사용자의 타임스탬프가 제공된 XML 서명기법

(XML Signature Schema with User's Timestamp)

이원진*, 전일수**
(Won-Jin Lee, Il-Soo Jeon)

요 약 XML에 대한 중요성이 높아짐에 따라, XML 보안에 관한 연구들이 활발히 이루어지고 있다. Apvrille와 Girier는 XML 서명에서 적시성을 제공하기 위해 time stamping protocol(TSP)이 제공된 XML 서명기법을 제안하였다. 그러나 이들의 기법에서 적시성의 신뢰성은 오직 TSA(Trusted Third Party)에 의존한다. 이러한 문제를 해결하기 위하여 분산모델과 연결모델을 통한 해결책을 제시하고 있지만 시스템의 부하가 커진다. 본 논문에서는 이러한 문제점을 해결할 수 있는 새로운 XML 서명기법을 제안한다. 제안한 서명기법은 사용자의 타임스탬프가 제공된 XML 서명기법으로서 적시성을 제공할 수 있고 TSA의 신뢰성 문제를 해결 할 수 있다.

핵심주제어 : XML. XML 보안, XML 서명, 타임스탬프

Abstract There are lots of XML security researches as growing importance of XML. Apvrille and Girier proposed a XML signature schema with a time stamping protocol(TSP) to support timeliness. However, the security of the timeliness in their schema depends on only a trusted security authority(TSA). To solve this problem, they suggested a solution by the distributed and linked model, but their solution has a big overhead. This paper proposes a new XML signature schema to solve the overhead problem. The proposed signature schema can offer the timeliness by the XML signature schema with user's timestmap and it solve the TSA's security problem.

Key Words : XML, XML Security, XML Signature, Timestamp

1. 서 론

2003년 공식 표준 언어로 XML(eXtensible Markup Language)을 승인됨에 따라 전자거래를 위한 정보검색과 데이터 전송에 XML의 활용도가 증가하고 있으며, 전자거래의 표준으로 그 중요성이 높아지고 있다[1][2]. 이처럼 XML이 광범위하게 이용됨에 따라 인터넷 상에서 발생할 수 있는 메시지 도청, 메시지 변조, 메시지 송신 및 수신 부인, 메시지 위조와 같은 보안상의 문제점이 존재하고 있다. 이러한 문제점을 해결하기 위하여 다양한 XML 보안의 정책들을 기반으로 하는 기법들이 개발되고 있다.

XML 보안에 관련된 정책들은 XML 암호화(Encryption), XML 서명(Signature), XKMS (XML Key Management Specification), SAML (Security Assertion Markup Language)[3], XACML(Access Control Markup Language) 등이 연구 되고 있다. 본 논문의 초점이 되는 XML 서명은 현재 사용 중인 시스템이 많이 있다. 또 XML 서명기법에 적시성을 제공하기 위하여 타임스탬프를 제시한 연구들로는 ASN.1구조를 이용한 XER(XML Encoding Rules), 타임스탬프가 추가된 XML 서명, 진보된 XML 서명(XML Advanced Electronic Signatures)등의 연구들이 있다[4-7][8][9]. 그러나 이러한 서명 기법들은 여러 가지 형태의 코딩기법이 결합되어 가독성(readability)이 떨어지고, 제대로 된 적시성

* 금오공과대학교 전자통신공학과
** 금오공과대학교 전자공학부

을 제공하지 못했다. 이러한 문제를 개선하기 위해서 Apvrille와 Girier는 XML에 기반한 TSP(Time Stamping Protocol)를 제공하는 서명 기법을 제안하였다[10]. Apvrille와 Girier는 XML서명에 TSP를 제안하기 위하여 XML schema로 표현된 요청과 응답 메시지를 서명하여 전송하였다. 하지만 Apvrille와 Girier가 제안 서명의 인증은 TSA(Trusted Security Authority ; 신뢰되는 제3자)의 신뢰성에 의존하였다. 이에 대한 해결책으로 분산 모델(Distributed model)과 연결 모델(Linked model)을 제시하고 있지만 이러한 해결책은 시스템의 많은 부하와 기억장소의 낭비라는 문제점이 존재한다.

