XML 보안 기술의 소개 및 표준화 동향

박 대 하* 김 영 갑** 문 창 주** 백 두 권**

◈ 목 차 ◈

1. 서 론

XML(eXtensible Markup Language)[1]은 데이터 또는 문서의 구조적인 표현을 통해 네트워크 상에서 정보 시스템 간의 정보 공유에 효율적이고 표준화된 메커니즘을 제공한다. W3C(World Wide Web Consortium)[2]에서는 XML 표준화 작업의 일환으로 문서의 구조와 내용을 정의하기 위한 XML Schema[3], 문서의 구성요소에 식별 가능한 명칭을 부여하기 위한 XML namespace[4], 문서의 일부분을 참조하기 위한 XPath[5], 문서의 형식을 변환할 수 있는 XSLT[6] 등을 권고안(recommendation)으로 제정하였다. 그 밖에도 W3C에서는 XML 기술을 다양한 어플리케이션에 활용할 수 있도록 지원하는 관련 표준안(예: XQuery[7], XPointer[8], XForm[9] 등)을 지속적으로 개발하고 있다. 표준화된 XML 기술은 XML 언어의 유연성과 확장성을 최대한 활용하여 단순한 문서의 전달이나 표시 방식을 넘어서 인터넷으로 연결된 여러 기업이나 조직간의 이질적인 어플리케이션 환경을 하나로 묶어주는 매개체의 역할로 간주되고 있다. 이러한 관점은 웹 환경의 어플리케이션 간에 기능을 공유할 수 있는 XML 프로토콜인 SOAP[10]이 표준화되고 UDDI[11], WSDL[12] 등의 관련 기술이 개발되면서 웹 서비스(Web Service)[13]라는 개방형 비즈니스 모델을 등장시켰다. 웹 서비스 플랫폼을 통한 어플리케이션 간의 통신은 XML 기반의 국제 표준인 ebXML[14]과 결합하여 B2C와 B2B 등의 전자상거래 시스템의 구축에 획기적인 전환점이 될 것으로 예상된다.

그러나, 개방형 환경을 지향하는 XML 기술이 전자상거래 어플리케이션의 기반을 이루기 위해서는 보안 문제에 대한 이슈가 해결되어야 한다. XML 형식으로 전송되는 데이터 또는 문서의 내용이 원하지 않는 제3 자에게 노출되거나 변조되지 않도록 보장해야 하며, XML 문서의 작성자 또는 출처에 대한 인증이 가능해야 한다. 또한 XML 프로토콜을 이용한 어플리케이션 간의 통신에서 양방향 인증이나 접근 권한에 대한 관리가 필요하다. 이러한 보안 요구사항은 기존의 네트워크 보안 관점에서 볼 때 특별한 내용이라고 할 수는 없지만 XML의 특징인 유연성과 확장성을 보장하고 현재 사용되고 있는 XML 표준 기술이나 개발 도구와 호환성을 유지할 수 있도록 XML 보안 기술을 표준화하는 것은 매우 중요한 작업이다. 현재 W3C에서는 XML 기술에 필요한 기본적인 보안 메커니즘인 전자서명과 암호화를 각각 XML Signature[15]과 XML Encryption[16] 권고안으로 제정하였으며, 보안 기술에 필요한 키 정보를 관리하기 위한 XKMS(XML Key Management Specification)[17]를 표준화하고 있다. OASIS (Organization for

* (주)시큐리티테크놀로지스
** 고려대학교 소프트웨어시스템 연구실

the Advancement of Structured Information Standards) [18]에서는 XML을 이용하는 어플리케이션 간에 인증 정보를 공유하고 접근 권한을 부여할 수 있도록 SAML (Security Assertion Markup Language)[19]과 XACML(XML Access Control Markup Language)[20]를 표준화하고 있다.

본 투고에서는 XML 보안 기술에 대한 요구사항과 이에 따른 보안 표준의 종류와 특성을 살펴보고, 현재 W3C와 OASIS에서 개발되고 있는 XML 보안 기술의 표준화에 대한 현황 및 그 활용성을 제시하고자 한다.

2. XML 보안 기술의 요구사항

XML 기술이 개방형 네트워크 환경의 매개체 역할을 하는 점에서 볼 때, 기존의 네트워크 보안에서 다루어진 요구사항이 XML 보안에서도 그대로 반영되어야 한다. XML로 표현된 내용이 허가되지 않은 외부에 노출되지 않도록 하는 기밀성(confidentiality), 허가되지 않은 내용의 변조를 검출할 수 있는 무결성(integrity), 내용의 작성자 또는 출처를 확인할 수 있는 인증(authentication), 네트워크 자원에 접근할 수 있는 권한을 결정하는 권한부여(authorization) 등이 필요한 보안 요구사항이다. 기밀성과 무결성은 네트워크로 전송되고 있는 상태에서 뿐만 아니라 임의의 중도 처리 시스템을 거쳐서 최종 어플리케이션에 의해 저장되는 상태까지 지속적으로 유지되어야 한다. 따라서 기존에 널리 사용되고 있는 SSL/TLS[21][22] 형태의 통신 채널 보안 방식보다는 S/MIME[23] 형태의 메시지 보안 방식이 더 적합하다. 또한 인증이나 부인봉쇄(non-repudiation)의 기능을 제공하기 위하여 전자 서명을 제공해야 하며, 기존의 전자 서명이 근간을 두고 있는 PKI(public-key infrastructure)[24]를 XML 보안 기술에서 적절히 활용해야 한다. 웹 서비스와 같이 XML 기반의 어플리케이션 간에 접근이 필요한 환경에서 자동화된 인증과 권한부여가 가능하도록 지원해야 한다.

