 하는 기능을 지원한다. MHS(Message Handling systems)응용서비스는 메시지를 정의된 봉투와 내용에 넣어 전달하는 역할을 한다. 주요 서비스로는, Composition(메시지와 응답을 생성), Transfer(수신인에게 메시지 전송), Reporting(발신인에게 메시지에 생긴 일을 보고), Conversion(수신인의 터미널이나 프린터에 디스플레이), Formatting(수신인의 변환처리 가능한 상태로 변환), Disposition(수신인이 메시지를 받은 후 하는 작업) 등과 같은 것이 있다. 사용자 단말기는 실제로 사용자가 EDI를 사용하는 컴퓨터를 말한다.

2.2 인터넷 EDI

인터넷 EDI 서비스는 거래 처리를 위한 문서 교환 방식과 지원 서비스에 따라서 그 수준이 다양하다. 교환 방식으로는 문서 파일을 교환하는 방법과 HTML 폼을 사용하는 방법이 있다. 전자는 자체적으로 생성한 EDI 문서 파일을 단순히 인터넷을 통해 웹서버로 전달하고, 수신자는 웹서버에 접속해 파일을 다운로드하여 수신하는 방법이며, 후자는 웹사이트에 접속하여 미리 짜여진 HTML 폼에 데이터를 입력, 수정하여 데이터베이스에 저장하였다가 수신자가 데이터를 검색하는 방법이다. 웹서비스 제공업체가 HTML 폼을 위해 다양한 템플릿을 제공하거나 폼을 편집할 수 있는 서비스를 제공하는 방법도 있다.

2.3 인터넷 EDI의 문제점

인터넷 EDI는 자체 네트워크를 갖지 못한 사용자들도 손쉽게 EDI 문서를 전송할 수 있으며, 단기간 내에 거래를 처리할 수 있고, EDI를 위한 별도의 소프트웨어가 필요없다는 점이 VAN/EDI에 비해 많은 장점이 되었다.[6]

그러나, 개방적인 TCP/IP 프로토콜을 이용함에 따라 보안상의 위험이 존재하게 된다. 전송되는 메시지의 분실이나, 변조, 가장 등의 위협 요소가 존재한다.

3. XML/EDI

3.1 XML/EDI의 필요성

전통적인 EDI의 단점을 극복 할 수 있는 XML의 장점은 다음과 같다.

(1) XML은 필요한 요소를 추출하여 DTD를 작성함으로써 다양한 형태의 EDI문서도 교환이 가능하다.

(2) EDI 표준의 변화 또는 사설 표준의 변화에 따라 추가적인 엘리먼트가 요구 될 경우 DTD를 수정함으로서 쉽게 해결 할 수 있다.

(3) 교환을 위해 필요한 엘리먼트를 추출하여 자신에게 적절한 DTD를 개발함으로서 XML은 문서 구조정보를 가질 수 있다. 이는 EDI 시스템과 데이터베이스와의 상호 연계를 위한 경우에도 XML/EDI 태그가 바로 데이터베이스의 스키마로 매칭 될 수 있기 때문에 직접적으로 연계될 수 있어 중요한 의미를 가지며, 검색의 경우에도 강력한 기능을 발휘할 수 있다.

4. 암호시스템

4.1 대칭키 암호 시스템(Single-Key Cryptosystem)

대칭키 암호 시스템은 데이터를 암호화하는 키와 복호화하는 키가 같거나 하나의 키를 알았을 때 다른 키를 쉽게 알 수 있다. 이를 위해서는 송신자와 수신자가 미리 키를 공유하고 있어야 한다. 대칭키 암호 시스템은 암호화 키 크기가 공개키 암호 시스템 보다 상대적으로 작기 때문에 효과적인 암호시스템을 구축할 수 있다. 그러나 정보교환 당사간에 같은 키를 공유하여야 하므로 다수의 사람과 정보교환이 필요할 경우에는 많은 키를 유지 관리해야 하는 문제점이 있다.[1]