본 논문에서는 Apvrille와 Girier가 제안한 기법의 문제점을 해결하기 위해 보다 효율적인 타임스탬프가 제공된 XML 서명 기법을 제안하였다. 제안된 기법은 사용자가 자신이 서명한 시점의 타임스탬프 정보를 해쉬된 문서와 함께 TSA에게 요청메시지를 보내고 요청메시지를 받은 TSA는 사용자의 타임스탬프 정보와 해쉬 문서를 검증 한 후, 자신의 타임스탬프 정보를 포함하여 사용자에게 응답 메시지를 보낸다. 응답 메시지를 받은 사용자는 자신의 타임스탬프 정보의 변경 유무를 확인·검증 한 후 이를 통한 트랜잭션을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 일반적인 XML 서명과 타임스탬프가 제공된 XML 서명에 관한 연구들에 대해서 간략하게 살펴보고, 3장에서는 기존 연구의 문제점을 해결하기 위한 사용자의 타임스탬프가 제공된 새로운 XML 서명 기법을 제안하고, 4장에서는 기존의 연구들과 본 논문에서 제시한 기법을 비교 분석한다. 마지막으로 5장에서는 결론을 제시한다.

2. 관련연구

본 장에서는 먼저 XML 서명에 대해서 살펴보고, 기존의 타임스탬프가 제공된 XML 서명 연구에 대해서 살펴본다.

2.1 XML 서명

XML 서명은 XML 문서에 대해 XML 형태의 서명을 생성하고 검증할 수 있는 기법이며, 전자문서에 대해 인증, 무결성, 부인봉쇄 등의 정보보호 서비스를 제공하고, 현재 표준화가 완료되었다. 이 표준은 W3C와 IETF가 함께 개발한 것으로 현재 나와 있는 표준 명세들로는 서명문법과 절차, 정규화 XML, 배타적 정규화 XML, XPath 필터, XML 서명 요구사항등이 있다.

기존의 전자서명은 수신자 측에서는 송신자가 보낸 데이터를 메시지와 서명으로 분리한 후 각각의 다이제스트 값을 생성하여 비교하였다. 따라서 수신자 측에서는 다이제스트를 계산해야 하는 단점을 가진다.

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

그림 1. XML 서명

하지만 XML의 경우 문서에 수신자가 생성한 다이제스트와 서명 값이 포함되어 있기 때문에, 수신자 측에서는 서명을 검증하기 위해서 송신자가 보낸 데이터를 메시지와 서명으로 분리하여 다이제스트 값을 계산할 필요가 없다는 장점을 가진다[11][12]. XML 서명 문서는 그림 1과 같이 signature 엘리먼트로 표현된다.

XML 서명의 처리 규칙은 생성 규칙과 검증 규칙으로 구성된다[13].

① 서명 생성 규칙

각 서명 대상 객체에 대하여 객체에 Transform들을 적용하고 결과 객체에 대해 다이제스트를 계산하여 Reference 엘리먼트들을 생성한다. 생성된 Reference 엘리먼트들과 SignatureMethod, CanonicalizationMethod를 포함하는 SignedInfo 엘리먼트를 생성한다. 그리고 생성된 SignedInfo 안에 지정된 알고리즘을 사용하여 SignedInfo에 대하여 정규화를 수행하고, 그 결과물에 SignatureMethod

를 사용하여 SignatureValue를 계산한다. 마지막으로 SignedInfo, Object(s), KeyInfo, SignatureValue 등을 포함하는 Signature 엘리먼트를 생성한다.

② 서명 검증 규칙

검증과정에서는 SignedInfo 안의 각 Reference 엘리먼트에 포함된 다이제스트를 검증하는 참조 검증(Reference Validation)단계와 SignedInfo를 바탕으로 계산된 서명을 검증하는 서명 검증(Signature Validation) 단계로 구분할 수 있다. 참조 검증 단계에서는 SinedInfo 원소를 정규화하고, Reference 엘리먼트에 정의된 DigestMethod를 사용하여 다이제스트를 구한다. 그리고 SingedInfo Reference 안의 DigestValue와 생성된 다이제스트 값을 비교하고, 일치하지 않으면 검증 실패로 간주한다. 서명 검증 단계에서는 KeyInfo 또는 외부 소스로부터 키 정보를 얻은 다음에, CanonicalizationMethod[14]를 사용하여 SignatureMethod의 정규형을 구하고, SingedInfo 엘리먼트에 대해서 SignatureValue를 비교·검증한다.

