웹 서비스 메시지 보안을 위한 표준 기술 연구

차석일°, 김현희, 이형석, 신동규
세종대학교 컴퓨터공학과

Study of Standard Technology for Web Service Message Security

Cha, Suk-Il Kim, Hyun-Hee Lee, Hyung-Suk Shin, Dong-Kyoo
Sejong University
E-mail : {kiry, hyunhee, bestehen}@gce.sejong.ac.kr, shindk@sejong.ac.kr

요 약

웹을 이용한 서비스는 위와 같은 여러 장점을 가지고 있지만 각종 데이터 및 문서가 웹 상에 존재하므로 가상공간에서의 문서의 처리가 위조나 변경이 가능하다. 이러한 웹 상에서의 전송 시 발생할 수 있는 수많은 역기능들을 줄일 수 있는 가장 강력한 방법은 암호 응용 기술을 전자상거래 시스템 구축에 사용함으로써, 기밀성(confidentiality), 무결성(integrity), 인증(authentication) 등의 보안 서비스를 제공하는 것이다. 이에 본 논문에서는 현재 진행중인 표준화 단체의 동향을 파악하고 WS-Security 명세서를 통해 웹 서비스 보안의 전반적인 기술을 분석한다.

1. 서론

차세대 e-Business의 기반은 웹 서비스로 구축될 것이라는 전망이 제기되어 온 가운데, 전 세계적으로 MS, IBM, Sun Microsystems 등과 같은 대형 소프트웨어 벤더들간에 웹 서비스 관련 인프라 소프트웨어 및 개발 툴 부분에서 경쟁이 치열하게 전개되고 있다. 그러나 현재 대부분의 XML 기반 e-Business 프레임워크의 연구 및 지원은 개별적인 구성 컴포넌트의 세부 설계 기법 및 구현 방법 개발에만 집중되어 있다. 과거와는 달리 국내의 각종 IT 분야에서 보안 요소가 부가적인 요구사항이 아닌 핵심 개발 요소라는 인식이 확대되어 가고 있다 [1].

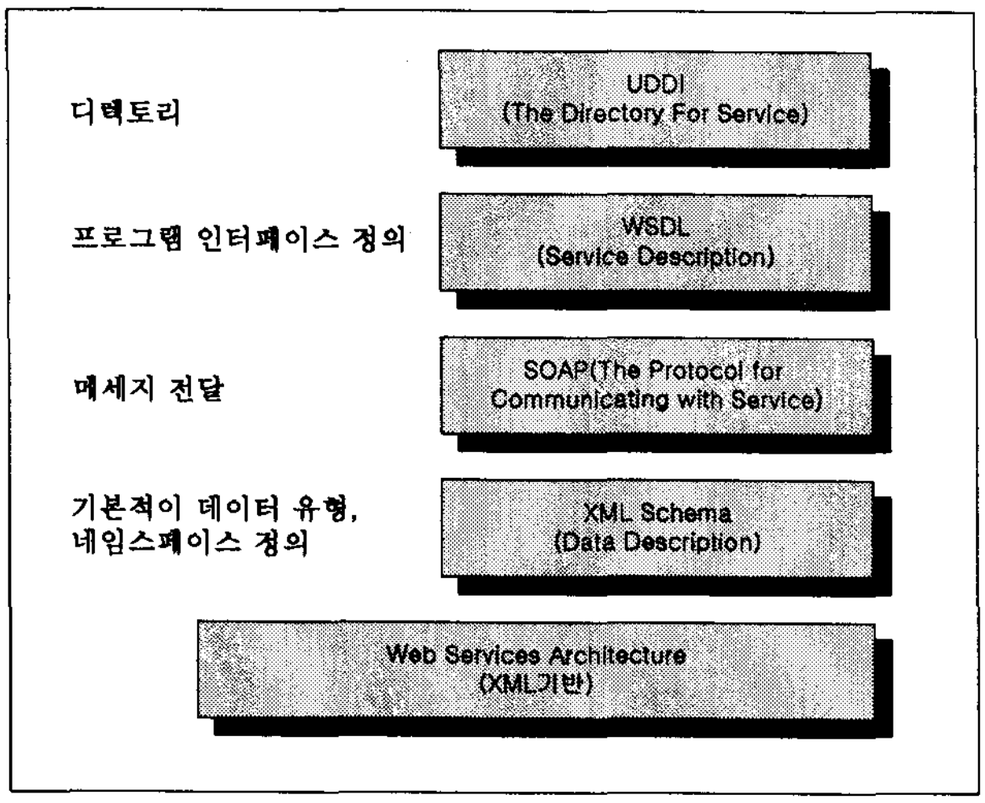
인터넷 보안기술 중 공개키 암호화 시스템을 이용해 향상된 보안 수준을 제공하기 위한 기반 기술인 PKI(Public Key Infra-structure)는 각종 응용 시스템 및 전자상거래 기반 기술로서 현재 가장 중요한 기술로 인식되고 사용되고 있다. 본 논문에서는 XML기반 차세대 PKI기술인 XKMS(XML Key Management Service) 보안 모듈 설계를 통해 웹 서비스에 보다 안전한 보안 서비스를 제공하기 위한 방안을 연구한다.

