XML 전자서명을 이용한 XML/EDI 시스템 보안에 관한 연구

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

이 경 록 *
Lee Kyong Rok
서 장 훈 *
Seo Jang Hoon
박 명 규 **
Park Myong Kyu

Abstract

As Internet spreads rapidly, the industrial structure is changing to a new paradigm. The previous EDI system was asked to change and WEB EDI, E-Mail EDI, FTP EDI etc. which are based on the internet appeared. These days the XML/EDI which has XML document appeared.

The XML/EDI consider advantages and disadvantages of VAN/EDI and EDI which based on the internet.

Also, EDI system has to assure a safe exchange between sender and receiver. But, the internet has security problems because it uses a open TCP/IP protocol. Although there are many methods for security, it is being studied with XML concept.

On this paper, we will suppose the XML/EDI system model with XML Signature, and build a procedure of electronic signature and delivery of document between sender and receiver.

* 명지대학교 산업공학과 박사과정
** 명지대학교 산업공학과 교수

1. 서 론

1.1 연구 목적

세계는 인터넷의 대중화로 인해 많은 자료들로 넘쳐나고 있다. 이에 따라 국가나 기업들의 업무 형태 또한 많은 변화를 겪고 있다. 따라서, 국가와 기업이 경쟁에서 살아남기 위해서는 정보의 효율적 관리와 공유, 그리고 가치 있는 정보로 가공하는 것이 필요하다. 이러한 환경에서 등장한 것이 EDI(Electronic Data Interchange)이다. 오늘날 많은 주문서, 송장과 같은 B2B(Business to Business) 문서들은 EDI 메시지들로서 교환된다.

EDI는 기업 간의 거래 데이터를 교환하기 위한 표준 포맷이며, 메모와 같은 자유로운 포맷의 문서에 대한 것이 아니라, 구매요청서, 송장, 납품지시서와 같은 구조화된 형식을 갖는 문서에 대한 것이다. 이러한, EDI가 인터넷의 보급이 확산됨에 따라 문서 전달을 위해 굳이 고비용의 사설망을 사용하기보다는 인터넷망을 기반으로 하려는 움직임이 시작되었고, 다양한 형태의 EDI들이 나타나게 되었다.

그러나, 사설망을 인터넷망으로 대체한다고 해서 기존의 EDI를 대신할 수는 없다. HTML로 표현된 인터넷 EDI는 문서의 구조를 정의할 수 없고, HTML 파일 자체만으로는 문서로서의 의미를 가질 수 없다고 할 수 있다.

또한, 인터넷망을 사용함으로 인해 보안상의 문제점이 생긴다.

인터넷은 TCP/IP라는 개방적인 프로토콜을 사용함으로 인해 사설망에서의 네트워크 구조보다 보안상의 위험에 많은 부분 노출되어지게 된다. 반면, VAN/EDI는 가입자 이외에는 네트워크에 대한 접근이 힘들고, 폐쇄적이며, 상대적으로 문서 전달과정에서 노출될 위험은 적다.

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.

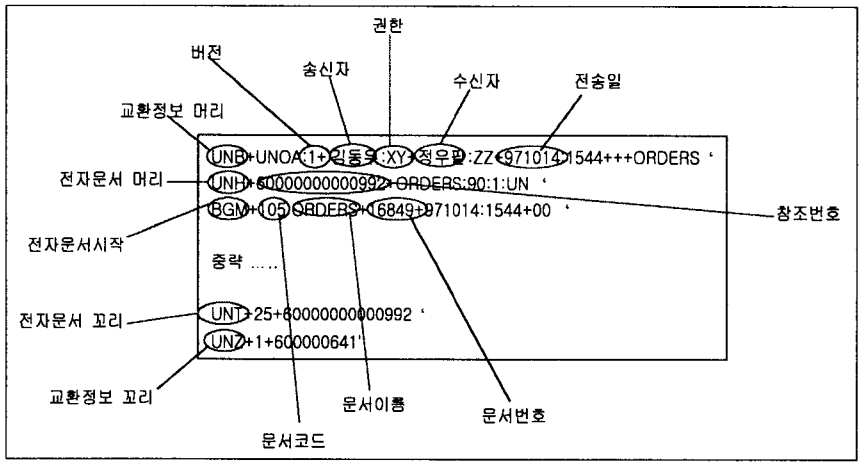
본 논문은 이러한 사회적 변화에 대응할 수 있는 XML/EDI 시스템과 이 시스템에 XML-Signature를 사용하여 얻을 수 있는 장점에 대하여 알아보고, 이를 활용하여 XML-Signature를 통해 서명을 거친 문서를 전달하는 과정을 구현하는 것이 연구목적이다.