이러한 보안 요구사항에 더불어 XML 보안 기술은 XML이 제공하는 장점을 최대한 이용할 수 있는 방향으로 개발되어야 한다. 보안 기술의 적용 과정에서 기존의 XML 표준이 변형되거나 축소되지 않아야 한다. 또한 기존에 사용되고 있는 XML 도구와 XML 프로토콜 등에 충돌 없이 적용될 수 있어야 한다. 이미 XML 표준을 기반으로 개발된 도구나 프로토콜과의 충돌은 XML 보안 기술의 사용을 제한하게 된다. 개발자의 입장에서는 기존 응용 프로그램의 변동을 최소화해야 한다. 만일 XML 보안을 적용하기 위해서 기존의 애플리케이션의 많은 부분이 변경되어야 한다면 XML 보안 기술의 사용을 주저하게 될 것이다.

W3C와 OASIS의 XML 보안의 표준화는 앞서 언급한 XML 보안 기술의 요구사항에 따라 이루어지고 있다. 대표적인 XML 보안 표준은 다음과 같다.

- ▶ XML 전자서명(XML Signature) - XML 문서와 관련 정보의 무결성을 보장하고 전자서명의 생성과 검증을 통하여 메시지와 서명자의 인증 기능을 제공
- ▶ XML 암호화(XML Encryption) - XML 문서와 관련 정보를 암호화하는 절차와 표현 방식을 정의하여 메시지의 전송 또는 저장에 요구되는 기밀성을 제공
- ▶ XML 키 관리 명세(XKMS) - PKI 환경에서 XML 전자 서명의 처리에 필요한 공개키의 배분 및 등록 프로토콜을 정의
- ▶ 보안 선언 표기 언어(SAML) - 자원에 대한 접근을 통제하고 트랜잭션 참가자의 신원을 확인할 수 있도록 인증 및 권한부여에 필요한 정보를 XML 형식으로 표현
- ▶ XML 접근통제 표기 언어(XACML) - 권한부여에 따른 결정을 위해서 접근 통제 정책의 작성에 필요한 규칙을 XML 형식으로 표현

3. XML 보안 표준의 특징 및 연관성

이 장에서는 앞서 언급한 XML 보안 표준의 특징에 대해서 각 표준별로 기술하고, 다른 XML 표준과의 연관성에 대해서 살펴보기로 한다.

3.1 보안 표준의 특징

1) XML 전자 서명(XML Signature)

흔히 수신자가 송신자로부터 어떤 정보를 받았을 때 이 정보의 무결성을 판단하고 정보 제공자를 인증하기 위한 방법으로 전자 서명(digital signature)을 사용한다. SSL/TLS와 같은 보안 통신 채널이나 CRC와 같은 검사 기법은 네트워크 상에서 문서의 교환 시에는 무결성이 보장되지만 일단 문서가 서버나 클라이언트로 전송된 후에 문서의 무결성은 보장되지 않는다. 그러므로 문서의 전송 도중에서뿐만 아니라 저장된 문서의 무결성을 보장할 수 있는 전자 서명이 필요하다. XML 전자 서명은 무결성뿐만 아니라 전자 서명을 생성하고 증명하여 문서 작성자(또는 제공자) 및 출처에 대한 이동성 있는 인증을 가능하게 하는 표준화된 방법을 제공한다.

XML 전자 서명은 전체 XML 문서나 특정 XML 요소(element) 및 요소의 내용, 임의의 관련 문서, 서명에 포함된 속성 등 다양한 대상에 대해서 서명을 생성하고 검증할 수 있도록 해준다. <Signature> 요소를 사용해서 서명값과 관련 정보를 표현하며, 이 요소를 전자 서명의 대상에 되는 원본 XML 문서에 내부에 추가하거나(이를 enveloped signature라고 함) 원본 XML 문서를 <Signature> 요소 내부에 추가하는 방법(이를 enveloping signature라고 함)을 사용할 수 있다. 전자 서명은 기존의 공개키 암호화 방식(예: RSA 또는 DSA)을 해쉬 함수와 함께 사용하므로 논리적으로 동일한 XML 문서라도 물리적인 표현 방법의 차이(부호화 방법, 공백 문자, 요소의 속성 순서 등)에 따라 전혀 다른 서명값이 생성될 수 있다. 따라서 XML 문서의 유연성을 보장하기 위하여 서명 정보에 표준화된 변형 기법을 참조할 수 있도록 포함시켜야 한다.