4.2 공개키 암호 시스템 (Public-Key Cryptosystem)

암호화하는 키와 복호화하는 키가 서로 다르고, 하나를 알더라도 그에 대칭되는 키를 알기 어려운 암호 시스템을 말한다. 두 개의 키 중에서 하나의 키를 공개하고 나머지 하나를 비밀키로 자신이 보관하여 사용하는 것이다.[1] 여기서 공개되는 키를 공개키(public key)라고 하며 자신이 보관하는 키를 비밀키(private key)라고 한다. 사용자 A가 사용자 B에게 통신할 때 A는 B의 공개키를 이용하여 암호화해서 보내면 B는 받은 메시지를 자신의 비밀키로 풀어 내용을 확인할 수 있다. 따라서 공개키 암호 시스템의 사용자는 오직 자신의 개인키만 보관하면 되기 때문에 대칭키 암호 시스템보다는 훨씬 적은 수의 키가 관리된다. 그러나 암호화를 위한 키 크기는 상대적으로 크기 때문에 데이터 처리량이 적고, 연산 수행 능력이 떨어진다.

5. XML-Signature

전자서명은 문서나 메시지를 보낸 사람의 신원이 진짜임을 증명하기 위해 사용되는 서명이다. 이것은 또한 전달된 메시지나 문서의 원래 내용이 변조되지 않았다는 것을 보증하기 위해 사용될 수도 있다. 전자 서명에 대한 조건 및 특징은 <표 1>과 같다.[14]

< 표 1 > 전자 서명의 조건 및 특징

조 건	특 징
· 서명은 서명되고 있는 메시지에 의존 하는 형태이어야 한다. · 위조와 부인을 방지하기 위해 송신자에게 있어서 유일한 어떤 정보를 이용해야만 한다. · 서명을 만들기가 비교적 쉬어야 한다. · 서명을 인식하고 확인하기가 쉬어야 한다. · 전자 서명을 위조하는 것이 계산적으로 실행 불가능해야 한다. · 기억 장소에 전자 서명의 복사본을 유지하는 것이 실용적이어야 한다.	· 그 서명의 저자와 날짜와 시간을 확인 할 수 있다. · 서명할 때의 내용을 인증할 수 있다. · 서명은 분쟁을 해결하기 위해서, 제삼자에 의해서 확인될 수 있다.

전자 서명은 공개키 암호화시스템을 사용하는 직접 서명 방식과 신뢰할 수 있는 제삼자를 통해 서명을 생성/검증하는 중재자를 통한 간접 서명 방식으로 나눌 수 있다. 직접 서명 방식에는 다시 메시지 복원형 전자 서명 방식과 부가형 전자 서명 방식으로 나눌 수 있다. 두 전자 서명 방식 중 상대적으로 부가형 디지털 서명의 장점이 더 크고 현재 세계적인 추세도 이 방식을 선호하고 있다.

5.2 XML-Signature의 구조

XML Signature의 구조는 크게 Signature내에 SignedInfo의 필수 엘리먼트로 이루어진다.[15] Signature 엘리먼트의 하위 엘리먼트로 SignatureValue, KeyInfo, Object 엘리먼트가 포함될 수 있으며, KeyInfo와 Object는 Signature 엘리먼트 이내에 포함되어져야 한다. SignedInfo 엘리먼트의 하위 엘리먼트로 Canonicalization Method, SignatureMethod, ObjectReference 엘리먼트를 포함할 수 있다.

```
<Signature>
<SignedInfo>
(CanonicalizationMethod)
(SignatureMethod)
(ObjectReference)+
</SignedInfo>
(SignatureValue)
(KeyInfo)?
(Object)*
</Signature>
```

< 그림 5 > XML Signature의 기본적인 구조

5.3 XML Signature의 유효 검증

XML Signature가 포함된 문서는 다음 세 가지에 대해서 만족할 때 서명이 유효하다[14].

