전자서명을 이용한 XML 웹서비스의 메시지계층 보안

홍성표*, 김형균*, 이정기*, 이 준**
* 조선대학교 대학원 컴퓨터공학과
**조선대학교 전자정보공과대학 컴퓨터공학과
e-mail:hony1128@hanmail.net

A Secure to Message Level of XML Web service using Digital Signature

Seong-Pyo Hong*, Hyeong-Gyun Kim*, Jeong-Ki Lee*,
Joon Lee**
*Dept of Computer Engineering, Graduate School, Chosun University
**Dept of Computer Engineering, Chosun University

요 약

웹서비스의 보안정책은 전송계층과 어플리케이션 계층 두 부분에 적용되고 있으며, 분산 어플리케이션간 통신을 할 경우 데이터에 대한 암호화와 인증처리가 가장 중요한 부분이 될 것이다.

본 논문에서 제안한 시스템은 웹서비스 작업에서 클라이언트의 요청을 XML 서명 후 전송하여 요청을 받은 서버측에서 응답을 하기 전 정당한 클라이언트인가를 검증하는 단계를 거친다. 이렇게 함으로써 웹서비스 환경에서의 SOAP을 이용한 메시지계층 보안을 한층 강화시키려는 목적이다.

1. 서론

분산환경에서 플랫폼이나 프로그램언어에 독립적인 구현과 어플리케이션간의 통신을 할 수 있는 웹서비스는 프로그래밍 업계에서 최근 가장 주목받는 분야로 플랫폼과 언어에 중립적인 프로토콜인 HTTP나 XML을 사용함으로써 클라이언트에게 전체 시스템구현을 숨길 수 있어 기존 기술과 차별화된 서비스를 웹상에서 가능하게 한다. 하지만 웹서비스는 아직 기술적으로 성장기에 있는 기술이므로 수행성능, 확장성, 신뢰성, 보안 등 보완되어야 할 부분들이 아직 존재한다. 그 중에서도 개발자들이 가장 염려하고 있는 것 중의 하나가 Seb Service 시스템 상에서 발생할 수 있는 비 인가된 권한사용, 서비스 거부, 데이터 노출/변경, 송수신 부인 등의 보안과 관련된 문제들이다.

보안은 독립적인 어플리케이션을 구현할 때에도 중요한 부분이었으나 어플리케이션이 여러 네트워크에 분산되면서, 동적이고 개방된 웹 환경에서 서비스를 찾아내고 실행할 수 있는 새로운 모델이 가미되어 그 중요성이 더욱 강조되고 있다.[9-10]

이에 MS와 IBM, 로터스 등의 기존의 인프라를 그대로 활용하면서 보안상의 문제를 야기하지 않고 원격 어플리케이션을 원활하게 지원하는 새로운 프로토콜 기술로서 이른바 SOAP[2,3]을 공동으로 개발하였다.

특히 SOAP 1.1에서부터 지원되는 확장 모듈은 SOPA 메시지 자체에 보안기능을 둘 수 있는 메커니즘을 제공한다. SOAP Envelope의 Header 요소에 디지털 서명 정보를 담아갈 수 있도록 하였다. 여기서는 XML 암호화 같은 다른 보안 기능들도 조합해서 쓸 수 있도록 하였다. Body 요소에 캡슐화된 정보를 디지털 서명하여 Header 요소에 첨부시켜 보내면 서버측에서는 클라이언트의 요청을 받아서 서명을 검증한 후 결과를 돌려준다.[3,4]

본 논문에서 제안한 시스템은 웹서비스 작업에서 클라이언트의 요청을 XML 서명 후 전송하여 요청을 받은 서버측에서 응답을 하기 전 정당한 클라이언트인가를 검증하는 단계를 거친다. 이렇게 함으로써 웹서비스 환경에서의 SOAP을 이용한 메시지계층 보안을 한층 강화시키려는 목적이다.