그림 1은 XML 전자 서명의 예를 나타낸다. <SignedInfo> 요소([s02~s12])는 실제로 서명된 정보를 나타내며, <CanonicalizationMethod> 요소([s03])는 해쉬값을 만들기 전에 정규화하는데 사용되는 알고리즘을 가리킨다. <SignatureMethod> 요소([s04])는 전자 서명 알고리즘(이 예제에서는 표준 전자 서명 알고리즘인 DSA[25]를 사용)을 가리키며, <Reference> 요소([s05~s11]는 해쉬값을 생성하는 알

```
[s01]  <Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
[s02]    <SignedInfo>
[s03]      <CanonicalizationMethod
             Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
[s04]      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
[s05]      <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
[s06]        <Transforms>
[s07]          <Transform
                 Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
[s08]        </Transforms>
[s09]        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[s10]        <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
[s11]      </Reference>
[s12]    </SignedInfo>
[s13]    <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
[s14]    <KeyInfo>
[s15a]     <KeyValue>
[s15b]       <DSAKeyValue>
[s15c]         <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
[s15d]       </DSAKeyValue>
[s15e]     </KeyValue>
[s16]    </KeyInfo>
[s17]  </Signature>
```

(그림 1) XML 전자 서명의 예

고리즘과 생성된 해쉬값을 나타낸다.

<SignatureValue> 요소([s13])는 서명값을 가리키며, <KeyInfo>([s14~s16])는 서명값의 검증에 필요한 공개키 정보를 포함하고 있다.

2) XML 암호화(XML Encryption)

XML 암호화는 XML 문서가 전송 또는 저장될 때, 정보의 기밀성을 유지하기 위해서 사용된다. 무결성의 경우와 마찬가지로 SSL/TLS 등의 보안 통신 채널은 정보를 전송하는 동안만 기밀성을 제공하고 정보가 서버에 저장될 때의 기밀성은 제공하지 못한다. 따라서 XML 암호화는 허가되지 않은 제 3자에게 특정 정보를 노출하고 싶지 않을 경우에 사용된다.

XML 암호화는 주로 대칭키 알고리즘(DES 또는 Triple DES[26])을 사용하여 내용을 암호화하지만 대칭키의 효율적인 배분이 어려운 점을 고려하여 수신자의 공개키를 사용하여 대칭키를 암호화하는 방식을 함께 사용한다. XML 전자 서명과 마찬가지로 XML 문서 전체를 암호화하거나 특정 요소 및 그 내용을 암호화할 수 있고, 암호화된 내용과 사용된 키 및 알고리즘 정보를 패키지로 만들어 전송할 수 있도록 해준다. 필요한 경우에는 XML 전자 서명을 함께 적용하여 무결성과 인증을 보장할 수 있다.

그림 2는 XML 암호화의 예를 나타낸다. <EncryptedData> 요소([s03~s21])는 암호화된 정보를 나타내며, <EncryptionMethod> 요소([s04])는 내용의 암호화에 사용된 대칭키 알고리즘(이 예제에서는 Triple DES를 사용)을 가리킨다. <ds:KeyInfo> 요소([s05~s17])는 XML 전자 서명에 정의된 명칭을 사용하여 대칭키의 암호화에 사용된 수신자의 공개키를 표현하고, 내부에는 <EncryptedKey> 요소([s06~s15])를 이용하여 공개키 암호화 알고리즘(이 예제에서는 RSA[27]를 사용)과 암호화된 대칭키 값을 나타내고 있다. 마지막으로 <CipherData> 요소([s18~s20])는 실

```
        <PatientRecord
            xmlns="http://www.medical.org/" xmlns:lab="http://www.lab.org/tests">
[s01]       <Name>John Doe</Name>
            <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
[s02]
[s03]         xmlns='http://www.w3.org/2001/04/xmlenc#'>
              <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#3des-cbc'/>
[s04]         <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
[s05]           <EncryptedKey Id='EK' xmlns='http://www.w3.org/2001/04/xmlenc#'>
[s06]             <EncryptionMethod
[s07]               Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
                  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
[s08]
[s09]               <ds:KeyName>Dr Kutter's public key pair</ds:KeyName>
[s10]             </ds:KeyInfo>
[s11]             <CipherData>
[s12]               <CipherValue>xyzabc</CipherValue>
[s13]             </CipherData>
[s14]             <CarriedKeyName>Dr Kutter's symmetric key</CarriedKeyName>
[s15]
[s16]           </EncryptedKey>
[s17]           <ds:KeyName>Dr Kutter's symmetric key</ds:KeyName>
[s18]         </ds:KeyInfo>
[s19]         <CipherData>
[s20]
[s21]           <CipherValue>a17xj2z</CipherValue>
[s22]         </CipherData>
            </EncryptedData>
        </PatientRecord>
```

(그림 2) XML 암호화의 예

제 XML 문서의 암호화된 값을 포함한다.