(1) 서명 검증(Signature Validation)

SignatureValue가 CanonalizationMethod와 SignatureMethod를 내포하는 SignedInfo를 처리하여 얻은 결과값과 일치하는 가에 대한 검증을 말한다.

(2) 참조 검증(Reference Validation)

참조된 URI의 DigestValue와 SignedInfo에 내포되어 있는 DigestValue가 일치하는가에 대한 검증을 말한다.

(3) 신뢰/응용 제품에 대한 검증

응용 프로그램은 서명된 것을 신뢰하는가에 대한 검증을 말한다. 예를 들어, 사용된 키가 충분히 강한지와 서명이 얼마나 오래되었는지와 같은 지문에 대한 검증을 말한다.

5.4 Processing(처리) : 서명 생성과 검증에 관한 것

XML 문서에서 일련의 서명생성과 검증의 순서는 다음과 같이 한다[15].

(1) Reference Generation (참조 생성) : 서명될 각 object에 대해서 다음과 같은 순서로 생성한다.

① 응용 프로그램에 의해서 data object에 Transforms를 적용한다.

② Transforms에 의한 결과값에 대해서 digest value를 구한다.

③ reference 요소를 생성한다.

reference 요소내의 내포될 수 있는 요소들 :

- data object에 대한 identification (선택)
- transform (선택)
- digest 알고리즘
- DigestValue

(2) Signature Generation (서명 생성)

① SignatureMethod, CanonicalizationMethod, Reference를 가지는 SignedInfo를 생성한다.

② SignedInfo에 지정된 알고리즘에 의하여 SignedInfo에 대한 SignatureValue를 Canonicalize를 수행한 후에 계산한다.

③ SignedInfo, Object, KeyInfo, SignatureValue를 포함하는 Signature를 생성한다.

(3) reference Validation(참조 검증) : SignedInfo에 있는 각 Reference에 대하여 다음과 같은 순서로 검증한다.

① SignedInfo 내의 Canonicalization Method 에 기반하여 SignedInfo 요소를 canonicalize 한다.

② digest될 data object를 얻는다.

③ Reference에 지정된 DigestMethod를 이용하여 data object를 digest한다.

④ SignedInfo Reference에 있는 DigestValue와 위에서 생성한 digest 결과값을 비교하여 일치하면 검증 성공 그렇지 않으면 검증 실패가 된다.

(4) Signature Validation (서명 검증)
① CanonicalizationMethod를 기반으로하여 SignedInfo 요소를 canonicalize한다.
② KeyInfo 혹은 외부에서 검증 키(validation key)를 획득한다.
③ SignedInfo내의 SignatureMethod에 기반하여 SignatureValue를 검증한다.

6. XML-Signature를 적용한 XML/EDI 시스템

공급자와 구매자는 ASP(Application Service Provider)업체를 통해 인터넷 상에서 견적이나 주문서, 송장과 같은 거래문서를 교환한다. 구매자와 공급자는 거래를 위해 서로에게 문서를 주고 받을 수 있으며, ASP업체에서는 보안과 인증, 데이터 변환 서비스 등이 제공되어진다. 본 논문의 구현에서는 보안을 위한 암호화는 고려하지 않았으며, 인증을 위해 XML-Signature를 사용하였으며, 문서 전달과 문서 파일 관리에 중점을 두어 ASP 업체의 역할은 공급자와 구매자간에 전달되는 문서를 관리, 전송과 부가적인 서비스를 제공하는 것으로 제한하였다.

6.1 시스템의 개발 환경 및 구성

본 시스템의 구현을 위한 구축 환경으로는 표 3과 같다.