2. 관련 연구

2.1 웹서비스의 기본 구조

현재 W3C가 추진 중인 웹서비스 표준 규약에서 웹서비스의 아키텍처를 구성하고 있는 기본적인 표준들은 XML(Extensible Markup Language), UDDI(Universal Description,Discovery and Integration) WSDL(Web Service Description Language), SOAP(Simple Object Access Protocol) 등이 있다. XML은 인터넷을 통해 교환되는 데이터 표준 언어로서 오픈 프레임워크인 웹서비스의 기반 구조를 이루고 있다. XML 스키마(Schema)는 웹서비스의 기본적인 데이터 유형을 정의하는 역할을 한다. XML 스키마는 일종의 데이터 사전으로서 각 객체의 개념을 정의하고 객체

들간의 연관 관계를 정의하고 각 데이터에 의미를 부여하여 이질적인 데이터의 상호 호환을 가능하게 해 준다. UDDI는 웹서비스의 디렉토리 서비스를 담당하게 되는데 업체가 자사의 웹서비스를 온라인 디렉토리에 등록·광고하거나 외부에서 웹서비스를 검색하는데 사용된다. WSDL은 웹서비스의



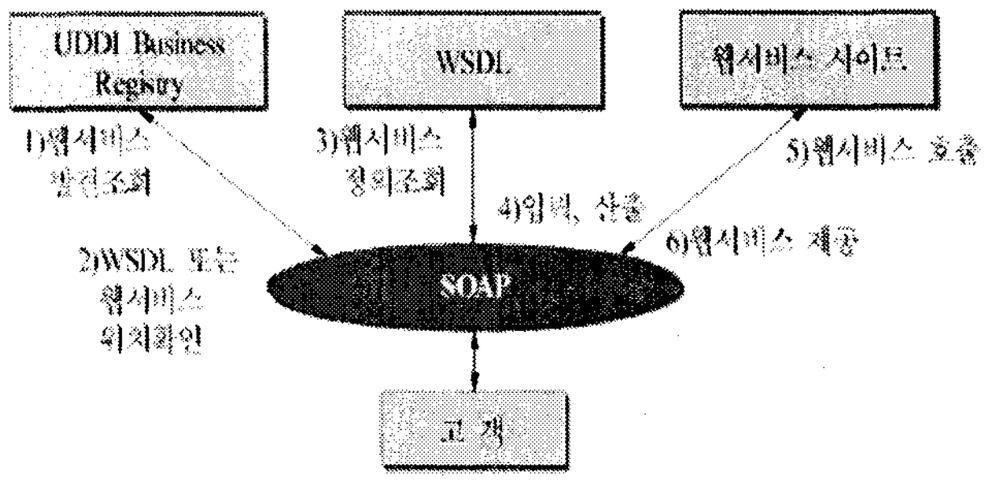
<그림 1> 웹 서비스의 기본 구조

서비스를 정의하는 언어로서 프로그램이나 인터페이스 정의 등 소프트웨어 업체가 웹서비스를 기술할 때 사용된다. SOAP은 분산된 환경의 정보를 교환하는 통신프로토콜로서 인터넷을 통해 다양한 웹서비스 사용자가 정보를 교환할 수 있는 통신의 역할을 담당하고 있다. 현재 UDDI, WSDL, SOAP은 MS와 IBM 주도로 어느 정도 표준이 제정된 상태이고, WSDL(Web Services Description Language)은 호환성 문제가 논의되고 있다.[3]

2.2 웹서비스 프레임워크