3) XML 키 관리 명세(XML Key Management Specification; XKMS)

XML 전자 서명과 XML 암호화에 사용되는 암호화 알고리즘에는 적당한 키가 함께 사용되어야 한다. 흔히 PKI 환경에서는 인증기관(certification authority)이라는 신뢰할 수 있는 제 3자를 통하여 키의 등록이나 키 정보의 공유가 이루어진다. XKMS는 클라이언트가 네트워크 상에서 믿을 수 있는 공유 키 관리 서비스들에 접근할 수 있는 프로토콜을 정의한다. 이 프로토콜은 크게 키 정보 서비스(XML Key Information Service Specification; X-KISS)와 키 등록 서비스(XML Key Registration Service Specification; X-KRSS)로 구분된다. X-KISS는 XML 전자 서명에 포함된 공개키 정보(<ds:KeyInfo> 요소)의 실제 내용

```
        <Register>
          <Prototype Id="keybinding">
[s01]       <Status>Valid</Status>
[s02]       <KeyID>mailto:Alice@cryptographer.test</KeyID>
[s03]       <ds:KeyInfo>
[s04]         <ds:KeyValue>
[s05]           <ds:RSAKeyValue>
[s06]             <ds:Modulus>
[s07]               998/T2PUN8HQInhfkJwA56UD0a1oYq7E
[s08]               uAszNqBoOqfarJlsfcVKLob1hGnQ/l6xw
[s09]             </ds:Modulus>
[s10]             <ds:Exponent>AQAB</ds:Exponent>
[s11]           </ds:RSAKeyValue>
[s12]         </ds:KeyValue>
[s13]         <ds:KeyName>mailto:Alice@cryptographer.test</ds:KeyName>
[s14]       </ds:KeyInfo>
[s15]       <PassPhrase>Pass</PassPhrase>
[s16]     </Prototype>
[s17]     <AuthInfo>
[s18]       <AuthUserInfo>
[s19]         <ProofOfPossession>
[s20]           <ds:Signature URI="#keybinding"
[s21]                [RSA-Sign (KeyBinding, Private)] />
[s22]         </ProofOfPossession>
[s23]         <KeyBindingAuth>
[s24]           <ds:Signature URI="#keybinding"
[s25]                [HMAC-SHA1 (KeyBinding, Auth)] />
[s26]         </KeyBindingAuth>
[s27]       </AuthUserInfo>
[s28]     </AuthInfo>
[s29]     <Respond>
[s30]       <string>KeyName</string>
[s31]       <string>KeyValue</string>
[s32]       <string>RetrievalMethod</string>
[s33]     </Respond>
[s34]   </Register>
[s35]
```

(그림 3) XKMS 키 등록 서비스의 예

(인증서 정보, 공개키 매개변수, 인증서 폐지 정보 및 유효성 상태 등)을 전송하는데 사용되며, X-KRSS는 클라이언트가 이러한 공개키 정보를 믿을 수 있는 인증 기관에 등록 또는 폐지를 요청하는데 사용된다.

그림 3은 XKMS를 이용하여 클라이언트가 생성한 공개키의 등록을 요청하는 예를 나타낸다. <Register> 요소([s01~s35])로 표현된 등록 요청에는 클라이언트가 생성한 RSA 공개키에 대한 정보(키 상태, 키 매개변수, 키 명칭 등)를 <Prototype> 요소([s02~s17])로 가지며, 클라이언트가 현재 공개키에 대응하는 개인키를 가지고 있는지를 확인할 수 있는 정보인 <AuthInfo> 요소([s18~s29]), 그리고 등록 처리의 결과로 서버에서 반환될 정보를 나타내는 <Response> 요소([s30~s34])를 포함하고 있다. 키 관리 서비스를 구현한 서버에서는 처리 결과와 인증서 관련 정보를 <RegisterResult> 요소를 사용하여 반환하게 된다.

4) 보안 선언 표기 언어(Security Assertion Markup Language; SAML)

인증은 어떤 사용자의 신원을 확인하는 절차이며, 자원에 대한 접근 통제나 트랜잭션의 참가자 확인, 그리고 지속적인 개인 정보의 보존에 사용된다. 또한 한 번 성공한 인증은 싱글사인온(single sign-on)이나 제3자 인증 서비스에도 사용될 수 있다. 권한부여는 인증된 사용자가 시스템의 특정 자원에 접근할 수 있는지 혹은 특정 행위를 할 수 있는지를 결정하는 과정이다.

SAML은 인증과 권한부여가 이루어진 시점과 방식을 명시하는 선언(assertion)에 포함되는 XML 어휘에 대한 정의와 전자 서명과의 연동 방법에 대해서 기술한다. 또한 SAML로 선언된 내용을 전송하기 위한 요청 및 응답 프로토콜과 SOAP와 같은 XML 프로토콜과의 연결을 정의하고 있다.