표 3. 시스템 구축 환경

운영체제	Windows 2000 Server
DBMS	MS SQL Server 7.0
Web Server	IIS 5.0, Resin 2.0.4
개발도구	Boland J-Builder, Visual InterDev 6.0
Web Browser	Internet Explorer 5.0
개발 언어	ASP, JSP, JavaScript, Java

인터넷 가상서점의 인터넷 서비스를 위하여 NT 4.0의 운영체제와 IIS 4.0 웹서버를 사용하여 인터넷 서버를 구축하였다. 데이터베이스와의 연동과 XML 데이터 전송을 위해 ASP와 Java Script를 사용하여 개발하였으며 클라이언트 측 기능을 위하여 Java Applet을 사용하였다. 또한, 효과적인 웹페이지와 스크립트를 작성하기 위하여 비주얼 인터데브 6.0을 사용하였다. 웹서버에서의 자바 처리를 위해 Resin을 이용하였다. XML 분석을 위한 파서로는 MSXML 파서와 Xerces를 사용하였으며 Xalan으로 JavaApplet에서의 DOM 처리를 하였다. 클라이언트 파일 접근에 따른 권한의 해결책으로 Microsoft SDK for Java를 이용한 Signed Applet을 사용하였으며, Applet에서의 키생성 및 암호화 처리를 위해 JCE를 사용하였다.

6.2 시스템의 DB Table, ERD

인터넷 가상서점의 서비스 구축을 위해 사용된 Data Type 과 Field를 정리하면 표4, 표5 와 같다.

표 4. 회원 정보 테이블

Field 명	Data type	내 용
member_id	char	회원 아이디
member_name	varchar	회원 이름
member_address	varchar	회원 주소
member_phone	varchar	회원 전화번호
member_pwd	char	회원 비밀번호

표 5. 메시지 테이블

Field 명	Data type	내 용
message_no	varchar	메시지 번호
message_title	varchar	메시지 제목
message_receiver	varchar	메시지 수신자
message_sendor	varchar	메시지 송신자
message_content	varchar	메시지 내용
message_file	varchar	첨부 화일
message_sendtime	varchar	메시지 송신시간
message_receivetime	varchar	메시지 수신시간

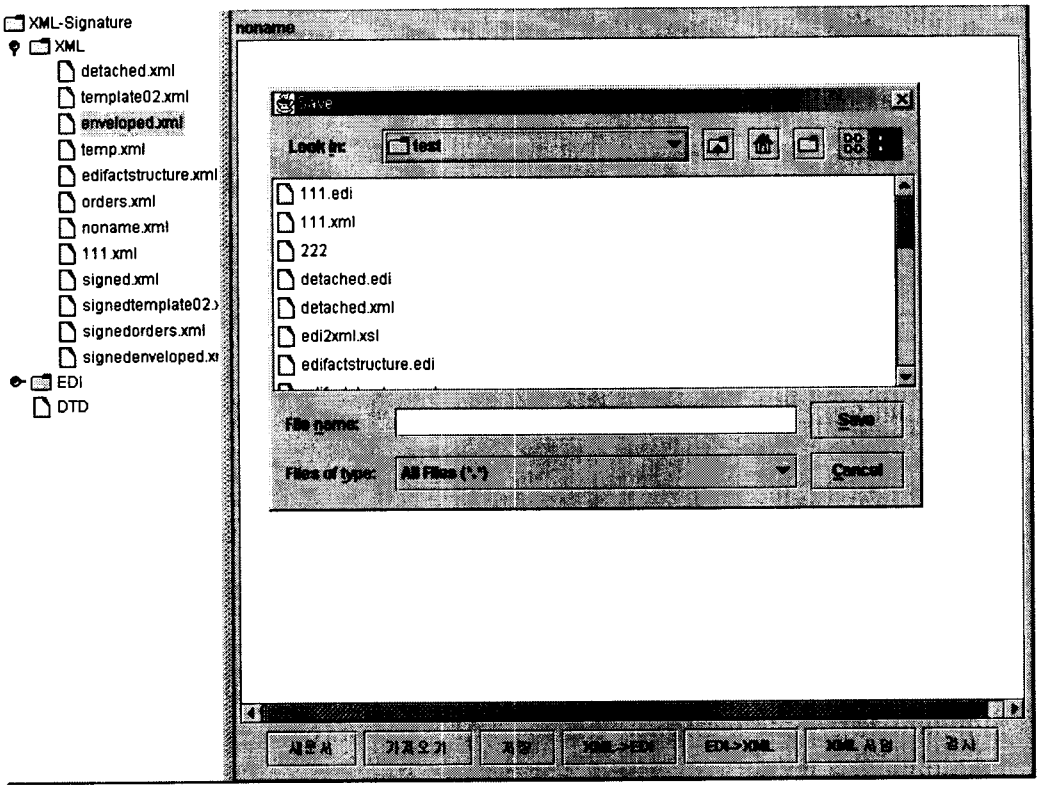
6.3 XML-Signature 처리 과정과 XML/EDI 시스템의 기능

