논문 2008-45CI-6-18

XML 보안 기반의 부동산 계약서 전자서명 생성 및 검증

(Generation and Verification of a Real Estate Contract Digital Signature Based on XML Security)

이 문 구*

(Moon-Goo Lee)

요 약

전자거래에 있어서 데이터의 무결성과 부인방지 같은 보안 서비스는 안전한 전자거래를 가능하게 하는 필수요소이다. 본 논문에서는 기존 부동산 거래 절차를 기반으로 안전한 부동산 전자상거래를 가능하게 하는 부동산 계약 전자서명 생성 및 검증 시스템을 구현하였다. 본 논문에 사용된 보안 서비스를 위한 기술적 배경은 XML 전자서명 기술로 XML 전자서명은 기존에 존재하는 전자서명 알고리즘에 XML을 적용한 전자서명 기법으로 기존 서명과는 다르게 부분 데이터에 대한 서명이 가능하기 때문에 매우 효율적이며 현재 널리 이용되고 있는 XML 기반 전자 상거래 시스템에 적용이 쉽다는 장점이 있다.

Abstract

Talking about reliability of E-commerce, the security services such as data integrity and non-repudiation are the most crucial elements. This thesis implemented the real estate contract digital signature system that makes this real estate E-commerce possible. The technical background used in this thesis for the security services is XML (eXtensible Markup Language) signature technique, which is a signature technique that applies XML on the existing digital signature algorithm. The advantage of using XML signature technique is that it is very efficient since signing for the partial data is possible, and it is easy to apply to the XML-based E-commerce system, which is most commonly used.

Keywords : : Availability, Integrity, Confidentiality, Non-repudiation, Message-digest

I. 서 론

안전한 부동산 전자상거래 시스템의 구현을 위해서는 가용성을 기반으로 비밀성, 무결성과 같은 정보보호의 3대 목표는 물론, 사용자의 정보보호와 거래 행위의 부인 방지(non-repudiation), 거래 당사자들의 신원확인 등과 관련한 보안 문제를 고려하여야 한다. 이러한 보안 문제를 해결하기 위하여 본 논문에서는 XML 보안을 기반으로 부동산 계약서 전자서명 생성 및 검증 시스템을 구현하였다. XML은 문서를 구조화하여 체계적으로 저장하고 관리할 때, 플랫폼에 독립적이면서 문서에 대한 정보의 전송과 교환이 편리한 언어로 전자상거래 활성화 및 인터넷 전자결제, 전자계약 등에 표준화된 방식이다[12].

그러므로 본 논문에서 제안하는 XML 보안 기반의 부동산 계약서 전자서명 생성 및 검증 시스템은 부동산 정보의 기밀성과 무결성 그리고 부동산 거래 상호 인증 기능이 제공된다. XML전자서명 방식은 다양한 형태의 디지털 콘텐츠 서명방식의 응용에 적합하다는 특징과 XML 기반의 응용시스템 통합이 용이하다는 장점을 갖는다. 본 논문에서 XML 문서의 키 관리는 X.509 인증 표준을 기반으로 설계 및 구현하였으며, RSA 암호 알고리즘과 SHA1 해쉬 함수 그리고 SunOS 5.6 환경에서

* 평생회원, 김포대학 IT 학부 인터넷정보과 교수
(Dept. of Internet Information, Division of IT, Kimpo College)

※ 이 논문은 2008학년도 김포대학의 연구비 지원에 의하여 연구되었음
접수일자: 2008년9월25일, 수정완료일: 2008년10월23일

JSP(jdk 1.3.1)로 구현하였다.

본 논문의 구성은 제 1장 서론에서 연구의 배경과 목적을 기술하였으며, 제 Ⅱ장은 XML 기반의 전자서명 보안기술에 대하여 서술하였다. 제 Ⅲ장은 본 논문에서 제안하는 XML 보안 기반의 부동산 계약서 전자서명 생성 및 검증내용과 소스코드를 기술하였다. 마지막으로 제 Ⅳ장에서는 결론 및 향후 연구방향 등에 대하여 기술하였다.