1.2 연구 내용 및 방법

본 논문은 XML-Signature를 적용한 XML/EDI 시스템의 구현을 목적으로 한다. 이를 위해서 XML/EDI 시스템의 구조와 XML-Signature를 분석하고, XML을 EDI 시스템에 적용될 경우의 효과와 XML-Signature를 사용했을 경우의 효과에 대해 알아보고,

XML-Signature를 사용한 XML/EDI 문서전달 절차를 구현하는 방안을 제시하였다. 이 시스템을 구현하기 위하여 Windows 2000 Server환경에서 IIS(Internet Information Server)를 웹서버로 사용하였으며, JSP처리를 위한 Resin 2.0.4를 사용하였다.

또한, 서명과 XML-EDI 변환을 위하여 클라이언트 인터페이스로 자바애플릿을 사용하였으며, 서버측 프로세스를 위해 ASP와 JSP를 사용하였다. 또한, 시스템에 접근권한을 주어 허가되지 않은 사용자의 접근을 구분하였으며, 클라이언트측 파일관리를 위해 서명된 애플릿(Signed Applet)을 사용하였다.



[그림 1 EDI 문서의 예]

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

[표 1 시스템 구축 환경]

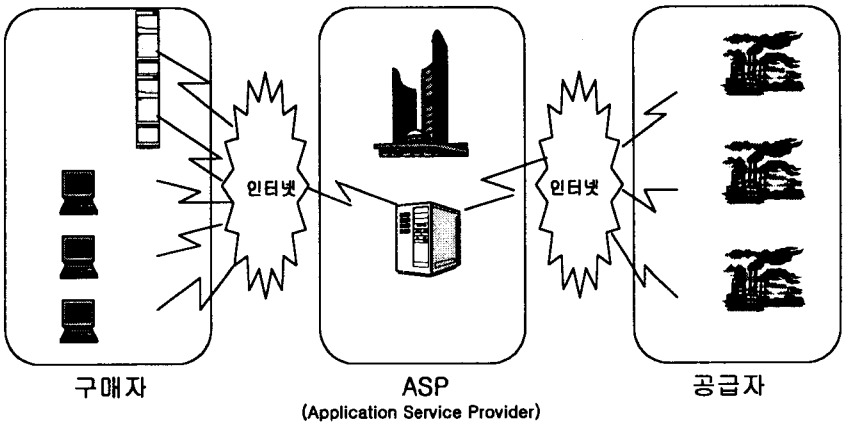
2. XML-Signature

XML Signature는 기존의 전자서명을 위한 알고리즘을 XML 문서에 적용한 전자서명 기법이다. XML Signature는 기존의 전자서명과 같이, 메시지를 다이제스트하고, 그

값에 개인키를 적용하여 서명 값을 생성하며, 수신자는 송신자의 공개키를 이용하여 메시지를 검증한다.
하지만, XML Signature는 Manifest라는 기능의 추가로 기존의 전자서명과는 조금 다른 형태를 띄게 된다. 기존의 전자서명을 사용하는 경우에는 문서 전체에 대하여 메시지 다이제스트를 생성하는 반면, XML 전자서명은 문서 내의 특정 부분만을 추출하여 메시지 다이제스트를 생성하므로 좀 더 효율적이라고 할 수 있다.
XML Signature는 XML 문서 내에 서명을 포함하는 것과 관계없이 어떤 형태의 데이터에 대해서도 무결성, 메시지 인증, 그리고 서명자 인증 서비스를 제공한다. 전자 서명은 데이터 무결성, 인증, 그리고 부인 봉쇄와 같은 정보보호 서비스를 제공한다.