6.3.1 로그인 기능

로그인(Login) 기능은 고객이 이 시스템을 사용할 수 있는지를 검사하여 외부 사용자나 접근이 제한되어 있는 사용자들이 들어오지 못하도록 하는 기능을 한다.

6.3.2 클라이언트 측 문서 파일 관리 기능

구매자와 공급자는 송, 수신을 위해 웹 사이트에 접속하여 문서를 작성하고 저장 및 삭제할 수 있다. 서명된 애플릿을 통해 제공되어지므로 그림 8 과 같이 클라이언트 측 파일 관리를 할 수 있다.



< 그림 8 > 클라이언트 측 파일 관리

6.3.3 XML문서의 EDI문서 변환 및 역변환 기능

전통적인 EDI 문서를 XML로 변환 또는 역변환 하는 기능을 한다. EDI 표준으로 작성된 문서를 애플릿을 사용하여 XML로 변환한다. XML문서에서 EDI문서로 변환하는 과정은 3가지 단계를 거친다.

① XML 문서를 읽어 들인다.

② 두 구조사이에서의 변환. XML요소를 EDI segment로 변환한다.

③ EDI Syntax에 따른 EDI 문서를 작성한다.

XML 파서는 XML 문서를 읽어들이는 것을 지원하며, XSLT 프로세서는 두 구조사이의 변환을 지원한다. 자바는 레거시 XML 문서를 EDI로 변환한다[8].

6.3.4 송신자의 XML-Signature 생성

서명의 생성은 NEC의 API를 사용하여 XML-Signature를 생성하였으며, Enveloped Signature 방식을 사용하여 XML 문서 내에 Signature 엘리먼트를 삽입시키도록 하였다.

그림 9의 서명이 포함된 문서의 내용을 보면 SignatureMethod에 서명을 위해 RSA-SHA1 알고리즘을 사용하고, 다이제스트를 위해 SHA1알고리즘을 사용하도록 명시되어 있음을 알 수 있다[14].

```
<?xml version="1.0" encoding="UTF-8"?>
<Order confirm="true">
    <Date>2000-03-10</Date>
    <Reference>AGL153</Reference>
    <DeliverBy>2000-04-10</DeliverBy>
    <Buyer>
        ---- 중략 ----
    </Seller>
    <Lines/>
    <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"/>
              <DigestValue>ITJ3hTi/YJwcKuPa+HcGqCPfxzI=</DigestValue>
            </Reference>
        </SignedInfo>
<SignatureValue>W6Z5FLLfyz33Bxs5UhqS0borkIgSD1rMyG8MIRx1FPs5R/kCo2beUxLORZz/gbuuwJSe2u6pFLLw3iSblyY81ACNwcWjF/edbK
6hVcj1I4AVowB8eqFMgiyeE2+dFtVnVkNemRjoDPBGpEyHSryaijZUwWlfTK3ognvulzXdbFU=</SignatureValue>
        <KeyInfo>
            <KeyValue>
              <RSAKeyValue>
<Modulus>AKwP1kqyqF9SLidkOXQ3jyxQL3fbOL/nqT1ejC4b+H1y5s99/Qs/Udp5pS5lXqq89/HwG1DopbMUry2PP+Yo6rV41rzG5/2yOr/M7HtJn
yhBOSgkl9RtfY+yLLZ51uNyy9SepG5mL/MAv4utCc9U8TLo4yhCVOBeLcsfs/loNa15</Modulus>
                  <Exponent>AQAB</Exponent>
              </RSAKeyValue>
            </KeyValue>
        </KeyInfo>
    </Signature>
</Order>
```