그림 4는 SAML을 이용한 인증 선언의 예이다. <Assertion> 요소([s01~s19]) 내에는 선언의 유효 기간과 이용자를 나타내는 <Conditions> 요소([s02~s06])와 선언의 유효성을 지원하는 정보(증거, 증명 방법, 갱신 포인터 등)를 나타내는 <Advice> 요소([s07~s10]), 그리고 인증 기법과 인증 시점 및 인증된 주체에 대한 정보를 나타내는 <Authentication Statement> 요소([s11~s17])가 포함된다. 부가적으로 선언 내용에 대한 무결성을 보장하기 위하여 XML 전자 서명의 형식으로 <ds:Signature> 요소([s18])를

```
        <Assertion>
[s01]     <Conditions NotBefore="dateTime" NotOnOrAfter="dateTime">
[s02]       <AudienceRestrictionCondition>
[s03]         <Audience>http:/www.example.com/Members</Audience>
[s04]       </AudienceRestrictionCondition>
[s05]     </Conditions>
[s06]     <Advice>
[s07]       <AssertionIDReference>id</AssertionIDReference>
[s08]       <Assertion>...</Assertion>
[s09]     </Advice>
[s10]     <AuthenticationStatement AuthenticationMethod="urn:ietf:rfc:2246"
[s11]       AuthenticationInstant="dateTime">
[s12]       <Subject>
[s13]         <NameIdentifier Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
[s14]           john_doe@example.com
[s15]         </NameIdentifier>
[s16]       </Subject>
[s17]     </AuthenticationStatement>
[s18]     <ds:Signature>XML Digital Signature for assertion</ds:Signature>
[s19]   </Assertion>
```

(그림 4) SAML 인증 선언의 예

추가할 수 있다.

5) XML 접근 통제 표기 언어(XML Access Control Markup Language; XACML)

SAML은 인증된 주체에 대한 여러 가지 정보를 제공하고 있지만 실제로 접근 통제가 이루어지는 목적 시스템에서는 자체적인 권한부여 규칙을 정의할 수 있어야 한다. 예를 들어, 외국 대사관에서 발급된 비자에는 소유자에 대한 여러 정보(이름, 생년월일, 주소 등)와 비자의 등급 등이 표현되어 있지만, 실제로 공항의 입국심사대를 통과하거나 외국에 취업을 하기 위해서는 기관별로 비자의 일부 내용과 다른 필요한 증명서를 이용하여 소유자의 행위(입국 또는 취업)를 결정하게 된다. SAML은 비자와 같은 역할을 하지만 XACML은 SAML이 제공하는 정보에 따른 권한부여를 결정할 수 있는 정책을 명시할 수 있도록 해준다.

그림 5는 XACML을 사용하여 권한부여를 결정할 수 있는 규칙을 작성한 예이다. <Target> 요소([s2~s16])는 규칙이 적용되는 대상을 나타내며, 접근의 주체를 가리키는 <Subjects> 요소([s3~s7])와 접근의 대상이 되는 자원을 가리키는 <Resources> 요소([s8~s12]), 그리고 접근 가능한 행위를 가리키는 <Actions> 요소([s13~s15])를 포함한다. <Condition> 요소([s17~s22])를 사용하여 규칙의 생성과 평가에 필요한 다양한 조건을 표현할 수 있다.

3.2 XML 표준 간의 연관성

XML 보안 표준은 기존의 XML 표준을 기반으로 개발되어 보안 기술의 적용 과정에서 이미 사용되고 있는 XML 도구와 XML 프로토콜 등에 충돌 없이 적용될 수 있도록 지원하고 있다. 또한 각각의 XML 보안 표준간에도 재사용을 원칙으로 하여 다른 보안 표준에서 제공하는 기능을 최대한 활용할 수 있도록 개

```
[s1]  <Rule RuleId="//medico.corules/rule3" Effect="Permit">
[s2]    <Target>
[s3]      <Subjects>
[s4]        <saml:Attribute
              AttributeName="RFC822Name" AttributeNamespace="//medico.com">
[s5]          <saml:AttributeValue>*</saml:AttributeValue>
[s6]        </saml:Attribute>
[s7]      </Subjects>
[s8]      <Resources>
[s9]        <saml:Attribute
              AttributeName="documentURI" AttributeNamespace="//medico.com">
[s10]         <saml:AttributeValue>//medico.com/records.*</saml:AttributeValue>
[s11]       </saml:Attribute>
[s12]     </Resources>
[s13]     <Actions>
[s14]       <saml:Action>read</saml:Action>
[s15]     </Actions>
[s16]   </Target>
[s17]   <Condition>
[s18]     <Equal>
[s19]       <AttributeDesignator
              AttributeName="urn:oasis:names:tc:xacml:identifiers:AccessSubject" />
[s20]       <AttributeDesignator AttributeName="patientName" />
[s21]     </Equal>
[s22]   </Condition>
[s23] </Rule>
```