Ⅱ. XML기반의 보안 기술

1. XML 전자서명 생성과 검증

XML 기반의 전자서명은 그림 1과 같이 송신 문서에 해쉬(hash)함수 알고리즘(SHA-1)을 적용하여 문서의 해쉬값 즉, 메시지 축약값(message digest)을 산출한다. 그리고 메시지 축약값(message digest)에 공개키 암호 알고리즘(RSA)을 이용하여 암호화하는데, 이때 암호화 키 값을 서명자의 개인키(private key)로 암호화하여 전자서명 값(Signature Value)과 문서 원본(Document D)을 수신자에게 전송한다[9].

이러한 XML 전자서명 생성과정은 XML문서에서 참조(Reference)생성과 서명(Signature)생성으로 나누어서 진행된다. 참조생성은 참조 요소 각각의 압축 값이 어떻게 계산될 지를 정의하는 것으로 서명 객체에 변환(Transform)들을 적용한 뒤 결과 객체에 대해 압축값을 계산하고 참조요소를 생성한다.

반면에, 서명 생성과정에서는 실제 서명값을 생성 하므로 <SignatureMethod>, <CanonicalizationMethod>, <Reference(s)>등을 포함하는 <SignedInfo> 요소를 생성하고 <SignedInfo> 안에 지정된 알고리즘을 사용하여 <SignedInfo>에 정규화를 적용한 후 <SignatureValue>를 계산하고, 마지막으로<SignedInfo>, <Object(s)>, <KeyInfo>, <SignatureValue>등을 포함하는 서명요소를 생성한다.

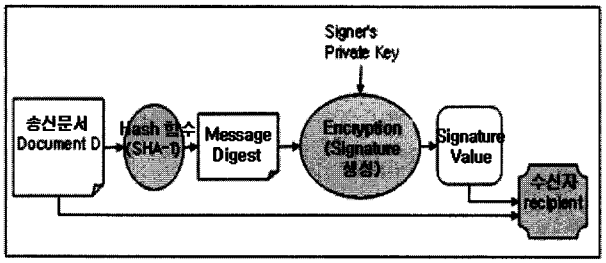

그림 1. XML 서명 생성과정
Fig. 1. XML Signature Generation Process.

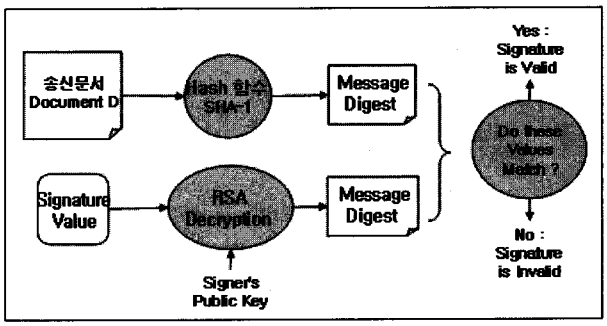

그림 2. XML 서명 검증과정
Fig. 2. XML Signature Verification Process.

XML 서명문과 문서가 수신측에 전송되면 수신측에서는 원본문서는 물론 XML 서명문서를 SAX(Simple API for XML)나 DOM(Document Object Model) 파서를 이용하여 해독한 뒤 그림 2와 같이 검증과정을 수행한다. 수신자는 문서원본(Document D)을 송신자와 같은 해쉬 알고리즘(SHA-1)을 적용하여 문서의 메시지 축약값(message digest)을 산출한다. 송신자가 보낸 암호화된 전자서명 값(Signature Value)을 서명자(송신자)의 공개키로 복호화하여 메시지 축약값을 산출한 후 두 메시지 축약값이 일치하면 송신된 전자서명은 유효하다는 것이 검증된다.

2. XML 전자서명 문법

XML 서명을 위한 요소의 구성은 그림 3과 같다.

```
<Signature ID?>
    <SignedInfo>
      <CanonicalizationMethod/>
      <SignatureMethod/>
      (<Reference URI?>
        (<Transforms>)?
        <DigestMethod>
        <DigestValue>
      </Reference>)+
    </SignedInfo>
    <SignatureValue>
    (<KeyInfo>)?
    (<Object ID?>)*
</Signature>
```

서명 문법 ? = zero or one occurrence
+ = one or more occurrences
* = zero or more occurrences

그림 3. XML 서명을 위한 요소의 구성
Fig. 3. Configuration of Element for XML Signature.

(1) <Signature>요소 : XML 서명문서의 루트 요소로서 <SignedInfo>, <SignatureValue>, <KeyInfo>,

<Object>요소 등으로 구성된다[3].