2.2 XML 서명에 관한 연구들

XML 서명을 위한 기존의 연구로는 ASN.1구조를 이용한 XER(XML Encoding Rules), 타임스탬프(Time Stamp)를 이용한 XML서명, 진보된 XML 서명 (XML Advanced Electronic Signatures)등의 연구가 있다[10]. 그러나 이러한 기법들은 서명은 제공하지만 여러 가지 형태의 코딩(coding)기법이 결합되어 가독성(readability)이 어렵고, 제대로 된 TSP(Time Stamping Protocol)는 제공하지 못하였다. 이러한 문제를 해결하기 위해서 Apvrille와 Girier는 XML에 기반하여 TSP를 제공하는 서명기법(AG 서명기법)을 제안하였다10].

AG 서명기법은 그림 2(a)(b)에서 볼 수 있듯이 요청(Request)과 응답(Response) 메시지 구조를 XML Schema로 표현하여 사용자와 TSA 사이에서 타임스탬프를 요청 또는 응답한다.

사용자는 그림 2(a)의 XML Schema로 정의한 해쉬된 문서를 TSA에게 전송한다. 요청 메시지를 받은 TSA는 사용자가 보낸 해쉬된 문서의 정당성을 확인한 후, 그림 3과 같이 자신의 타임스탬프 정보를 추가하여 서명한 후 사용자에게 그림 2(b)의 응답 메시지를 보낸다.

```
- <element name="TimeStampReq" type="tsp:TimeStampReqType">
 - <complexType name ="tsp:TimeStampReqType">
  - <sequence>
    <element name="MsgImprint" type="tsp:MsgImprintType"/>
    <element name="ReqPolicy" type="tsp:TSAPolicyType" minOccurs="0"/>
    <element name="Nonce" type="long" minOccurs="0"/>
   </sequence>
 </complexType>
 <attribute name="Version" type="positiveInteger"/>
 <attribute name="CertReq" type="boolean" default="false"/>
</element>
- <complexType name="tsp:MsgImprintType">
 - <sequence>
   <element name ="HashedMsg" type="ds:DigestValueType"/>
  </sequence>
  <attribute name="Algorithm" type="ds:DigestMethodType" use="required"/>
</complexType>
```

(a) 요청 구조

```
- <element name="TimeStampResp" type="tsp:TimeStampRespType">
 - <complexType name="tsp:TimeStampRespType">
  - <sequence>
    <element name="StatusInfo" type="tsp:statusInfoType"/>
    <element name="TimeStampDoc" type="ds:Signature"/>
   </sequence>
  </complexType>
</element>
```

(b) 응답 구조

그림 2. Apvrille와 Girier의 서명기법

다음 그림 3은 타임스탬프정보를 위한 XML Schema이다. 하지만 AG 서명기법은 다음과 같은 문제점을 가진다. 서명의 인증이 TSA의 신뢰성에 의존적이다. 이러한 문제를 해결하기 위해서 논문[10]에서는 분산 모델(Distributed model)과 연결 모델(Linked model)에 기반 한 해결책을 제시하고 있다.

```
- <complexType name="tsp:TimeStampInfoType">
 - <sequence>
    <element name="TSAPolicy" type="tsp:TSAPolicyType"/>
    <element name="MsgImprint" type="ds:DigestValue"/>
    <element name="Time" type="tsp:TSPTimeType"/>
...
</sequence>
...
</complexType>
- <simpleType name="TSPTimeType">
 - <restriction base="dateTime">
    <patterm value="\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:d{2}.\d{3}Z"/>
    <patterm value="\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:d{2}.\d{3}+\d{2}:\d{2}"/>
    <patterm value="\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:d{2}.\d{3}-\d{2}:\d{2}"/>
   </restriction>
</simpleType>
```

그림 3. TSA의 타임스탬프 정보

분산 모델에서는 타임스탬프에 대한 요청이 있을 때 이러한 요청을 여러 TSA에 보내어 신뢰성을 증가시키고자 하였고, 연결된 모델에서는 TSA에서 타임스탬프의 응답을 이전의 메시지를 저장하고 있는 리스트에 추가한다.