일반적인 웹서비스의 프레임워크 프로세스는 우선 고객이 SOAP이란 통신언어를 통해 UDDI내에 있는 웹서비스업체를 조회하고 WSDL 또는 웹서비스 위치를 확인한 후 웹서비스의 정의를 조회하고 고객이 자신의 필요한 정의를 입력 또는 산출한 후에 웹서비스 업체를 호출하면 웹서비스를 제공받게 된다.

웹서비스 프레임워크는 XML을 기반으로 해서 모든 정보를 등록소에 저장해 두고 상호간의 정보교환이 이루어지며, 통신언어로 SOAP을 사용한다는 점에서 ebXML 프레임워크와 유사한 점이 많다. 반대로 두 프레임워크간의 차이점을 살펴보면 웹서비스의 경우 서비스 정의는 WSDL, 디렉토리 서비스는 UDDI를 사용하는 것에 반해, ebXML의 경우 서비스 정의는 CPP(Collaboration Protocol Profile), 디렉토리는 ebXML 등록소를 따로 개발하여 운영한다는 점이다. 물론 이와 같은 차이점들



<그림 2> 웹서비스 프레임워크

은 개발업체 및 단체간의 협의에 의해서 얼마든지 호환성을 유지할 가능성이 크다. 하지만 현재 개발되고 있는 다수의 웹서비스 플랫폼의 경우 기본 요소 및 프로세스는 거의 유사하지만 플랫폼 개발 업체마다 상세 기술 및 제공되는 기능이 상이하기 때문에 개발된 플랫폼간의 상호운영성 문제3)가 크게 대두되었으며, 이러한 문제를 해결하기 위한 방안이 모색 중이다.

3. 웹 서비스 보안 및 WS-Security 분석

2002년 7월에 마이크로소프트, IBM, 베리사인은 WS-Security 명세를 공개된 표준화를 위해 OASIS에 제출된 상태이다. OASIS[4]에서는 Web Services Security TC를 구성해 표준화를 진행하고 있다.

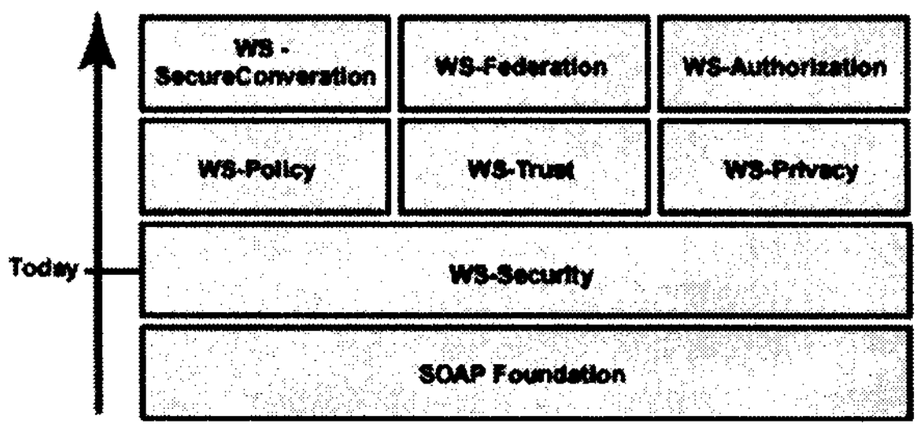
e-Business는 인터넷과 같은 네트워크를 통해 이루어지는 교역 파트너 사이에 정보 교환에 의존한다. 항상 보안상의 위험이