(2) <SignedInfo> 소 : <Canonicalization> 알고리즘, Signature 알고리즘, 하나 또는 그이상의 Reference를 구성요소로 한다. 또한 다른 서명과 객체에 의해 참조되는 것을 허락하는 선택적인 ID 속성을 포함한다.

(3) <CanonicalizationMethod>요소 : 서명값을 수행하기 전 XML 문서를 정규화하기 위해 요구되는 알고리즘으로, <Canonicalization>알고리즘에는 현재 minimal, Canonical XML with Comments, Canonical XML(omits comments) 세 가지 알고리즘이 사용된다.

(4) <SignatureMethod>요소 : 서명값을 발생하기 위해 사용되는 알고리즘은 SHA1기반의 DSA와 SHA1 기반의 RSA 등이 있다.

(5) <Reference>요소 : 서명될 데이터에 대해서 기술한다. <Reference>요소는 선택 적이며 ID를 통해서 다른 곳에서 참조할 수 있다. 또한 메시지 다이제스트 알고리즘, 메시지 다이제스트 값, <Transforms>요소를 포함 할 수 있다. URI 속성들은 URI Reference를 사용하는 데이터 객체를 정의하며 XML 전자서명 애플리케이션은 반드시 URI 구문(syntax)을 파싱한다[13].

(6) <Transforms> 요소 : 서명되기 전의 데이터 처리를 지시하는 강력한 메커니즘이다. 예를 들어 데이터의 일부분에만 서명을 하고 싶다면 데이터의 서브셋을 추출해야 할 것이다[13].

(7) <DigestMethod>요소 : 서명된 데이터에 적용될 축약 알고리즘을 확인하는 요소이다.

(8) <DigestValue>요소 : <DigestMethod>요소로부터 축약 출력된 인코딩 값을 포함하는 <Reference>요소 내의 필수적인 요소로 축약은 항상 Base64로 인코딩된다[13].

(9) <Manifest>요소 : Manifest는 서명할 자원의 집합을 말하는 것으로 서명할 데이터를 바로 가리킬 수도 있고, URI로 접근할 수 있는 웹 자원일 수도 있는데 즉 Manifest는 서명에 포함될 리소스의 리스트나 모음을 가리킨다[13].

(10) <SignatureValue>요소 : 디지털 서명의 실제적인 값을 포함하는데 Signature Method 요소에 정의된 알고리즘을 선택하여 생성된 값을 표현한다.

(11) <KeyInfo>요소: 식별자가 적절한 확인키를 찾는 작업을 원활하게 하기 위한 정보를 제공하는데 KeyInfo는 키 발생기를 통해 생성되는 키에 대한 정보를 입력 할 수 있다. <KeyInfo>요소는 선택적이기 때문에 사용자가 선택적으로 이용 할 수 있다. 또한 키 이름, 인증서 및 다른 공유키 관리정보를 포함한다[3].

(12) <Object>요소: Object 요소는 데이터를 Signature 요소의 안쪽과 SignedInfo 요소의 바깥쪽에 둘 수 있도록 한다. XML 암호화는 XML 문서를 암호화 알고리즘을 이용하여 암호화한 형태로 송,수신 함으로써 보안 수준을 향상시킬 수 있는데 XML 암호화는 문서 전체가 아니라 XML 문서내의 필요한 부분만 암호화하므로 암호화 속도가 빠르다는 장점이 있다[13].

III. XML기반의 부동산 계약서 전자서명

1. XML 전자서명 키 관리

XML 키 관리 설계(XML key management specification)는 X.509 인증서를 기반으로 하며, <KeyInfo>의 요소(element)는 키의 이름을 위한 단순한 문서 식별자인 <KeyName>, RSA나 DSA 공개키 값 인 <KeyValue>, 키 정보의 원격 참조를 위한 허용 <RetrievalMethod>, 그리고 X.509 인증서 기반의 관련 자료를 위한 <X509Data>, PGP 관련키와 식별자 <PGPData>, 간단한 공개키 기반 데이터 관련정보인 <SPKIData>, 키 동의 요소를 위한<MgmeDATA>등으로 구성된다.