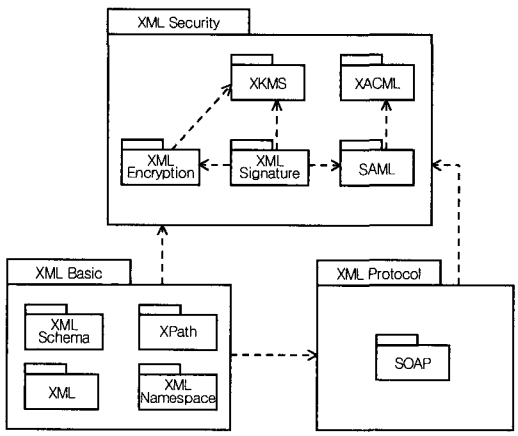
(그림 5) XACML 권한부여 규칙의 예

발되고 있다.

다른 XML 표준과 마찬가지로 모든 XML 보안 표준은 XML 명세와 XML 스키마, XPath, XML namespace 등의 기본적인 XML 표준을 바탕으로 작성되고 있다. 특히 XML 전자서명과 XML 암호화에서 서명 또는 암호화될 데이터를 XPath로 표현하여 문서의 특정한 일부분만 처리할 수 있도록 해준다. 또한 XML namespace를 사용하여 XML 암호화에서 암호화된 키 정보를 표현하는데 XML 전자서명의 <KeyInfo> 요소를 표현하거나, XML 전자서명에서 XML 암호화로 처리된 데이터를 암호화할 수 있는 확장성을 제공한다. XKMS에서도 공개키 정보를 표현하기 위하여 XML 전자서명을 이용하거나 개인키 정보를 암호화하는데 XML 암호화 표준을 이용하고 있다.

SAML은 자체적으로 무결성이나 기밀성을 제공하고 있지는 않지만 XML 전자서명이나 XML 암호화를 이용하여 선언된 내용의 무결성이나 기밀성을 지원하여 향상된 보안 기능을 제공할 수 있다. XACML은 SAML에 언급된 선언을 이용하여 세부적인 접근 통제 규칙의 설정이 가능하다. 응답 및 요청 프로토콜을 기반으로 운영되는 XKMS 및 SAML은 XML 프로토콜인 SOAP이나 기타 다양한 표준 프로토콜(HTTP, SMTP 등) 상에서 이용할 수 있다.

이와 같은 XML 보안 표준과 기타 XML 표준 간의 연관성을 UML의 컴포넌트 다이어그램으로 표현하면 그림 6과 같다(여기서 점선 화살표는 <<use>> 관계를 나타냄).



(그림 6) XML 보안 표준의 연관성

4. XML 보안 표준화 현황 및 활용

4.1 XML 보안 표준화 현황

XML 보안 기술의 표준화는 주로 W3C의 작업 그룹(working group)과 OASIS의 보안 서비스 기술 위원회(security service technical committee)에서 개발 작업을 진행하고 있다. W3C에서는 권고안(recommendation)의 작성이 최종 단계에 해당하며, 현재(2003년 3월) XML 전자서명과 XML 암호화는 권고안의 상태로 제공되고 있다. 현재 XKMS 2.0은 W3C에서 권고안으로 제공되기 이전 단계인 작업 초안(working draft)으로 아직 개발되고 있는 상태이다. OASIS에서는 SAML 1.0이 내부 표준안으로 제정되었으며, 현재 SAML 1.1에 대한 초안을 검토하고 있다. XACML 1.0은 최근 OASIS 표준안으로 제정되었다.

XML 보안 기술의 표준화 현황을 표준화 기관과 표준화 상태 및 관련 표준에 따라 정리하면 [표 1]과 같다.

4.2 XML 보안 표준의 활용

XML 보안 표준의 활용은 기술의 확장성(extendability)과 호환성(compatibility) 및 시장성(marketing)의 측면에서 살펴볼 수 있다.

3.2의 XML 보안 표준 간의 연관성에서 알 수 있듯이 추후 개발될 다양한 XML 보안 표준에서는 기존의 XML 보안 표준이 갖는 확장성을 최대한 수용하게 될 것이다. 현재 IBM과 Microsoft를 중심으로 개발 중인 웹 서비스 보안 표준(WS-Security)에서는 W3C의 권고안으로 제정된 XML 전자서명과 XML 암호화를 SOAP과 결합하여 사용하는 방안을 제공하고 있다. 또한 ContentGuard에서 XML 기반의 DRM(digital rights management; 디지털 저작권 관리) 표준으로 개발한 XrML에서도 XML 전자서명과

XML 암호화 표준을 이용하고 있으며, 전자상거래 표준으로 등장한 ebXML에서도 XML 등록기(registry)의 보안에 XML 보안 표준을 채택하고 있다. XML 보안 기술은 한국전자통신연구원(ETRI) 등 국책연구기관에서도 연구를 진행하였으며, 한국전산원과 함께 전자상거래통합포럼[28]에서 국내 표준화를 진행하고 있다.