WS-Security 명세는 위의 3업체가 추진 중인 웹 서비스 보안 로드맵의 일부로서 현재 추가적인 명세 작업을 진행 중이다. <그림 3>은 향후 발표될 보안 명세를 포함하는 웹 서비스 보안 전체 구성이다[5][6].

위의 보안 명세 중 위쪽의 6개의 보안 명세는 두

개의 부류로 나눠진다. 첫번째는 WS-Policy, WS-Trust, WS-Privacy로 구성되며, 정보가 인증됐는지, 그리고 상대방과 정보를 어떻게 공유할지 이해가 되었는지 여부를 담당한다. WS-Policy는 기업들이 받은 메시지가 자사가 필요로 하는 보안 수준을 만족시킬 때만 구매 주문을 받아들일 수 있도록 해준다. WS-Privacy는 전자상거래 웹사이트로 보내진 개인 정보에 대한 비밀성의 보장에 관련된 것이다.

두번째 부류는 WS-SecureConversation, WS-Federation, WS-Authorization이다.



<그림 3> 웹 서비스 보안 명세 구성

WS-SecureConversation은 웹 서비스가 요청자 메시지의 신원을 어떻게 확인하고 요청자가 서비스의 신원을 어떻게 확인하며 상호 신원 확인된 보안 문맥을 어떻게 구축하는지를 설명하는 것이고, WS-Federation은 WS-Security, WS-Policy, WS-Trust 및 WS-SecureConversation을 연합된 신임 시나리오를 구축하는 방법을 정의한 것이다. WS-Authorization은 웹 서비스에 대한 접근정책이 어떻게 지정되고 관리되는지를 설명한다. 이것은 고객이 어떻게 상이한 보안 기술을 사용하는 컴퓨팅 시스템에 접속하는지를 다룬다. 고객이 안전하게 통신할 수 있는 환경을 조성하며 한 기업에서 확인된 사람이 다른 기업에서도 확인 받을 수 있는 방법을 제공한다.

위의 명세 집합 중 WS-Security, WS-Policy, WS-Trust, WS-Privacy와 같은 명세들이 보안 기본 요소를 구성하게 될 것이고 이러한 초기 보안 명세가 완료되면 통합 보안 모델을 구성하는데 필요한 기술 요소를 위해 WS-SecureConversation, WS-Federation, WS-Authorization 등의 명세들이 추가 개발될 예정이다.

WS-Security 명세에서는 SOAP 헤더 내에 <Security> 엘리먼트를 새롭게 정의해 수신측에서 요구하는 보안 정보를 포함시킨다. <Security> 엘리먼트는 <UsernameToken> 엘리먼트를 통해 메시지 송신자를 식별하며 <Signature> 엘리먼트로 XML 전자서명을 검증하는데 필요한 서명 정보를 포함하도록 규정하고 있다.