2. 관련 기술

2.1 XML 서명

W3C에서 표준으로 권고된 XML 디지털 서명기술[4-6]은 기존의 디지털 서명을 XML을 이용하여 표현한 것으로, 기존의 디지털 서명과 같이 전체 문서

에 대해서도 서명할 수 있고, 또 XML Transform 기술[6, 8]을 이용하여 서명이 필요한 문서의 일부분에 대해서도 서명할 수 있어, 기존 문서의 재사용성을 높일 수 있다. 표 1은 XML 디지털 서명에 관련된 태그 집합들을 보여준다.

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

표 1. XML 서명관련 태그

디지털 서명은 전자화된 문서의 메시지 내용이 수정 및 변조되지 않았음을 보장하는 동시에 메시지의 주체인 사용자들의 신원을 제 3자에게 확인할 수 있게끔 하는 인증방식을 말한다. W3C에서 표준으로 권고된 XML 디지털 서명 기술은 기존의 디지털 서명을 XML을 이용하여 표현한 것으로, 기존의 디지털 서명과 같이 전체 문서에 대해서도 서명을 할 수 있고, 또 XML Transform 기술을 이용하여 서명이 필요한 문서의 일부분에 대해서도 서명 할 수 있어, 기존 문서의 재사용성을 높일 수 있다.[4-6, 8]