IBM, Microsoft, Sun Microsystems와 같은 대규모 IT 기업과 Baltimore, Entrust, Verisign 등의 대형 보안 관련 회사에서는 XML 보안 표준과 호환이 되는 솔루션을 자체적으로 개발하고 있다. XML 보안 표준은 보안 솔루션 간의 호환성을 지원하므로 추후 SSL/TLS와 같은 보안 프로토콜과 S/MIME과 같은 보안 메일은 물론이고 PKI 연동 클라이언트와 서비스, 웹 기반의 싱글사인온 등의 보안 솔루션이 대부분 XML 보안 표준에 따르는 솔루션으로 대치될 것으로 예상된다. 국내의 경우에도 한국정보인증, 이니텍, 비씨큐어, 시큐리티테크놀로지스 등의 PKI 업체에서 XML 보안 기술을 이용하여 전자상거래용 보안 솔루션을 개발하고 있지만, 현재 개발된 XML 보안 표준에 따른 호환성 테스트가 제대로 이루어져 있지 않은 현실이다.

국제적인 IT 리서치 회사인 ZapThink에서는 2006년도에 이르면 전체 보안 시장의 65%에 이르는 44억 불 규모의 매출이 XML과 웹 서비스 보안 솔루션에서 발생할 것으로 예상하고 있다. 웹 서비스를 기반으로 하

(표 1) XML 보안 기술의 표준화 현황

	표준화 기관 및 그룹	표준화 상태	관련 표준
XML 전자서명	• W3C/IETF joint : XML Signature Working Group	• 2000년 1월: 초안 승인 • 2002년 2월: 권고안 승인 - "XML Signature Syntax and Processing" • IETF에서 RFC3275로 표준화	• XML Signature Syntax and Processing • Canonical XML • XPath Filter • Additional XML Security URIs • XML Signature Requirements
XML 암호화	• W3C : XML Encryption Working Group	• 2000년 12월: 초안 승인 • 2002년 12월: 권고안 승인 - "XML Encryption Syntax and Processing"	• XML Encryption Syntax and Processing • XML Encryption Requirements • Decryption Transform for XML Signature • Additional XML Security URIs
XKMS	• W3C : XML XKMS Working Group	• 2003년 3월 현재 : 작업 초안	• XML Key Management Requirements • XML Key Management Specification • XML Key Management Specification - Bulk Operation
SAML	• OASIS : Security Service Technical Committee	• 2002년 11월 : OASIS open 표준	• Security Assertion Markup Language v1.0
XACML	• OASIS : Security Service Technical Committee	• 2003년 2월 : OASIS open 표준	• eXtensible Access Control Markup Language v1.0

는 전자 거래, 전자 계약, 전자 입찰 등의 다양한 전자상거래 어플리케이션과 전자 정부의 구축 등에서 쉽게 XML 보안 기술을 이용할 수 있게 되면 현재의 예상을 능가하는 시장성을 확보할 수 있을 것이다.

5. 결 론

지금까지 XML을 기반으로 무결성, 기밀성, 인증, 접근 통제 등의 보안 요구사항을 지원할 수 있는 XML 보안 기술에 대해서 살펴보고 W3C와 OASIS를 중심으로 진행 중인 표준화 현황을 정리하였다.

현재 W3C에서 권고안으로 승인된 XML 전자서명과 XML 암호화는 가장 널리 사용되는 기술이며 다수의 회사에서 보안 솔루션으로 이미 개발하였거나 개발하고 있다. 반면에 XML 키 관리 명세인 XKMS는 아직 초안 상태로써 일부 회사의 구현 간에는 호환성을 보장하기 어려운 현실이다(물론 XML 암호화의 호환성 확보도 여전히 의문사항이다). OASIS에서 XML 기반의 인증과 권한부여를 제공하기 위한 표준안으로 채택된 SAML과 XACML은 웹 서비스 보안 표준으로 통합되는 과정이므로 .NET이나 J2EE와 같은 웹 서비스 플랫폼의 구현 단계에서 호환성을 보장할 수 있을 것으로 보인다.

XML 보안 표준이 제공하는 확장성과 호환성 및 시장성은 XML 적용 분야의 급속한 확장을 가져오고 보안 기술의 손쉬운 적용을 보장하므로 미래의 정보 시스템이 갖는 중요한 자산을 보호하는 기반이 될 것이다.