다음 <예제 1>은 사용자명 보안 토큰을 가진 메시지이다.

```
(001) <?xml version="1.0" encodig='utf-8"?>
(002) <S:Envelope xmlns:S ="http://www.w3.org/2001/12/soap-envelpe"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
(003)   <S:Header>
(004)     <m:path xmlns:m="http://schemas.xmlsoap.org/p/">
(005)      <m:action>http://fabrikam123.com/getQuote</m:action>
(006)      <m:to>http://fabrikam123.com/stocks</m:to>
(007)      <m:id>uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</m:id>
(008)     </m:path>
(009)     <wsse:S ecurity
          xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
(010)       wsse:UsernameTokerId="MyID">
(011)         <wsse:Username>Zoe</wsse:Username>
(012)       </wsse:UsernameToken>
(013)       <ds:Signature>
(014)         <ds:SignedInfo>
(015)           <ds:CanonicalizationMethod
                Algorithm=
                  "http://www.w3.org/2001/10/xml-exc-c14n#"/>
(016)           <ds:SignatureMethod
                    Algorithm=
                  "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
(017)           <ds:Reference URI="#MsgBody">
(018)             <ds:DigestMethod
                      Algorithm=
                  "http://www.w3.org/2000/09/xmldsig#sha1"/>
(019)             <ds:DigestValue>LyLsF0P l4wPU...</ds:DigestValue>
(020)           </ds:Reference>
(021)         </ds:SignedInfo>
(022)         <ds:SignatureValue>DJbchm5gK...</ds:SignatureValue>
(023)         <ds:KeyInfo>
(024)           <wsse:S ecurityTokenReference>
(025)             <wsse:R eferenceURI ="#MyID"/>
(026)           </wsse:S ecurityTokenReference>
(027)         </ds:KeyInfo>
(028)       </ds:Signature>
(029)     </wsse:S ecurity>
(030)   </S:Header>
(031)   <S:Body Id="MsgBody">
(032)    <tru:StockSymbol xmlns:tru="http://fabrikam123.com/payloads">
           QQQ
         </tru:StockSymbol>
(033)   </S:Body>
(034) </S:Envelope>
```

<예제 1>

첫 번째 두 행은 SOAP 메시지 네임스페이스 정보이다. (003) 행은 이 SOAP 메시지와 연관되어 있는 헤더로 시작한다. (004)에서 (008) 행은 이 메시지를 어떻게 전송하는지를 지정한다.

(009)행은 명세에서 새롭게 정의한 <Security> 엘리먼트로서 의도된 수신자를 위한 보안 정보를 포함하고 있다.

(010)에서 (012) 행은 메시지와 연관되어 있는 보안 토큰을 지정한다. 이 경우에는 <UsernameToken>을 사용하여 클라이언트의 사용자명을 지정한다. 여기서 서비스가 패스워드를 알고 있다고 가정한다.

(013)행에서 (028) 행은 전자서명을 지정한다. 이 서명은 서명된 엘리먼트의 무결성을 보증한다. 서명은 XML 전자서명 명세를 이용한다. 이 예제에서 서명은 사용자 패스워드에서 생성된 키에 기반하고 있다.

(014) 행에서 (021) 행은 전자서명 및 관련 정보를 나타낸다. (015) 행은 설명되고 있는 데이터를 어떻게 정규화 할 것인지를 지정한다.

(017) 행에서 (020) 행은 서명된 엘리먼트를 선택한다. 구체적으로, (017) 행은 <S:Body> 엘리먼트가 서명되었음을 가리킨다. 이 예제에서는 메시지 본문만 서명되었다.

(022) 행은 XML 서명 명세에서 정의한대로 서명되고 있는 정규화된 형태의 서명 값을 지정한다.

(023) 행에서 (027) 행은 이 서명과 관련된 보안 토큰을 어디에서 찾을 것인지에 대한 힌트를 제공한다. 구체적으로, (024)~(025) 행은 보안 토큰이 지정된 URL에서 발견될 수 있음을 가리킨다.

(031) 행과 (033)행은 SOAP 메시지의 본문을 포함하고 있다[7].

다음 예제는 보안 토큰, 서명, 암호화 사용을 보여준다. 이 예제에서 우리는 메시지 본문과 함께 불변적인 라우팅 헤더를 선택하는 가상의 "Routingransform"을 사용한다.

이 예제의 몇 가지 핵심 부분을 설명하면 다음과 같다.

(003)-(051) 행은 SOAP 메시지 헤더를 포함하고 있다.

(004)-(009) 행은 메시지 라우팅 정보를 지정한다. 이 경우 우리는 getQuote" 액션을 요구하는 메시지를 http://fabrikam123.com/stocks 서비스에 보내고 있다.

(010)-(050) 행은 <Security> 헤더 블록을 나타낸다. 이것은 메시지에 대한 보안 관련 정보를 가지고 있다.