3. XML-Signature를 적용한 XML/EDI 시스템

[그림 2]는 본 논문에서 구현하려고 하는 시스템의 시나리오를 보여준다. 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.



[그림 2 시스템 시나리오]

4. XML-Signature 처리 과정 및 구현 시스템의 기능

4.1 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 클라이언트 측 파일 관리를 할 수 있다. 클라이언트 측 파일관리 화면은 [그림 3]과 같다.

[그림 3 클라이언트 측 파일관리]

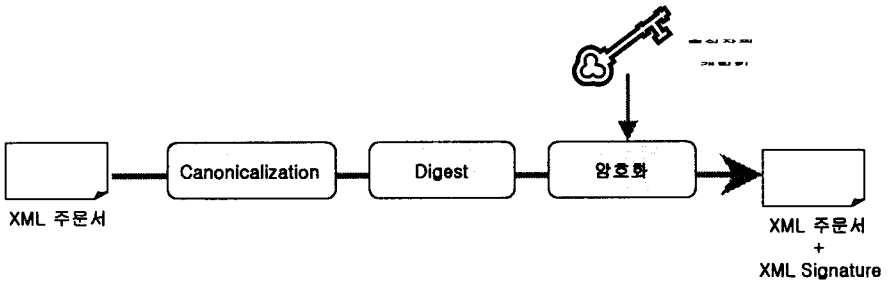
4.2 키 생성

송신자와 수신자간의 서명은 RSA 공개키 암호시스템에 의해 생성된 공개키와 비밀키로써 생성되고 검증된다. 송신자는 전달할 디지털 컨텐츠에 자신의 비밀키로서 서명을 생성하고, 수신자는 서명에 포함되어 있는 송신자의 공개키로 검증을 하게 된다.
회원 가입과 같이 시스템에 처음 접근하였을 경우, 클라이언트의 로컬디스크에 키 쌍(keypair)을 생성하여 keystore에 저장하게 되며, 이후 keystore에서 키를 가져와서 서명에 관련된 작업을 이루게 된다.

4.3 XML-Signature 생성과 검증

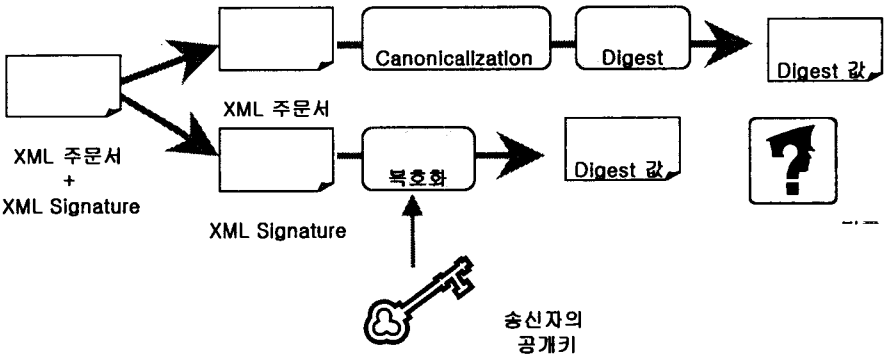
송신자로부터 수신자에게 문서를 전달하기 전까지 XML 전자서명을 위해 [그림 4]와 같은 3단계의 처리과정을 거치게 된다.

송신자에 의해 작성된 XML 문서는 정규화 과정을 통해 Canonical XML 문서로 변환한다. Canonical XML은 두 XML 문서간의 동일성을 문법적인 수준에서 확인하기 위한 것으로, 제공하는 원칙에 따라 XML 문서를 변환한다.
Canonicalization과정을 거친 문서는 해쉬함수를 통해 Digest과정을 거치고, 이는 송신자의 개인키로써 암호화 과정을 거쳐 최종적으로 서명을 생성하게 된다. 처음 작성된 XML 문서와 함께 디지털 서명은 수신자에게 전달되게 된다.