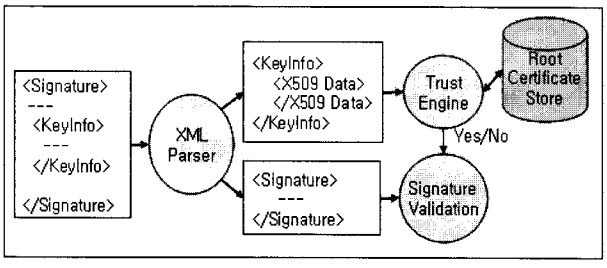

그림 4. XML 키 관리 설계
Fig. 4. Design of XML key management.

2. XML기반의 부동산 계약서 전자서명

XML 서명이 이루어지는 부분은 부동산 중개자로부터 계약서 내용의 무결성을 유지하기 위해서 부동산 중개자의 비밀키를 이용하여 전자서명을 하는 부분이다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<xsl:schema xmlns:xsl="http://www.w3.org
/2001/XMLSchema"elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xsl:element name="계약서종류">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="계약서" maxOccurs="unbounded">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="부동산의표시">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="소재지">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="시도" type="xsl:string"/>
<xsl:element name="구" type="xsl:string"/>
<xsl:element name="동" type="xsl:string"/>
<xsl:element name="세부주소" type="xsl:string"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="토지지목" type="xsl:string"/>
<xsl:element name="토지면적" type="xsl:float"/>
<xsl:element name="건물용도" type="xsl:string"/>
<xsl:element name="건물면적" type="xsl:float"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="계약내용">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="매매대금" type="xsl:int"/>
<xsl:element name="계약금" type="xsl:int"/>
<xsl:element name="중도금" type="xsl:int"/>
<xsl:element name="잔금" type="xsl:int"/>
<xsl:element name="계약날짜" type="xsl:date"/>
<xsl:element name="중도금날짜" type="xsl:date"/>
<xsl:element name="잔금날짜" type="xsl:date"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="매도인정보">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="매도인주소" type="xsl:string"/>
<xsl:element name="매도인주민번호" type="xsl:string"/>
<xsl:element name="매도인성명" type="xsl:string"/>
<xsl:element name="매도인전화" type="xsl:string"/>
<xsl:element name="매도인휴대폰" type="xsl:string"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="매수인정보">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="매수인주소" type="xsl:string"/>
<xsl:element name="매수인주민번호" type="xsl:string"/>
<xsl:element name="매수인성명" type="xsl:string"/>
<xsl:element name="매수인전화" type="xsl:string"/>
<xsl:element name="매수인휴대폰" type="xsl:string"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="중개업자정보">
<xsl:complexType>
<xsl:sequence>
```

그림 5. 부동산 계약서 스키마 파일
Fig. 5. Real Estate Contract Schema file.

따라서 계약내용 중에서 부동산 소재지, 계약내용, 매도자, 매수자의 주민번호를 추출하여 압축하고 압축값에 부동산 중개자의 비밀키를 이용하여 암호화 한다. XML 서명은 문서 작성자에 대한 정보를 전자서명 방식을 이용하여 XML 문서에 첨부한 형태로 변환시켜 교환하는 방식이다. XML 서명은 기존에 존재하는 전자서명 알고리즘에 XML을 적용한 서명 기법으로 기존 전자서명과 같이 메시지를 축약(digest)하고, 그 값에 개인키(private key)를 적용하여 서명값을 생성하며, 수신자는 송신자의 공개키(public key)를 이용하여 메시지를 검증(verification)한다. 원시(source) 전자 서명을 검증하기 위해서 서명자는 반드시 검증 방법에 대한 정보를 추가해야 한다. 검증 방법에 관한 정보는 수신자에 대한 정보와 검증키뿐만 아니라 사용된 알고리즘에 대한 정보도 포함되어야 한다.

XML이나 평문으로 이루어진 데이터에 서명을 할 경우 텍스트가 하나의 애플리케이션에서 다른 애플리케이션으로 전송될 때 실제 의미는 변화가 없지만 라인 종료문자가 바뀐다거나, 공백이 추가 또는 삭제되어지는 경우처럼 2진의 옥텟 값을 다르게 할 수 있는 여러 가지 문제가 있다. 이런 경우 서명의 확인은 사실상 불가능해진다. 이러한 문제를 해결하기 위해서 텍스트의 규