(011)-(013) 행은 메시지와 연관된 보안 토큰을 지정하고 있다. 이 경우 Base64로 인코딩된 X.509 인증을 지정한다. (012) 행은 인증의 실제 Base64 인코딩을 지정하고 있다.

(014)-(026) 행은 메시지 본문을 암호화하기 위해 사용되는 키를 지정한다. 이것은 대칭 키이기 때문에 암호화된 형태로 전달된다. (015) 행은 키를 암호화하기 위해 사용되는 알고리즘을 정의한다. (016)-(018) 행은 대칭형 키를 암호화하기 위해 사용되는 키의 이름을 지정한다. (019)-(022) 행은 대칭형 키의 실제 암호화된 형태를 지정한다. (023)-(025) 행은 이 대칭형 키를 사용하는 메시지의 암호화 블록을 확인한다. 이 경우에는 본문을 암호화하기 위해서만 사용되었다. (Id="enc1")

(027)-(049) 행은 전자서명을 지정한다. 이 예제에서 서명은 X.509 인증에 기반하고 있다. (028)-(040) 행은 무엇이 서명되고 있는지를 가리킨다. 구체적으로, (029) 행은 정규화 알고리즘을 가리킨다. (030) 행은 서명 알고리즘을 가리킨다.

```
(001) <?xml version="1.0" encoding="utf-8"?>
(002) <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
(003)   <S:Header>
(004)     <m:path xmlns:m="http://schemas.xmlsoap.org/rp/">
(005)       <m:action>http://fabrikam123.com/getQuote</m:action>
(006)       <m:to>http://fabrikam123.com/stocks</m:to>
(007)       <m:from>mailto:johnsmith@fabrikam123.com</m:from>
(008)       <m:id>uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</m:id>
(009)     </m:path>
(010)     <wsse:Security>
(011)       <wsse:BinarySecurityToken
              ValueType="wsse:X509v3"
              Id="X509Token"
              EncodingType="wsse:Base64Binary">
(012)       MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
(013)       </wsse:BinarySecurityToken>
(014)       <xenc:EncryptedKey>
(015)         <xenc:EncryptionMethod Algorithm=
                "http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
(016)         <ds:KeyInfo>
(017)           <ds:KeyName>CN=Hiroshi Maruyama, C=JP</ds:KeyName>
(018)         </ds:KeyInfo>
(019)         <xenc:CipherData>
(020)           <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(021)           </xenc:CipherValue>
(022)         </xenc:CipherData>
(023)         <xenc:ReferenceList>
(024)           <xenc:DataReference URI="#enc1"/>
(025)         </xenc:ReferenceList>
(026)       </xenc:EncryptedKey>
(027)       <ds:Signature>
(028)         <ds:SignedInfo>
(029)           <ds:CanonicalizationMethod
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(030)           <ds:SignatureMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
(031)           <ds:Reference>
(032)             <ds:Transforms>
(033)               <ds:Transform
                    Algorithm="http://...#RoutingTransform"/>
(034)               <ds:Transform
                    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(035)             </ds:Transforms>
(036)             <ds:DigestMethod
                  Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(037)             <ds:DigestValue>Ly+sF094hP4wPU...
(038)             </ds:DigestValue>
(039)           </ds:Reference>
(040)         </ds:SignedInfo>

(041)         <ds:SignatureValue>
(042)           Hp1ZkmFZ/2kQLXDJbchm5gK...
(043)         </ds:SignatureValue>
(044)         <ds:KeyInfo>
(045)           <wsse:SecurityTokenReference>
(046)             <wsse:Reference URI="#X509Token"/>
(047)           </wsse:SecurityTokenReference>
(048)         </ds:KeyInfo>
(049)       </ds:Signature>
(050)     </wsse:Security>
(051)   </S:Header>
(052)   <S:Body>
(053)     <xenc:EncryptedData
            Type="http://www.w3.org/2001/04/xmlenc#Element"
            Id="enc1">
(054)       <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#3des-cbc"/>
(055)       <xenc:CipherData>
(056)         <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(057)         </xenc:CipherValue>
(058)       </xenc:CipherData>
(059)     </xenc:EncryptedData>
(060)   </S:Body>
(061) </S:Envelope>
```