그렇게 함으로서 TSA에 의한 타임스탬프의 위조를 방지할 수 있다고 제시하였다. 그러나 이러한 기법은 상당히 시스템 많은 부하와 기억장소의 낭비를 가져온다.

그림 4는 분산 분산모델과 연결모델이 제시된 AG의 서명기법의 전체적인 시스템 구성을 보여준다.

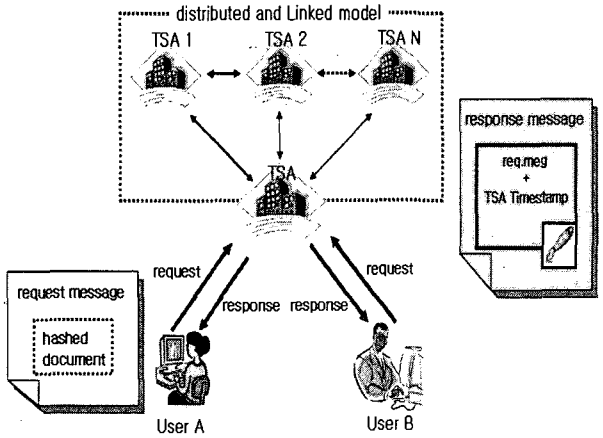

그림 4. 분산모델과 연결모델이 제시된 AG의 서명기법

3. 사용자의 정보가 추가된 XML 서명기법

본 장에서는 AG 서명기법이 가지는 문제를 해결하기 위하여 사용자의 타임스탬프정보가 포함한 XML 서명기법을 제안한다. AG가 제시한 TSP는 요청 메시지를 보내는 사용자가 단지 문서의 해쉬값을 보내면, TSA는 응답 메시지를 생성하기 위해서 사용자로부터 받은 해쉬값의 정당성 여부를 확인한 후 자신의 타임스탬프를 추가하여 받은 문서의 해쉬값에 서명하여 전송하는 방식을 사용한다. 그러므로 서명된 정보에 포함한 타임스탬프정보의 정확성은 단지 TSA의 신뢰성에 의존하게 된다. 본 논문에서는 AG 서명 기법이 가지는 TSA의 신뢰성에만 의존하는 문제점을 해결하기 위하여 두 가지 XML 서명기법을 제시하였다.

3.1 사용자의 타임스탬프가 추가된 기법

제시된 기법에서는 사용자는 자신이 서명한 시점의 타임스탬프 정보를 해쉬된 문서와 함께 TSA에게 요청메시지를 보내고 요청메시지를 받은 TSA는 사용자의 타임스탬프정보와 해쉬문서를 검증 한 후, 유효하다면 자신의 타임스탬프정보를 포함하여 사용자에게 응답 메시지를 보낸다. 응답 메시지를 받은 사용자는 자신의 타임스탬프정보의 변경 유무를 확인·검증 한 후 이를 통한 트랜잭션을 수행한다. 이처럼 제시하는 새로운 기법은 기존의 AG 서명기법이 가지는 타임스탬프의 정확성이 TSA의 신뢰성에 의존하는 존재하는 문제점을 해결한다. 사용자의 타임스탬프가 추가된 기법의 전체적인 처리과정은 그림 5와 같다.

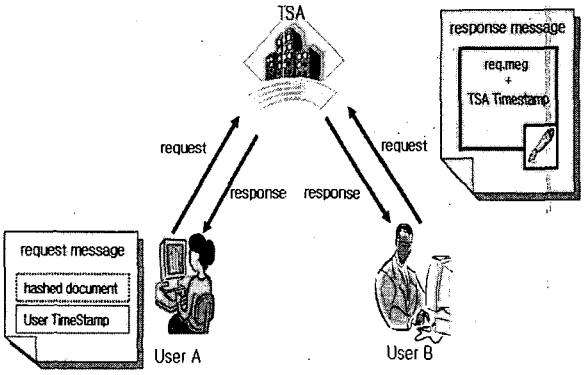

그림 5. 사용자의 타임스탬프가 추가된 기법

하지만 제안된 서명기법은 위조 될 수 있는 잠재적인 문제점이 존재한다. 이러한 문제점을 해결하기 위해서 본 논문에서는 사용자의 서명된 타임스탬프가 추가된 기법을 제안하였다.