```
<xsl:element name="사무소소재지" type="xsl:string"/>
<xsl:element name="사무소명칭" type="xsl:string"/>
<xsl:element name="등록번호" type="xsl:string"/>
<xsl:element name="중개업자전화" type="xsl:string"/>
<xsl:element name="중개업자휴대폰" type="xsl:string"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="계좌정보">
<xsl:complexType>
<xsl:sequence>
<xsl:element name="매도자계좌번호"/>
<xsl:element name="매도자은행코드"/>
<xsl:element name="매수자계좌번호"/>
<xsl:element name="매수자은행코드"/>
<xsl:element name="중개업자계좌번호"/>
<xsl:element name="중개업자은행코드"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
<xsl:element name="기타사항"/>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
</xsl:sequence>
</xsl:complexType>
</xsl:element>
</xsl:schema>
```

그림 6. 부동산 계약서 스키마 파일(계속)
Fig. 6. Real Estate Contract Schema file (continue).

격화를 사용하는데 데이터를 적절하게 규격화(canonicalization)하는 것은 견고함과 보안이 강화된 서명을 위해서 필수적이다. 제안하는 부동산 거래 계약서 전자서명 생성 및 검증 시스템의 대상문서 구문 정의를 위해서 스키마를 그림 5와 그림 6으로 설계하였다[2].

XML 스키마 설계는 XML이 허용하는 구문을 정의하는 방법을 설계하는 것으로 압축과 전자서명을 위한 시스템 구현은 자바의 보안라이브러리를 이용하였다.

그림 7은 부동산 계약을 위한 부동산 소재지, 계약내용, 매도자 및 매수자의 개인정보 등에 대한 XML 서명을 해쉬함수(SHA-1)로 압축하는데, 그 일부분인 부동산소재지에 대한 축약값(digest value) 산출을 위한 소스코드이다. MessageDigest md =Message Digest. get Instance("SHA1")은 SHA1 알고리즘을 이용하여 메시지 다이제스트 객체를 생성하며 md라는 새로운 객체를 선언하는 것이다.

메시지 다이제스트를 얻기 위해서는 byte[] raw= md.digest()를 이용하는데 이러한 과정을 수행하고 RSA 알고리즘과 개인키(private key)를 이용하여 전자서명값을 생성한다. 전자서명을 위해서는 전자서명을 위한 객체 생성을 해야 하는데 객체를 생성할 때는 Signaturesig=Signature.getInstance (SHA1 With RSA)를 이용하여 전자서명 객체를 생성한 뒤 byte[] signatureBytes=sig.sign()을 이용하여 전자서명을 한다.

```
import java.io.*;
import java.security.*;
import sun.misc.*;
public class SHA1_test{
   public static void main(String[] args) throws
Exception{
MessageDigest md =
 MessageDigest.getInstance("SHA1");String strVar
= "서울시 영등포구 여의도동 여의도아파트 100동 500
호";
byte[] data = strVar.getBytes("UTF8");
          md.update(data );
          byte[] raw = md.digest();
   BASE64Encoder encoder = new
BASE64Encoder();
   String base64 = encoder.encode(raw) ;
   System.out.println("SHA1===="+base64);
   }
}
```

그림 7. 자바를 이용한 축약값 소스코드
Fig. 7. DigestValue source code using JAVA

전자서명은 XML 문서에 동봉한(enveloping)[2]서명 방법을 이용하여 그림 8과 같이 XML 전자서명 문서를 생성하였다. XML 서명 소스코드에서 SignatureMethod는 RSAwithSHA1이 사용되었고 Reference 에서는 Id 속성을 이용하여 압축될 문서의 위치를 알려준다.

즉 object 태그의 부동산 소재지가 압축대상이 된다. DigestMethod에서는 SHA1을 사용하였고, DigestValue는 지정된 문서의 압축값을 base64로 인코딩한 결과를 저장하고 SignatureValue는 압축값을 RSA를 이용하여 암호화한 결과를 base64로 인코딩한 결과가 저장된다.

XML 전자서명을 검증하는 과정은 매도자와 매수자가 사전 협의된 거래조건을 확인하는 과정에서 부동산 중개자가 서명한 계약 내용을 부동산 중개자의 공개키(public key)를 이용하여 검증할 때 발생되고, 매도자 및 매수자가 계약 동의를 하는 과정을 부동산 중개자가 확인할 때 발생된다. XML 전자서명을 검증하는 과정은 <Reference> 검증과 <Signature> 검증과정으로 나누어서 검증을 하는데 먼저 <Reference> 검증과정에서는 <SignedInfo> 요소를 정규화하고 해쉬 값을 계산하는 대상 객체를 추출하고 해쉬 값을 계산한다. 마지막으로 <SignedInfo> <Reference> 안의 <DigestValue>와 생