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
      ..
      <Transforms>
        <Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-200103
15"/>
      </Transforms>
      <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>5WLICryAcs9ZylBM3lZ5MGsliM0=</DigestValu
e>
    </Reference>
  </SignedInfo>
  <SignatureValue>
PeMYtziu40CnKo0DOBC+tBxoY8pSS9fGUeaAjCQYRttc5yCkbh
93EA==
  </SignatureValue>
```

표 2. XML 서명문서의 예

2.1.1 서명문서 생성

XML 디지털 서명문서의 생성에는 크게 두 가지 생성 절차가 있는데, 첫 번째는 Reference 생성이고, 두 번째는 서명 생성이다. Reference 생성은 사용자 문서에 여러 가지 Transform을 적용하고, Transform 된 문서에 대해 해쉬 값을 계산한다.

서명 생성은 위에서 생성한 Reference 부분을 포함한 서명정보 부분, 즉 SignedInfo 엘리먼트 영역을 사용자의 개인키를 이용해 서명값을 계산하는 과정이다. 여기서 서명전에 서명할 부분에 대한 무결성을 위해 Canonicalization을 수행한다. 그림 1은 XML 서명문서 생성과정을 나타낸다.

생성된 서명 문서는 enveloped, enveloping, detached 세 가지 형태로 구분된다. 서명될 문서 안에 Signature 엘리먼트가 포함되어 있으면 enveloped 서명이고, 서명될 문서가 Signature 엘리먼트 안에 포함되어 있으면 enveloping 서명이며, Signature 엘리먼트와 서명될 문서가 한 XML 문서 안에 없으면, detached 서명 형태이다.

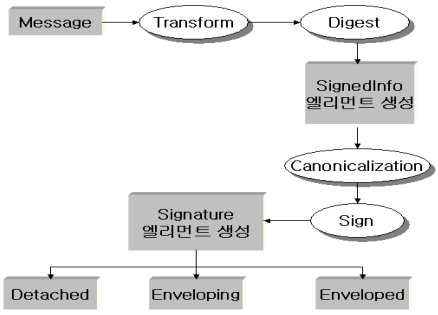

그림 1. XML 서명문서 생성

2.1.2 서명문서 검증

XML 디지털 서명 문서의 검증 역시 Reference 검증과 서명 검증 두 부분으로 나뉘어지는데 두 가지 검증 절차는 다음과 같다.

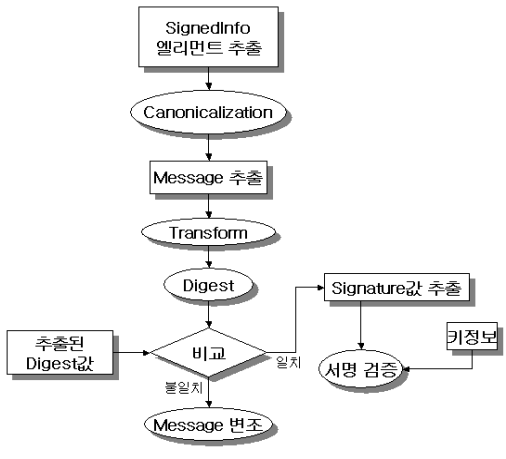

그림 2. XML 서명문서 검증

Reference 검증절차는 수신된 서명 문서의 SignInfo 엘리먼트를 추출하여 이 부분에 대한 무결성을 위해 Canonicalization을 수행하고, Reference 생성과정과 동일하게 사용자 문서에 여러 가지

Transform을 적용한 다음 해쉬 값을 계산한다. 여기서, 사용자 문서는 Reference 엘리먼트의 URL에서 얻을 수 있다. 이렇게 검증시 계산한 해쉬 값과 서명문에 포함된 해쉬 값을 비교하여 그 값들이 동일할 경우 Reference 검증은 성공하게 된다.

서명 검증 절차는 사용자의 공개키에 해당하는 정보를 얻는다. 이 공개키로 서명 값을 검증하게 된다. 검증 결과가 유효하다면 수신 받은 디지털 서명문서는 유효하다고 판단한다. 그림 2는 XML 서명문서 검증 과정을 나타낸 것이다.

2.2 SOAP(Simple Object Access protocol)

SOAP은 분산환경에서의 정보교환에 사용되는 경량(Lightweight) 분산 컴퓨팅 프로토콜이다. SOAP에서는 텍스트기반(text_based) XML을 프로토콜로 사용한다. 텍스트기반의 XML이라는 점 덕분에 SOAP은 하드웨어 플랫폼, 운영체제, 프로그래밍 언어, 그리고 네트워크 하드웨어 플랫폼 전 영역에 걸쳐서 상호운용성이 매우높다.

SOAP은 하드웨어 플랫폼, 운영체제, 프로그래밍 언어, 네트워크 하드웨어 플랫폼에 종속되지 않는다. 다른 분산컴퓨팅 시스템과는 달리 SOAP은 HTTP, XML과 개방형 표준안을 근간으로 만들어졌다.

SOAP 명세는 크게 메시징(Messaging)과 RPC(Remote Procedure Call) 두 부분으로 구성된다. 메시징 부분은 분산 시스템간에 주고받는 메시지 구조를 정의한다. RPC 부분은 메시지안에 원격 프로시져 호출/응답을 포함(Embeded)시키는 방법을 정의한다.