3.2 사용자의 서명된 타임스탬프가 추가된 기법

3.1절에서 제시된 서명기법 사용자의 타임스탬프의 정보를 일반 텍스트(Plain text)형태로 전송되어 도청자가 위조할 수 있는 잠재적인 문제점이 존재한다. 이러한 문제점을 해결하기 위해서 본 논문에서는 사용자의 서명된 타임스탬프가 추가된 기법을 제시하였다. 전체적인 처리과정은 그림 6과 같다.

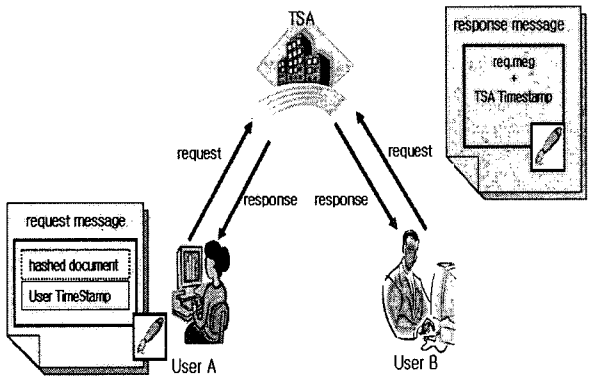

그림 6. 사용자의 서명된 타임스탬프가 추가된 기법

3.2.1 요청 구조

전체적인 요청메시지 구조의 형태는 사용자가 TSA에게 요청 메시지를 보낼 때, 해쉬된 문서와 자신이 서명한 타임스탬프정보가 포함된 요청 메시지를 생성하여 TSA에게 전송한다. 그런 다음 TSA가 보낸 응답 메시지를 검증 한 후 저장한다. 제안된 XML 서명 기법을 위한 네임스페이스는 그림 7과 같이 XML schema로 정의한다.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://ournamespace"
    xmlns:tsp="http://ournamespace"
    xmlns:ds="http://www.w3.org/2001/09/xmldsig#"
    elmentFormDefault="qualified">
```

그림 7. 네임스페이스 정의

그림 7의 네임스페이스 정의는 여러 개의 XML 언어로 된 요소와 속성을 포함하는 문서상의 이름 충돌 문제를 방지하기 위해서 사용되고 같은 이름을 다른 목적으로 사용하는 여러 필드를 포함하는 컴포넌트들을 결합하여 문서를 작성할 때 유용하게 사용된다.

```
- <element name="TimeStampReq"
      type="tsp:TimeStampReqType">
  -<complexType name ="tsp:TimeStampReqType">
   -<sequence>
     <element name="MsgImprint" type="tsp:MsgImprintType"/>
     <element name="ReqPolicy" type="tsp:TSAPolicyType"
      minOccurs="0"/>
     <element name="Nonce" type="long" minOccurs="0"/>
     <element name="UserSingedTimeStamp" type="ds:Signauture"/>
    </sequence>
   </complexType>
   <attribute name="Version" type="positiveInteger"/>
   <attribute name="CertReq" type="boolean" default="false"/>
  </element>
```

(a) Request 구조

```
-<UserSingedTimeStamp>
                ...
-<Object>
  -<SignatureProperties>
   -<SignatureProperty Id="TimeStampReqInfo" Target="ReqInfo">
    -<ReqInfo>
     <!-- the type of this element is TimeStampReqType..-->
                ...
    </ReqInfo>
   </SignatureProperty>
  </SignatureProperties>
 </Object>
</ UserSingedTimeStamp >
```

(b) User's signed Timestamp 정의