```
<Signature>
 <SignedInfo>
   <SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="#address">
      <DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
       <DigestValue>
        e67xLZ6gU6OiG1CF5UQ6UDafiYU=.
       </DigestValue>
    </Reference>
    </SignaedInfo>
   <SignatureValue>
CE22wR2kDGsQIm6i4w0rH7M9ZGLM3YmqERac1XbHNW
h2uZbfRStzrKr3yfJp+cR1IXm5ryHfKc+A9HgprLcu6c2/5wp
F3kFuKOC9TRPu2bE+hMEcW3Ye7pSQWGvq58T72AHUc
Ody9TgbLE9yyCXzEgUdQ2bqRkK0wwfFi+/Lsp
  </SignatureValue>
  <Object>
  <소재지 Id="address">
  서울시 영등포구 여의도동 여의도아파트 100동 500호
    </소재지>
   </Object>
   </Signature>
```

그림 8 XML 전자서명 생성
Fig. 8. Generation of XML Digital Signature.

```
<Signature Id="Purchase Order=1">
 <SignedInfo>
<SignatureMethodAlgorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1"/>
   <Reference URI="#address">
     <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <KeyInfo>
  <RetrievalMethod
Type="http://www.w3.org/2000/09/xmldsig#X509Data"
URI="http://www.myserver.com/certchain.xml"/>
  <X509Data>
    <X509 Certificate> certificates </X509 Certificate>
     <X509 Certificate> names </X509 Certificate>
   <X509 Certificate> related data </X509 Certificate>
  </X509Data>
   <KeyValue>
   <RSA KeyValue>
<Modulus>uuoRfdCnnx1pnV33LzOehledVLo9EoCZ9VZk
+Hi5CDAkyy1wOIUxEl4bsR21v0CGr21T6Itwvo+Wt9w==
     </Modulus>
    <Exponent>EQ==</Exponent>
   </RSA KeyValue>
    </KeyValue>
</KeyInfo>
<DigestValue>e67xLZ6gU6OiG1CF5UQ6UDafiYU=.
</DigestValue>
   </Reference>
   </SignaedInfo>
   <SignatureValue>
CE22wR2kDGsQIm6i4w0rH7M9ZGLM3YmqERac1XbHN
Wh2uZbfRStzrKr3yfJp+cR1IXm5ryHfKc+A9HgprLcu6c2/5
wpF3kFuKOC9TRPu2bE+hMEcW3Ye7pSQWGvq58T72A
HUcOdy9TgbLE9yyCXzEgUdQ2bqRkK0wwfFi+/LspQ
   </SignatureValue>
   <Object>
     <소재지 Id="address">
    서울시 영등포구 여의도동 여의도아파트 100동 500호
     </소재지>
   </Object>
   </Signature>
<SecureDoc>
 <EncryptedKey   Id="부동산 중개자의 비밀키"
                Recipient="매도자 및 매수자">
<EncryptionMethodAlgorithm="http://www.w3.org/2001/0
4/xmlenc#rsa-1">
  <CipherData>
   <CipherValue>
mPCadVfOHnkdjHmdOOVk000dsdC'HkLiOPhAqPWjFdW
+HnnMfz5H4hjKlWr8nVxZmvv6Sm49FgZpanvv6VwBhjQ
Ryu77IFFBccHkLmSeONg12Yo
   <CipherValue>
 </CipherData>
</SecureDoc>
```

그림 9. XML 전자서명 검증