[그림 4 XML 문서의 송신과정]

서명 처리된 문서를 수신자에게 보내면 수신자는 문서에 포함된 XML-Signature에 의해 문서에 대한 검증을 하게 된다.
검증 과정은 [그림 5]와 같이 이루어지며, 크게 2가지 단계의 과정을 통해 얻어진 Digest 값을 비교한다.
우선, 첫 번째 단계는 수신된 문서에서 송신자가 작성한 컨텐츠를 추출하여 송신자가 Digest 값을 만들었던 것과 같은 과정을 거쳐서 Digest 값을 얻는다.
두 번째는 수신된 문서에서 XML-Signature 엘리먼트를 추출하여 암호화된 서명값을 KeyInfo 엘리먼트에 포함되어 있는 송신자의 공개키로써 복호화하여 Digest 값을 얻는다.



[그림 5 수신자의 검증과정]

첫 번째와 두 번째 단계의 결과값으로 얻어낸 Digest 값을 비교하여 일치하게 되면 문서의 변조 여부와 송신자에 대한 인증을 할 수 있다.

1) 송신자의 Signature 생성 기능

서명의 생성은 NEC의 API를 사용하여 XML-Signature를 [그림 6]과 같이 생성하였으며, Enveloped Signature 방식을 사용하여 XML 문서 내에 Signature 엘리먼트를 삽입시키도록 하였다.

Signature 엘리먼트 생성과정은 세 가지 단계로 이루어 진다.

① 비어있는 Signature 엘리먼트를 문서 안에 삽입한다.

```
<Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#" />
```

② 삽입된 Signature 엘리먼트 안에 서명을 위한 Method들을 정의한다.

- CanonicalizationMethod
- DigestMethod : SHA1 알고리즘을 지원한다.
- SignatureMethod : DSAwithSHA1과 RSAwithSHA1 알고리즘을 지원한다.

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 서명된 문서를 수신했을 경우 이 키 값으로 서명을 검증할 수 있다.
- SignatureValue : Digest를 송신자의 개인키로 암호화한 최종 서명 값이다.

[그림 6 XML-Signature 생성]

[그림 7]의 Signature 엘리먼트를 보면 SignatureMethod에 서명을 위해 RSAwithSHA1 알고리즘을 사용하고, 다이제스트를 위해 SHA1알고리즘을 사용하고, 다이제스트 값과 서명의 결과값 들이 명시되어 있음을 알 수 있다.

```
<Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo xmlns="">
    <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>OJXdPA5XPfaP4I99GIp2N3+zgOM=</DigestValue>
    </Reference>
  </SignedInfo>

  <SignatureValue xmlns="">ImMiw4Dj5aUvO8XQ60P6DxwmWdBOc6vQ7U1xDl0u6jIKgXpS598
   fWxLpukKlL+u+tQy9X6MKfug0SJlDvxviR63GTXVCLTo6IIM8qJq+usC+tjXKnIyh0tZvsMC
   hK6YMYOpeXSqre8JALoZjJIjeLyucWACILkCqChJ/m1RXCLk=</SignatureValue>
  <KeyInfo xmlns="">
    <KeyValue>
      <RSAKeyValue>
        <Modulus>AN+Wc0+iLsleJmHKp0NhTDWe/2O39suAUpFWqx3frf0Qlpd1O8ncTDllOQYgh6
          adFvCVoN/g/8qxZXuQNIAiqKtKYBMoE0tDPKsjDdxiOEf6P/CldcO9D6YHDLdj1aTdb5o4
          2wI2nbWXcY7sQpkVBS8bnJ4X4AxQtgLu+zKCVKZh</Modulus>
        <Exponent>AQAB</Exponent>
      </RSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

[그림 7 XML-Signature 예]