```
- <element name="TimeStampReq" type="tsp:TimeStampReqInfoType ">
  -<complexType name ="tsp:TimeStampReqInfoType ">
   -<sequence>
    <element name="ReqTime" type="tsp:TimeType"/>
   </complexType>
  </element>
 -<simpleType name="TimeType">
  -<restriction base="dateTime">
            ...
  </restriction>
  </simpleType>
```

(c) ReqInfo 구조

그림 8. Request 메시지 구조.

그림 8은 사용자의 서명이 추가된 XML 서명 기법의 요청메시지 구조를 XML 스키마로 표현한 형태이다.

요청메시지 구조에서는 그림(a) Request 정의가 TSA에게 전송되고, *"UserSingedTimeDoc"* 엘리먼트는 XML 서명의 형태로 규정한다.

그림(b)의 UserSingedDoc"의 형태는 XML서명의 형태로 서명되고, 실제 서명이 되는 항목으로는 그림(c)의 내용들이다. 즉 사용자는 TSA에게 요청을 할 때, 자신의 타임스템프의 정보가 포함된(그림 c) 메시지 구조를 자신이 서명을 하여(그림 b) TSA에게 요청를 한다(그림 a).

그림(c)를 보면 사용자가 서명한 타임스탬프정보를 추가하기 위하여 *<element name="ReqTime" type="tsp:TimeType">* 엘리먼트를 사용 하였다. 또 type의 형태 *"tsp:TimeType"*는 아래의 *"simpleType"* 형태로 표현되고, 사용자가 보내는 타임스탬프의 범위를 제한하기 위해서 *"restriction"* 엘리먼트를 사용하였고, 이름(name)은 *"dateTime"*으로 설정하였다.

"restriction" 엘리먼트는 이미 정의된 사용자 정의 심플 타입을 제한하여 새로운 심플 타입을 정의하는 것으로 값의 범위를 제한하기 위해서

사용되는 엘리먼트이다. 제한할 내용을 담고 있는 패싯(facet)엘리먼트로는 *"parttern"*을 사용하여 서명시간의 문자 데이터 포맷을 정한다. 사용자는 타임스탬프를 이러한 포맷으로 기술하여, 자신이 서명한 타임스탬프 정보와 함께 TSA에 보낸다.

3.3 응답 구조

응답구조는 요청구조 보다 좀 더 복잡한 형태를 가진다. TSA는 사용자로부터 받은 해쉬된 문서의 정당성과 타임스탬프의 정당성을 확인한 후 유효하다면 자신의 타임스탬프를 추가하여 요청메시지에 포함된 해쉬된 문서와 사용자의 타임스탬프를 서명하여 응답 메시지를 생성한다. 응답구조는 AG가 제시한 서명기법에 그림 2(b)의 응답메시지 형태와 유사하다.

TSA가 응답메시지를 사용자에게 전송할 때 자신의 타임스탬프정보가 그림 9와 같이 서명되어진다. 실제 서명되어지는 서명 항목에 해당되는 "TimeStampInfoType" 부분이 그림 10과 같이 수정된다. 그림 10의 타임스탬프정보 정의 부분에서는 사용자가 보낸 서명된 타임스탬프 정보와 TSA의 타임스탬프정보를 포함해서 정의한다.

```
-<TimeStampDoc>
 -<SignedInfo>
   <CanonicalizationMethod Algorithm="..."/>
   <SignatureMethod Algorithm="..."/>
  -<Reference URL="#TimeStampInfo"
   Type="http://www.w3.org/2000/09/xmldslg#SignatureProperties">
    <DigestMethod Algorithm="..."/>
    <Digestvalue>...</DigestValue>
   </Reference>
 -</SignedInfo>
  </SignatureValue>...</SignatureValue>
  <KeyInfo>...</KeyInfo>
 -<Object>
  -<SignatureProperties>
  -<SignatureProperty Id="TimeStampInfo" Target="TSTInfo">
   -<TSTInfo>
     <!-- the type of this element is TimeStampInfoType ..-->
     ...
    </TSTInfo>
   </SignatureProperty>
  </SignatureProperties>
 </Object>
</TimeStampDoc>
```

그림 9. TSA's 서명구조