Fig. 9. Verification of XML Digital Signature

성된 다이제스트 값을 비교하는데 일치하지 않으면 검증 실패로 간주한다. <Signature> 검증과정에서는 <KeyInfo> 또는 외부소스로부터 검증키 정보를 얻고 <CanonicalizationMethod>를 사용하여 <Signature Method>의 정규형을 구하고, <SignedInfo> 요소에 대해 <SignatureValue>를 검증한다. 여기서 주의할 것은 <Signature> 검증이나 <Reference> 검증의 경우 <DigestValue> 와 <SignatureValue>의 비교는 대수적으로 행해지거나 바이너리 옥텟열을 이용해야 한다. base64 인코딩의 텍스트를 비교하면 안 되는데 그것은 화이트스페이스 표현 관례의 불일치로 인해 실패하기 때문이다[13].

Ⅳ. 결 론

전자상거래에 필요한 정보보호기술로 현재 사용되고 있는 TLS(Transport Layer Security), SSL(Secure Sockets Layer) 또는 IPSEC(IP Security)은 채널을 통과하는 모든 정보에 대한 보안은 제공하지만 전달된 데이터가 저장될 때, 터널을 통과한 뒤 복호화 된 데이터에 대한 기밀성이 파괴되거나 무결성 및 인증 등의 보안서비스를 제공받지 못한다.

반면 XML 보안은 유연하기 때문에 많은 사용자들에게 관심이 되고 있으며 XML 암호화는 문서 전체가 아니라 XML 문서 내에 필요한 부분만 암호화하므로 암호화 속도가 빠르다는 장점이 있다[6]. 본 논문에서는 안전한 부동산 계약을 위하여 XML기반으로 부동산 거래 전자서명 생성 및 검증 시스템은 암호화를 위하여 XML 문서의 키 관리는 X.509 인증표준을 기반으로 설계 및 구현하였으며, RSA 암호 알고리즘과 SHA1 해쉬함수 그리고 SunOS 5.6 환경에서 JSP(jdk 1.3.1)로 구현하였다.

XML 데이터에 대해 전자서명을 위한 해쉬함수와 암호 알고리즘을 적용함으로써 XML을 기반으로 한 전자상거래 시스템에 신뢰할 수 있는 보안 서비스를 제공할 수 있으며 본 연구에서 사용하고 있는 XML 전자서명을 응용하여 웹에서 사용되고 있는 다양한 응용분야에 활용될 수 있도록 지속적인 연구를 진행하고자 한다.

참 고 문 헌

[1] 김주한 외, *XML 암호화 표준 동향*, 정보보호학회

지, 제11권 제4호, 2001.
[2] 김영진, 이문구 "*XML 보안 기반의 부동산 계약서 전자서명 생성 시스템 설계*", 석사학위논문, 이화여자대학교, 2003.
[3] 지석진, *XML 전자서명에 관한 연구*, 석사학위논문, 세종대학교, 2001.
[4] 안철범, 나현묵, *통합의료정보 시스템을 위한 XML DTD 설계 및 구현*, 전자공학회 논문지 제 40권 CI편 제 6호 2003. 11
[5] 이종호, *XML과 전자상거래*, 정보문화사, 2002.
[6] 허영백 옮김, *XML 보안:서명과 보안을 위한 새로운 구문*, 피어슨에듀케이션 코리아, 2002.
[7] D.Eastlake, *XML-Signature Syntax and Processing*, 2002,
[8] Donald E. Eastlake III and Kitty Niles, *Secure XML:The New Syntax for Signatures and Encryption*, Pearson Addison Wesley, 2002.
[9] Blake Dournaee, *XML security*, McGraw Hill, 2002.
[10] Brett McLaughlin, *Java & XML 2nd Edition*, O'REILLY, 2001.
[11] D.Eastlake, XML-Signature Syntax and Processing, 2002.
[12] Simson Garfinkel and Gene Spafford, *Web Security-Privacy & Commerce*, O'REILLY, 2002.
[13] http:\\www.i-sec.co.kr/school/secuinfo/secuinfo18.html

저 자 소 개

이 문 구(평생회원)
1984년 숭실대학교 전자계산학과 학사 졸업.
1993년 이화여자대학교 전산교육학 석사졸업.
2000년 숭실대학교 컴퓨터 시스템 공학박사 졸업.
2000년~현재 김포대학 IT학부 인터넷정보과 부교수
<주관심분야 : 네트워크 프로그래밍, 인터넷 보안, 시스템 보안, 암호화 알고리즘, 전자상거래 보안, 침입 탐지시스템, 침입 차단시스템>