```
<Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  -- 중략 ---
  <Object xmlns="">
    <Manifest Id="MyFirst">
      <Reference Id="REF_01" URI="#ida">
        <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>2kTO7iz8vwRdTZgiUPy2IXrcq8k=</DigestValue>
      </Reference>
      <Reference Id="REF_02" URI="#idb">
        <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>FzFcDhvb2y+Hxebig03xy296JXI=</DigestValue>
      </Reference>
    </Manifest>
  </Object>
</Signature>
```

[그림 8 XML-Signature의 Manifest 엘리먼트]

[그림 8]은 서명될 내용을 문서의 일부나 여러 문서로 할 경우에 사용되는 Manifest 엘리먼트를 사용한 예이다. Reference 엘리먼트의 URI 속성을 통해 서명에 필요한 컨텐츠를 지시하고, 다이제스트 값을 포함하게 된다. 여러 컨텐츠에 대해 하나의 서명만을 사용함으로써 더 효율적인 서명을 얻어낼 수 있다.

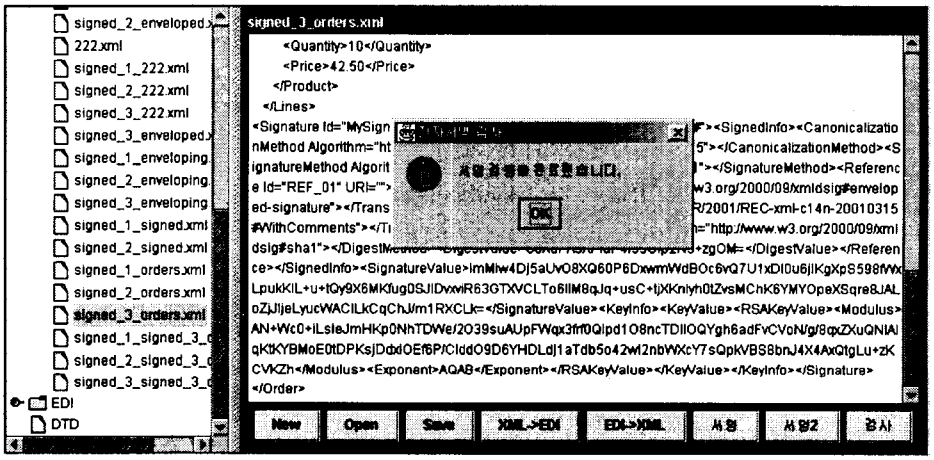
2) 수신자의 XML-Signature에 대한 검증 기능

송신자로부터 받은 XML 문서는 첨부되어 있는 송신자의 서명을 통해 전달과정에서 일어날 수 있는 문서의 변조와 정당한 송신자 여부에 대해 확인할 수 있다.
XML Signature가 포함된 문서는 다음 세 가지에 대해서 만족할 때 서명이 유효하다.

① 서명 검증(Signature Validation)
SignatureValue가 CanonalizationMethod와 SignatureMethod를 내포하는 SignedInfo를 처리하여 얻은 결과 값과 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.

본 연구에서는 서명 검증과 참조 검증에 대한 검증을 처리하였다.



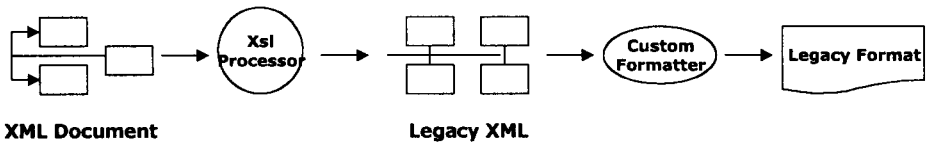
[그림 9 XML-Signature를 통한 문서의 무결성 검증]

4.4 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.

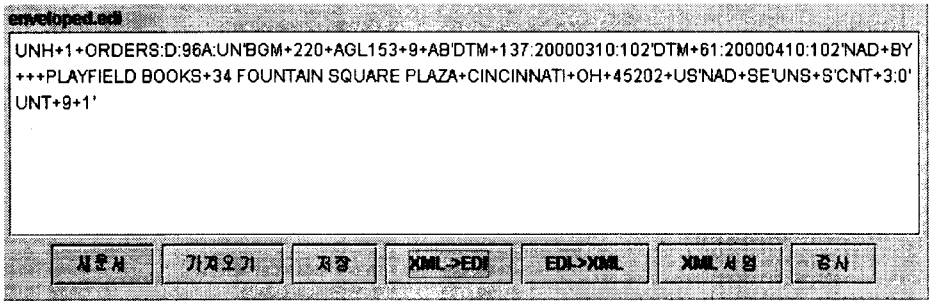
① XML 문서를 읽어 들인다.
② 두 구조사이에서의 변환. XML요소를 EDI segment로 변환한다.
③ EDI Syntax에 따른 EDI 문서를 작성한다.

XML 파서는 XML 문서를 읽어들이는 것을 지원하며, XSLT 프로세서는 구조사이의 변환을 지원한다. 자바는 레거시 XML 문서를 EDI로 변환한다. XML 문서가 변화되는 과정과 변화된 상황을 [그림 10], [그림 11]과 같다.



Completing XSL with a custom formatter.

[그림 10 XML 문서가 변환되는 과정]



[그림 11 XML 문서에서 EDI 문서로 변환]