```
-<complexType name="tsp:TimeStampInfoType">
 -<sequence>
   <element name="TSAPolicy" type="tsp:TSAPolicyType"/>
   <element name="MsgImprint" type="ds:DigestValue"/>
   <element name="ResTime" type="tsp:TSPTimeType"/>
   <element name="Nonce" type="long" minOccurs="0"/>
   <element name="Accuracy type=" AccuracyType" minOccurs="0"/>
   <element name="X509Qualifiers" type="ds:X509Data"/>
   <element name="UserSingedTimeStamp" type="ds:Signauture"/>
 </sequence>
  ...
 </complexType>
-<simpleType name="TSPTimeType">
 -<restriction base="TSAdateTime">
  ...
  </restriction>
</simpleType>
```

그림 10. TSA's TimeStampInfo 정의

즉, 응답요청 메시지를 받은 TSA는 정당성을 확인한 후 유효하다면 사용자의 서명에 다시 서명을 한 후 응답메시지를 사용자에게 보낸다. 응답 메시지를 받은 사용자는 검증 후 저장한다.

4. 분 석

본 장은 논문에서 제안한 사용자의 타임스탬프가 제공된 XML 서명기법과 기존의 XML 서명 기법과 XML 서명기법에 타임스탬프를 제공하는 진보된 XML 서명기법 그리고 Apvrille와 Girier가 제안한 XML 서명기법들 간의 특성을 비교 분석한다. 특성을 비교할 때, 다음과 같은 평가 항목으로 분석한다.

- TSP 제공 : XML에 TSP가 적용되었는가의 여부 확인을 하여야 한다.
- 가독성·조작성 : 가독성과 조작성은 XML이 가진 장점을 최대한 활용할 수 있어야 한다.
- XML 서명과 호환성 : 일반적으로 XML 서명과 호환성을 가져야 한다.
- 타임스탬프의 위조 가능 : TSA에 의해 문서가 위조될 수 있는 환경에서 사용자는 자신의 문서의 위조 유·무를 발견 할 수 있어야 한다.

일반적인 XML 서명기법은 가독성과 조작성이 우수하며, XML 서명과의 호환성을 제공하는

반면에, TSP가 제공되지 않으며, TSA에 의한 타임스탬프 정보가 위조 가능하다는 문제점을 가진다.

XML 서명기법에 타임스탬프를 제공하는 진보된 XML 서명 기법에서도 가독성과 조작성이 우수며, XML 서명과의 호환성을 제공하지만, 완벽한 TSP를 제공하지 못하며, 타임스탬프 정보가 위조 가능하다.

AG 서명기법에서는 가독성과 조작성이 우수하고, XML 서명과의 호환성을 제공하며, 요청, 응답 구조 형태의 TSP도 제공한다. 하지만 서명의 안정성이 TSA의 신뢰성에 의존하므로 TSA에 의한 타임스탬프정보가 위조될 잠재적 가능성을 가진다.

본 논문에서 제안한 TSP가 제공된 XML 서명기법에서는 사용자의 정보가 추가된 서명 기법으로 사용자의 타임스탬프가 추가된 기법과 사용자의 서명된 타임스탬프가 추가된 기법을 제안하였다. 제안한 서명기법은 사용자의 타임스탬프 정보를 일반 텍스트(Plain text)형태로 보내거나 서명을 하여 전송되어 TSA에 의한 타임스탬프의 위조 가능성을 방지 할 수 있다. 또한 XML 문법만으로 작성하여 가독성과 조작성이 쉬우며, XML 서명과의 호환성과 요청, 응답 구조 형태의 TSP를 제공한다.

<표 1> XML 서명기법에 대한 비교

서명기법 / 평가항목	XML 서명	진보된 XML 서명	Apvrille와 Girier의 XML서명	제안된 XML 서명
TSP 제공	제공안됨	제공안됨	제공	제공
가독성, 조작성	쉽다	쉽다	쉽다	쉽다
XML서명과호환성	제공	제공	제공	제공
타임스탬프의 위조가능	제공안됨	제공안됨	위조가능	위조불가

<표 1>에서 보여준 바와 같이 본 논문에서 제안한 사용자의 타임스탬프가 제공된XML 서명기법이 다른 기법보다 우수함을 확인 할 수 있다.

5. 결 론

본 논문에서는 XML 서명기법에 타임스탬프를 제공하는 연구들이 가지는 문제점 해결하기 위하여 두 가지 새로운 XML 서명기법을 제안하였다. 제안한 기법은 사용자의 타임스탬프를 추가한 기법과 사용자의 서명된 타임스탬프를 이용한 기법으로 구성된다. 제안한 기법은 기존의 관련 연구들에서 여러 가지 형태의 코딩기법의 결합으로 인한 상대적인 어려움과, 제대로 된 타임스탬프를 제공하지 못한다는 점 그리고 서명의 안정성은 오직 TSA의 신뢰성에 의존적이라는 문제점을 해결할 수 있었다. 더불어 본 논문의 분석에서 제시한 네 가지 평가항목을 통해 기존의 다른 기법들보다 우수함을 확인하였으며, 이러한 서명 기법을 통하여 보다 효율적인 XML 보안을 제공 할 수 있을 것으로 기대한다.