SOAP은 HTTP를 사용할 수 있기 때문에 기존의 웹서버, 프록시 서버, 방화벽과 같은 인프라를 그대로 이용할 수 도 있다. SOAP은 HTTP 외에도 SMTP나 JMS와 같은 프로토콜과도 함께 사용할 수 있다.

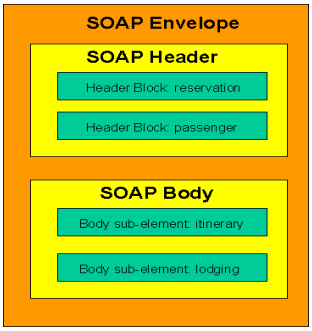

그림 3. SOAP 메시지 구조

2.3 XML 기반 웹서비스 보안

웹서비스의 보안정책은 전송계층과 어플리케이션 계층 두 부분에 적용되고 있으며, 분산 어플리케이션간 통신을 할 경우 데이터에 대한 암호화와 인증처리가 가장 중요한 부분이 될 것이다. 이러한 전송계층에서의 보안은 웹서비스가 이용하는 프로토콜의 전송계층 자체의 보안을 말하며, 전송계층은 또 다시 point-to-point와 end-to-end 관점에서 보안을 다룰 수 있다.

Point-to-point는 한쪽에서 다른 한쪽으로 데이터를 보낼 때 중간매개자(intermediaries) 없이 직접적으로 신뢰된 채널을 통해 데이터를 전송하는 방법이고, end-to-end는 중간매개자가 존재하여 간접적으로 데이터를 전송하는 방법으로, 이는 웹서비스 제공자가 한 명이 아니라 웹서비스를 제공하는 서버들이 중간 매개자가 될 수 있다는 것을 말해준다. 이러한 전송계층에서의 보안은 주로 SSL(Secure Socket Layer) 프로토콜을 통해서 제공된다. 그러나, 이러한 전송계층에서의 보안은 주고받는 모든 정보에 대해 암.복호화를 수행해야 하기 때문에 보안이 필요치 않은 데이터를 선택하여 제외할 수 없으므로 웹서버의 성능에 미치는 부담이 커져 분산환경인 웹서비스에 사용하는데 있어 효율성을 고려해 볼 때 적합하지 않은 면이 있다.

또한, SOAP 모델에 있어서 중요한 개체인 end-to-end 간의 메시지를 재전송해주는 중간계층에서의 신뢰성 또한 완전히 보장되지 않기 때문에 end-to-end 간의 안전한 통신 또한 보장할 수 없으며, 통신 링크 중 어느 하나라도 안전하지 않을 경우 end-to-end 간의 보안이 유지되지 않는 문제점도 발생한다.

그로 인해, 어플리케이션 계층에서는 XML 문서 내의 태그나 엘리먼트, 값 등 필요한 부분만을 선택하여 암호화할 수 있는 XML 기반 보안기술을 포함하는 보안방법을 이용한다. 어플리케이션 계층에서의 보안을 위해서 SOAP을 직접 수정할 수 있으며, 수정한 SOAP 메시지는 어떠한 프로토콜을 통해서도 전송가능하다. 이러한 측면에서 보안이 필요한 정보에만 end-to-end 간에 보안서비스를 해주는 어플리케이션 계층의 보안방법이 어플리케이션간 프로세서가 연결되어 상호작용하는 웹서비스와 더 잘 어울리는 면이 있다. 즉, XML 기반 웹서비스의 어플리케이션 계층의 호출을 책임지는 SOAP 메시지에 암호화와 전자서명을 해주어 기밀성, 메시지 인증, 무결성, 부인봉쇄 등의 보안 서비스를 제공해줄 수 있다.[3,7]][9-10]

3. 시스템 설계

본 논문에서 제안하는 전체 시스템은 그림 4와 같이 내부적으로 구성되어 있다.

클라이언트는 키쌍 생성모듈, XML 서명/검증 모듈, XML 암.복호화 모듈로 구성된다.

키쌍 생성 모듈은 XML 문서의 서명과 암호화에 필요한 키쌍을 생성하는 기능을 제공한다.

XML 서명/검증 모듈은 사용자 인증을 위해 XML

문서에 대한 전자서명을 생성하고 전자서명된 문서에 대한 검증을 수행하는 모듈이다. XML 서명/검증 모듈은 XML 문서뿐만 아니라 임의의 디지털 컨텐츠에 대한 인증, 무결성 및 부인봉쇄 기능을 제공한다.

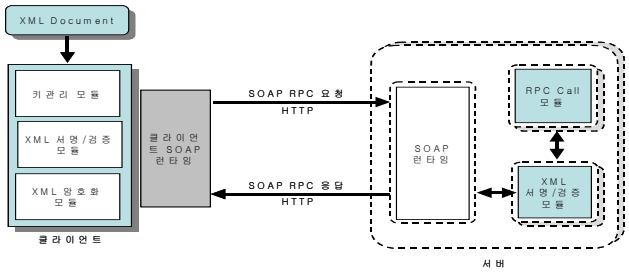

그림 4. 시스템 구성도

본 논문에서 제안된 시스템은 XML 문서의 일부 또는 전체 문서에 대한 전자서명을 수행할 수 있도록 하였으며, 하나의 전자서명 키로 다수의 문서를 전자서명 할 수 있도록 하였다. 전자서명 된 문서는 표준에 명시된 Detached, Enveloped, Enveloping 유형의 전자서명을 생성할 수 있도록 하였다. 그림 5는 서명/검증 처리과정을 나타낸다.

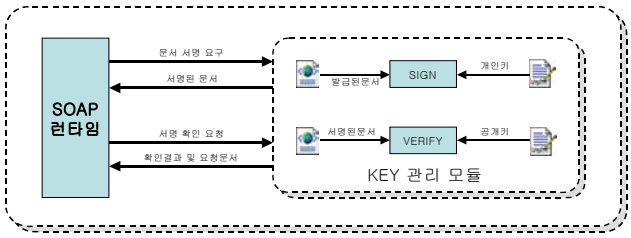

그림 5. XML 서명/검증 모듈