5. 결론 및 향후 연구과제

본 연구에서는 XML/EDI 시스템에서의 인증과 무결성을 위하여 XML-Signature를 사용하여 구현하였다.

EDI는 정보를 단순히 전달하는 단계에서 벗어나서 전체 시스템의 정보 통합의 중요한 수단이며 이에 따라 EDI의 활용은 광범위하게 발전하고 있다. 인터넷의 발전으로 EDI 시스템에도 변화를 위한 노력이 이루어지고 있으며, 그에 따라 등장하는 문제점들에 대한 해결책들도 다양하게 연구되고 있다. EDI에 XML을 도입함으로써 기존 EDI의 많은 문제점들을 해결할 수 있게 되었으며, 보안이라는 난제에 XML에 적합한 XML-Signature를 적용함으로써 보다 더 효과적인 보안 대응책이 되었다.

본 연구는 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.

6. 참고문헌

[1] 김철(1996), 암호학의 이해, *영풍문고*
[2] 김형도(2000), B2B 전자상거래 @XML, *배움터*
[3] 박창섭(2001), 암호이론과 보안, *대영사*
[4] 이종호(2001), XML과 전자상거래, 정보문화사
[5] 창태우 외 2인(1997), 인트라넷 기반 전자문서교환 시스템에 관한 연구, 한국경영과학회/대한산업공학회, *'97 춘계공동학술대회 논문집*
[6] 한국전산원(1996), 공문서 전자 유통 방안
[7] 홍승필 외 1(1998), 정보보안 기술과 구현, *파워북*
[8] Benoit Marchal(2000), Applied XML Solutions, *SAMS*
[8] Frank Boumphrey 외 11인(1999), XML APPLICATIOS, *정보문화사*
[9] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto(2000), XML and Java, *이한 출판사*
[10] Kathy & Mary(2000), The JFC Swing Tutorial, *정보문화사*
[11] Patrick Naughton, Herbert Schildt(1999), The Complete Reference Java2, *OSBORNE*
[12] Richard Blair 외 12인(2000), Professional ASP XML, *정보문화사*
[13] NEC Corporation(2001), XML-Signature, http://www.sw.nec.co.jp/soft/xml_s/appform_e.html

저 자 소 개

이경록 : 한양대학교 공학사, 명지대학교 산업공학 석사, 현재 명지대학교 산업공학 박사과정, 효성물산 대표이사
관심분야는 품질공학, e-business, SCM, CRM

서장훈 : 명지대학교 산업공학과를 졸업하고, 동 대학원 산업공학과 석사를 취득하였으며, 현재는 산업공학 박사과정에 있으면서, 명지대 리서치 파크 전임 연구원으로 재직중이다. 주요 관심분야는 e-Business, ERP, 품질공학, Data-Mining.

박명규 : 한양대학교 산업공학과 졸업. 미국 일리노이 공대에서 산업공학 석사, 건국대학교 대학원 산업공학과에서 박사학위를 취득하였으며, 현재 명지대학교 산업공학과 교수로 재직중이다. 주요 관심분야는 TQM, QE, METHODS ENG, 재고 물류관리, 확률모형, FORECASTING, 시스템분석 등이다.