그림 9. 서명이 포함된 파일

Signature 엘리먼트 생성과정은 세가지 단계로 이루어 진다.

① 비어있는 Signature 엘리먼트를 문서안에 삽입한다.
② 삽입된 Signature 엘리먼트안에 서명을 위한 Method들을 정의한다.
- CanonicalizationMethod
- DigestMethod
- SignatureMethod

③ 참조된 URI와 Method들로 값을 생성해 Signature 엘리먼트에 첨가한다.
- DigestValue : DigestMethod에서 정의된 SHA1 알고리즘으로 계산된 결과값이다. 수신자가 검증할 경우 URI의 내용을 다시 Digest해서 이 값과 비교해서 문서내용에 변조가 있는지를 검증하게 된다.
- RSAKeyValue : 송신자의 공개키 값으로 수신자가 이 문서를 수신했을 경우 이 키값으로 서명을 검증할 수 있다.
- SignatureValue

6.3.5 수신자의 XML-Signature에 대한 유효성 검증 기능

송신자로부터 받은 XML 문서는 첨부되어 있는 송신자의 서명을 통해 전달과정에서 일어날 수 있는 문서의 변조와 정당한 문서전달을 확인할 수 있다. 앞에서 언급한

XML-Signature의 세 가지 검증에서 서명 검증과 참조 검증에 대한 검증을 처리하였다. 그림 10은 문서의 무결성을 검증하는 화면이다.