<예제 2>

(031)-(039) 행은 서명되고 있는 메시지의 부분을 확인한다. 구체적으로, (033) 행은 변환을 확인한다. 이 가상의 변환은 라우팅 헤더와 메시지 본문의 불변적인 부분을 선택한다. (034) 행은 (033) 행에서 선택된 메시지 부분을 사용하기 위한 정규화 알고리즘을 지정한다. (036) 행은 규범화된 데이터에서 사용되는 다이제스트 알고리즘을 가리킨다. (037) 행은 규범화된 데이터의 지정된 알고리즘에서 나오는 다이제스트 값을 지정한다.

(041)-(043) 행은 실제 서명 값을 가리킨다.

(044)-(048) 행은 이 서명에 사용된 키를 가리킨다. 이 경우 이것은 메시지에 포함된 X.509 인증서이다. (046) 행은 (011)-(013) 행으로의 URI 링크를 제공한다.

메시지 본문은 (052)-(060) 행에 의해 표현된다.

(053)-(059) 행은 XML 암호화를 사용하여 암호화된 메타 데이터와 본문 형식을 표현한다. (053) 행은 엘리먼트 값이 교체되고 있음을 가리키고 이 암호화를 확인한다. (054) 행은 암호화 알고리즘을 지정한다. (055)-(058) 행은 실제 암호 텍스트 (즉 암호화의 결과)를 포함한다. 키가 이 암호를 참조하기 때문에 키에 대한 참조는 포함시키지 않는다.

4. 결론

본 논문에서는 웹 서비스의 보안에 관련된 표준화 단체의 동향을 파악하고 WS-Security 명세서의 분석을 통해 웹 서비스의 전반적인 보안 기술 및 내부적인 구조를 분석하였다. 웹 서비스가 보다 광범위하게 적용되고 방화벽, 부하 조정자(load balancers), 메시징 허브와 같은 중개자를 지원하기 위한 애플리케이션 토폴리지가 계속 발전하고 또한 기업이 직면하는 위협에 대한 인식이 더 잘 이해됨에 따라, 웹 서비스에 대한 추가적인 보안 사양에 대한 필요가 더욱 분명해지고 있다.

현재, 인터넷은 믿을 수 있는 장소가 아니며, 몇몇의 웹 서비스를 위해 보안제품과 기본적인 골격이 보안구조로 형성되었다. 웹서비스 보안에서의 입증, 메시지 기밀성, 서명 그리고 무결성과 같은 중요한 문제에 대한 다각도의 연구를 활발히 진행해야하겠다.

5. 참고문헌

[1] Simple Object Access Protocol(SOAP), http://www.w3.org/TR/2002/WD-soap12-part1-20020626

[2] Web Services Description Language (WSDL), http://www.w3.org/TR/2002/WD-wsdl12-20020709

[3] 정부연 웹서비스의 개념과 관련 기업에 미치는 영향, 정보통신정책 제14권 7호, 2002. 4.

[4] OASIS Web Services Security TC, http://www.oasis-open.org/committees/wss/

[5] Eduardo B. Fernandez, Web Services Security Current status and the future, http://www.webservicesarchitect.com/content/articles/fernandez01.asp

[6] 웹 서비스 세계에서의 보안, 아키텍처 및 로드맵 제안 http://www-903.ibm.com/developerworks/kr/webservices/library/ws-secmap.html#1

[7] Web Services Security (WS-Security) Version 1.0 05, 2002년 4월, http://www-903.ibm.com/developerworks/kr/webservices/library/ws-secure.html

[8] 변광준 "웹 서비스 기술과 전망," 한경 Enterprise IT Directions Track E, 2002. 4.