전체 시스템 처리과정은 아래와 같다.

① 전자서명과 암호화에 사용될 키쌍을 생성한다.
② 클라인언트는 XML 문서를 암호화하고, 암호화된 문서에 서명한다.
② 서버측 SOAP-RPC 메소드를 호출한다.
③ 서버는 서명을 확인한다.
④ 서명 검증이 성공하면 XML 문서를 복호화한 후 XML 파서를 통해 XML 문서를 파싱한다.
⑤ 서명검증이 실패하면 서명 실패 메시지를 생성한다.
⑥ XML 문서에 포함된 요청 서비스를 호출하여 실행한 후 결과값을 얻는다.
⑦ 결과 값이 포함된 XML 문서를 다시 암호화하고, 암호화된 문서에 서명한 후 클라이언트측에 전송한다.
⑦ 클라이언트는 서명된 문서를 확인하여 서명검증이 성공하면 XML 문서를 복호화한다.

4. 결론 및 향후 연구과제

최근 가장 주목받고있는 웹서비스는 플랫폼과 언어에 중립적인 프로토콜인 HTTP나 XML을 사용함으로써 클라이언트에게 전체 시스템구현을 숨길 수 있어 기존 기술과 차별화 된 서비스를 웹상에서 가능하게 한다. 하지만 웹서비스는 아직 기술적으로 성장기에 있는 기술이므로 수행성능, 확장성, 신뢰성, 보안 등 보완되어야 할 부분들이 아직 존재한다.

그 중에서도 개발자들이 가장 염려하고 있는 것 중의 하나가 웹서비스 시스템 상에서 발생할 수 있는 비 인가된 권한사용, 서비스 거부, 데이터 노출/변경, 송수신 부인 등의 보안과 관련된 문제들이다.

이에 MS와 IBM, 로터스 등의 기존의 인프라를 그대로 활용하면서 보안상의 문제를 야기하지 않고 원격 어플리케이션을 원활하게 지원하는 새로운 프로토콜 기술로서 이른바 SOAP을 공동으로 개발하였다. 특히 SOAP 1.1에서부터 지원되는 확장 모듈은 SOPA 메시지 자체에 보안기능을 둘 수 있는 메커니즘을 제공한다.

본 논문에서는 웹서비스 작업에서 클라이언트의 요청을 포함하고있는 SOAP 메시지에 XML 서명 후 전송함으로써, 요청을 받은 서버측에서 클라이언트의 요청을 수행하기 전에 정당한 클라이언트인가를 검증하는 단계를 거친다. 이렇게 함으로써 웹서비스 환경에서의 SOAP을 이용한 메시지계층 보안을 한층 강화시키려는 목적이다.

향후 본 시스템과 차세대 PKI 기술인 XKMS와 연동하여 사용자의 편의성과 보안성을 강화하는 방안에 대한 연구를 계속할 예정이다.

참고문헌

[1] W3C "Hypertext Transfer Protocol HTTP/1.1 ",1999, ftp://ftp.isi.edu/in-notes/rfc2616.txt

[2] W3C, "SOAP Version 1.2 Part 1:Messaging Framework", Candidate Recommendation19 Decem ber2002, http://www.w3.org/TR/2002/CR-soap12-p art1-20021219

[3] W3C, "SOAP Security Extensions: Digital Signature", W3C NOTE 06 February 2001, http://www. w3.org/TR/2001/NOTE-SOAP-dsig-20010206/

[4] W3C, "XML Signature Syntax and Processing", Recommendation 12 February 2002, http://www.w3.org/ TR/2002/REC-xmldsig-core-20020212

[5] Blake Dournaee, "XML Security", McGraw-H ill, 2002

[6] IBM, "XML Security Suite", 2002, http://www. alphaworks.ibm.com/tech/xmlsecuritysuite

[7] Apache, "Apache SOAP v2.3.x Documentation ", 2002, http://ws.apache.org/soap/docs/index.html

[8] W3C, "Decryption Transform for XML Signature", W3C Working Draft 18 October, 2001

[9] Joe, Web Service Gotchas, IBM, 2002

[10] Cauldwell, Professional XML Web Services, Wrox, 2001