참 고 문 헌

[1] W3C, Extensible Markup Language(XML), http://www.w3c.org/XML

[2] William J.Pardi, *"XML in Action, Web Technology,"* Microsoft Press, 1999

[3] OASIS, Security Assertion Markup Language, http://www.oasis-open.org/committees/security/

[4] D. Eastlake, J. Reagle and D. Solo, (Extensible Markup Language) XML-Signature Syntax and Processing, Network Working Group, RFC 3275, March 2002.

[5] ETSI Technical Committee Security (SEC), XML Advanced Electronic Signatures (XAdES), Technical Specification 101 903 v1.1.1, February 2002.

[6] ITU-T Recommendation X.690, Information technology-ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), OSI networking and system aspects - Abstract Syntax Notation

One (ASN.1), Series X: Data networks and open system communications, Dec. 1997.
[7] ITU-T Recommendation X.693, Information technology - ASN.1 encoding rules: XML encoding rules(XER), OSI networking and system aspects - Abstact Syntax Notation One(ASN.1), Series X : Data networks and open system communications, Prepublished version at http://www.itu.int/ITUY/studygroups/com17/languages/X.6930901.pdf, Dec. 2001.
[8] W. Diffie, M. Hellman, "New Directions in Cryptography," *IEEE Trans. on Information Theory*, IT-22, pp.644-654, 1976.
[9] Y.Zheng, "Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost(Signature)+Cost(Encry-ption)," *LNCS 1294*, pp.165-179, 1997.
[10] A.Aprville and V.Girier, *"XML Security Time Stamping Protocol,"* Proceedings of the Information Security Solutions Europe Confrence (ISSE 2002) Oct. 20
[11] *An Introduction to XML DigitalSigntures*; http://www.xml.com/pub/a/2001/08/08/xmldsig.html.
[12] W3C, XML-Signature Syntax and Processing, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, 2002.
[13] M.Bartel, J.Boyer, B.Fox, B.LaMacchia and E.Simon, *"XML Signature Syntax and Processing."* http://www.w3.org/TR/xmldsig-core/, 2002.
[14] W3C, "Canonical XML Version 1.0," http://www.w3.org/TR/2000/CR-xml-c14n-20001026, 2000.

이 원 진 (Won-Jin Lee)

학생회원
- 2002년 2월 경일대학교 컴퓨터공학과(공학사)
- 2004년 8월 경북대학교 산업대학원 컴퓨터공학과(공학석사)
- 2005년 현재 금오공과대학교 대학원 전자통신공학과(박사과정)
- 2005년~현재 금오공과대학교 전자공학부 시간강사

<관심분야> : 정보보호, XML 보안, 센서네트워크 보안

전 일 수 (Il-Soo Jeon)

정회원
- 1984년 2월 경북대학교 전자공학과(공학사)
- 1988년 2월 경북대학교 대학원 전자공학과(공학석사)
- 1995년 2월 경북대학교 대학원 전자공학과(공학박사)
- 1984년~1985년 삼성전자(주)
- 1989년~2004년 경일대학교 컴퓨터공학과 교수
- 2004년~현재 금오공과대학교 전자공학부 조교수

<관심분야> : 정보보호, 패턴인식