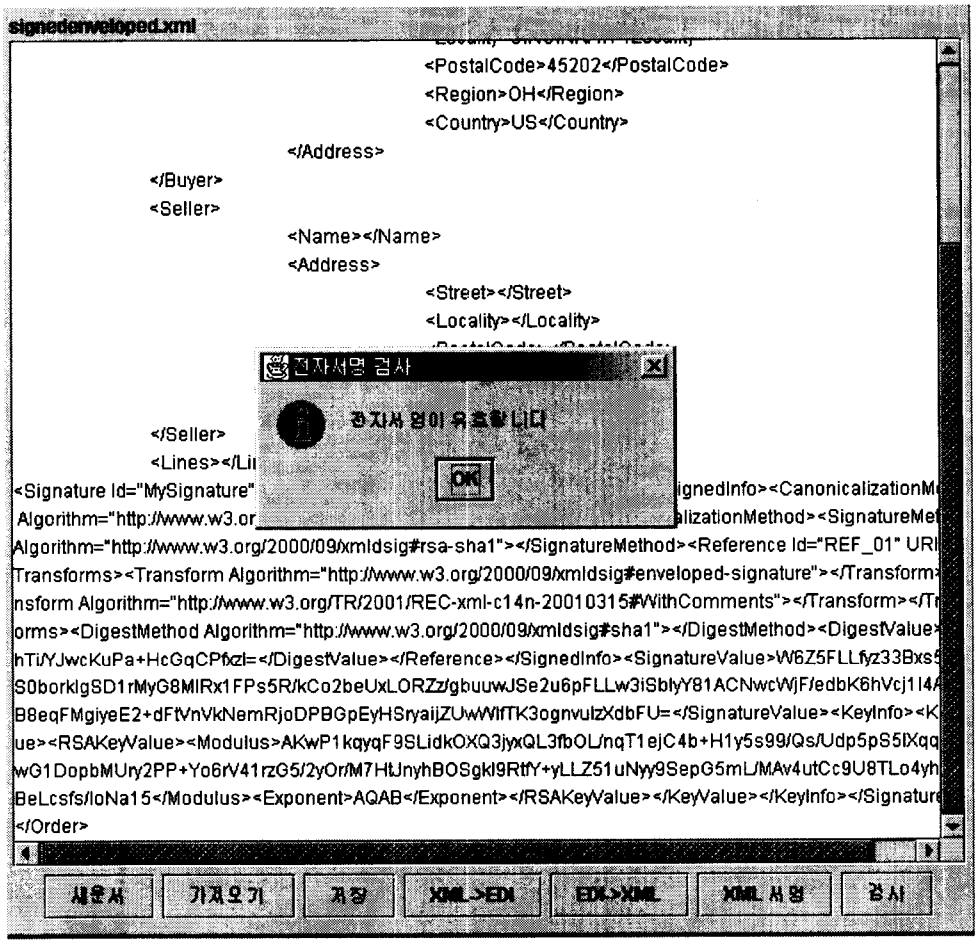


그림 10. XML-Signature를 통한 문서의 무결성 검증

7. 결론 및 향후 연구과제

본 연구에서는 XML/EDI 시스템에서의 인증과 무결성을 위하여 XML-Signature를 사용하여 구현하였다. 또한 XML/EDI 시스템의 구조와 장, 단점을 분석하고 XML-Signature를 통해 인터넷 보안의 약점을 해결할 수 있도록 하였으며, 서명 모듈을 클라이언트 측에 제공함으로써, 보안 문제에 좀 더 접근할 수 있도록 하였다.

추후로 XML-Signature에 초점을 두어 본 연구의 시나리오에서 배제되었던 암호화 모듈을 추가함으로써 좀 더 확실한 안전성을 갖춘 기능을 제공할 수 있도록 해야할 것이다.

8. 참고문헌

[1] 김철, "암호학의 이해", 영풍문고, 1996.12.
[2] 김형도, "B2B 전자상거래 @XML", 배움터, 2000.12.
[3] 박창섭 , "암호이론과 보안", 대영사, 2001.8.
[4] 이종호, "XML과 전자상거래", 정보문화사, 2001.2
[5] 창태우 외 2인, "인트라넷 기반 전자문서교환 시스템에 관한 연구", 한국경영과학회/대한산업공학회 '97 춘계공동학술대회 논문집.
[6] 한국전산원, 공문서 전자 유통 방안, 한국전산원, 1996.
[7] 홍승필 외 1, "정보보안 기술과 구현", 파워북, 1998.5.
[8] Benoit Marchal, "Applied XML Solutions", SAMS, 2000.
[8] Frank Boumphrey 외 11인, "XML APPLICATIOS", 정보문화사, 1999.
[9] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, "XML and Java", 이한 출판사, 2000.8.
[10] Kathy & Mary, "The JFC Swing Tutorial", 정보문화사, 2000.1
[11] Patrick Naughton, Herbert Schildt, "The Complete Reference Java2", OSBORNE
[12] Richard Blair 외 12인, "Professional ASP XML", 정보문화사, 2000.1.
[13] Simon St. Laurent, XML : A PRIMER, IDG Books worldwide, Inc, 1998.
[14] NEC XML-Signature
http://www.sw.nec.co.jp/soft/xml_s/appform_e.html
[15] XML-Signature Syntax and Processing
http://www.w3.org/TR/2001/CR-xmldsig-core-20010419/

저 자 소 개

정경찬 : 명지대학교 산업공학과 졸업(2001)
현재 명지대학교 대학원 산업공학과 석사과정.
주요 관심분야는 ERP, SCM, CRM등이다.

박정선 : 서울대학교 산업공학과 졸업(1983)
KAIST 경영과학 석사(1985)
텍사스 주립대학(오스틴) MIS 박사(1993)
현재 명지대학고 산업시스템공학부 부교수
주요 관심분야는 전자상거래 응용/보안, 에이전트 개발, DB응용