참 고 문 헌

[1] W3C - eXtensible Markup Language(XML), http://www.w3.org/XML/

[2] W3C Homepage, http://www.w3c.org

[3] W3C - XML Schema, http://www.w3.org/XML/Schema

[4] W3C - XML Namespace, http://www.w3.org/TR/REC-xml-names/

[5] W3C - XML Path Language(XPath) 2.0, http://www.w3.org/TR/xpath20/, 2002

[6] W3C - XSL Transformations(XSLT) Version 1.0, http://www.w3.org/TR/xslt, 1999

[7] W3C - XQuery 1.0:An XMl Query Language, http://www.w3.org/TR/xquery/, 2002

[8] W3C - XML Pointer Language (Xpointer), http://www.w3.org/TR/xptr/, 2002

[9] W3C - XForms-The Next Generation of Web Forms, http://www.w3.org/MarkUp/Forms/

[10] W3C - Simple Object Access Protocol (SOAP) 1.1 W3C Note 08 May 2000, http://www.w3.org/TR/SOAP/

[11] OASIS - UDDI Specification TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec

[12] W3C - Web Services Description Language (WSDL) Version 1.2 W3C Working Draft 3 March 2003, http://www.w3.org/TR/wsdl12/

[13] W3C - Web Services Activity, http://www.w3.org/2002/ws/

[14] ebXML Homepage, http://www.ebxml.org/

[15] D. Eastlake 3rd, et.al. XML-Signature Syntax and Processing, IETF RFC3275, http://www.ietf.org/rfc/rfc3275.txt, March 2002

[16] T. Imamura, et al., XML Encryption Syntax and Processing W3C Recommendation, http://www.w3.org/TR/xmlenc-core/, 04 March 2002

[17] W3C - XML Key Management Specification (XKMS 2.0) W3C Working Draft, http://www .w3.org/TR/xkms2/, 18 March 2002

[18] OASIS Homepage, http://www.oasis-ope.org/home/index.php

[19] OASIS - Security Assertion Markup Language (SAML),http://www.oasis-open.org/committees/

[20] OASIS - eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/

[21] A. O. Freier, et al., Netscape Communications, SSL 3.0 Specification, Draft302, http://wp.netscape.com/eng/ssl3/draft302.txt, 1996

[22] T. Dierks, et al., The TLS Protocol Version 1.0, IETF RFC2246, ftp://ftp.rfc-editor.org/ in-notes/rfc2246.txt, 1999

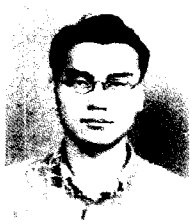
[23] B. Ramsdell, Brute Squad Labs, S/MIME Version 3.1 Message Specification, IETF RFC2633, http://www.ietf.org/internet-drafts/draft-ietf-smime-rfc2633bis-03.txt , 2003

[24] R. Housley, et al., Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC2459, ftp://ftp.rfc-editor.org/in-notes/rfc2459.txt, 1999

[25] W. M. Daley, R. G. Kammer, Digital Signature

Standard (DSS), U.S. DoC/NIST, FIPS PUB 186-2, http://csrc.nist.gov/cryptval/dss.htm
[26] P. Karn, et al., The ESP Triple DES Transform, RFC1851, http://www.watersprings.org/pub/rfc/rfc1851.txt, 1995
[27] B. Kaliski, et al., PKCS #1: RSA Cryptography Specifications Version 2.0, RFC2437, ftp://ftp.rfc-editor.org/in-notes/rfc2437.txt, 1998
[28] ECIF Homepage, http://www.ecif.or.kr/

◐ 저 자 소 개 ◑



박 대 하
1992 고려대학교 컴퓨터학과 학사
1994 고려대학교 일반대학원 컴퓨터학과 석사
1996 고려대학교 일반대학원 컴퓨터학과 박사과정 수료
1999~현재 : (주)시큐리티테크놀로지스 책임연구원
관심분야: XML 보안, 보안 프로토콜, 이동코드 보안, 임베디드 시스템 보안

문 창 주
1997년 고려대학교 컴퓨터학과 학사
1999년 고려대학교 컴퓨터학과 석사
2000년~현재 : 고려대학교 컴퓨터학과 박사과정
관심분야 : 컴포넌트, 보안공학, RBAC

김 영 갑
2001년 고려대학교 식량자원학과 학사
2001년~현재 : 고려대학교 컴퓨터학과 석사과정
관심분야 : 보안 프로토콜,이동코드 보안, 보안 명세

백 두 권
1974년 고려대학교 수학과(학사)
1977년 고려대학교 대학원 산업공학과(석사)
1983년 Wayne State Univ. 전산학과(석사)
1985년 Wayne State Univ. 전산학과(박사)
1986년~현재 : 고려대학교 컴퓨터학과 교수
1989년~현재 : 한국 정보과학회 평의원/이사
1991년~현재 : 한국 시뮬레이션 학회 이사/부회장
1991년~현재 : ISO/IEC JTC1/SC32 국내위원회 위원장
1999년~현재 : 정보통신진흥협회 데이터 기술위원회 의장
2002년~현재 : 고려대학교 정보통신대학 학장
관심분야 : 데이터베이스, 소프트웨어 공학, 데이터 공학, 컴포넌트 기반 시스템, 메타데이터 레지스트리, 정보 통합