SPKI/SDSI를 이용한 메시지 보안 프로토콜 설계 및 구현

곽문상° 홍영식
동국대학교 컴퓨터공학과
{mskwak°, hongys}@dongguk.edu

Design and Implementation of Message Security Protocol using SPKI/SDSI

Moon-Sang Kwak°, Young-Sik Hong
Department of Computer Engineering, Dongguk University

요 약

네트워크의 발달과 분산처리 시스템의 확산에 따라 정보의 노출이나 손실 및 변경과 같은 보안상의 문제들이 크게 증가하고 있다. 따라서 메시지처리와 보안에 대한 많은 연구가 진행 중에 있으며 X.400 메시지처리시스템과 메시지 보안 프로토콜이 대표적인 예이다. 하지만 메시지 보안 프로토콜은 X.509 공개키 기반구조를 사용하고 있어 인증단계가 복잡하다는 단점이 있다. 이에 본 논문은 기존 X.509 공개키 기반구조를 보다 단순화시켜 유연하게 지원하기 위해 SPKI/SDSI기반구조의 SPKI/SDSI인증형식을 갖춘 인증서를 정의하여 X.509 인증구조보다 간소화한 메시지 보안 프로토콜을 설계 및 구현하였다.

1. 서 론

X.509 공개키 기반구조(PKI:Public Key Infrastructure)를 사용하는 메시지 보안 프로토콜(MSP:Message Security Protocol)[1]은 루트(root)에서부터 시작하는 계층적인 구조의 전역이름(Global Name)을 이용하고 있어 사용자간 상호인증 시 인증기관사이의 상호인증관계에 의존하므로 인증과정이 복잡하다. 또한 권한이나 이름을 하나의 인증서에 나타내야 하므로 인증서를 소유하고 있는 사용자의 신상에 변경사항이 생기면 기존 인증서를 폐기하고 새로운 인증서를 재발급해야한다. 따라서 인증서 폐기와 재발급에 따른 많은 비용이 발생하는 문제점이 있다.[2]

본 논문에서는 X.509 공개키 기반구조의 메시지 보안 프로토콜에서 인증구조를 단순화하고 유연하게 지원하기 위해 SPKI/SDSI(Simple Public Key Infrastructure/Simple Distributed Security Infrastructure)기반구조[2][3][4][5]의 SPKI/SDSI 인증형식을 갖춘 지역이름공간(Local Name Space)을 사용하는 인증서를 정의하여 인증구조를 간소화하였다.

본 논문은 다음과 같이 이루어져있다. 2절에서는 메시지 보안 프로토콜과 SPKI/SDSI에 대하여 살펴보고, 3절에서는 본 연구에서 제안한 방법을 기술하며, 4절에서는 구현 및 성능을 분석하고, 5절에서 결론을 맺는다.

2. 관련연구

2.1. 메시지 보안 프로토콜

메시지 보안 프로토콜은 X.400 메시지처리시스템(MHS:Message Handling System)을 기반으로 하며, 메시지 보안 프로토콜 사용관리자(MSP UA:MSP User Agent)컴포넌트를 통해 기밀성, 무결성, 접근제어, 데이터인증, 송·수신자 부인봉쇄 등의 송·수신자간 메시지보안서비스(write-to-reader security service)를 제공하는 종단사용자간 프로토콜(end-to-end-user Protocol)이다. 그리고 메시지 보안 프로토콜의 메시지구조는 내용(content)과 봉투(envelope)로 구성되어 있는 X.400 메시지처리시스템의 메시지 구조와는 다르게 X.400 메시지 원문내용(Original Content)을 캡슐화(Encapsulating)하고 보안헤딩(Security Heading)을 포함하는 새로운 메시지 형식이 정의되어 있다. 또한 메시지 보안 프로토콜은 다중등급보안(MLS:Multi Level Security)을 가진 메시지를 처리할 수 있도록 접근카드인 포테자카드(Fortezza Card)를 이용하여 사용자의 접근을 통제하고 있고, 수신자의 인증서(Certificate), 사용자주요자료(UKM:User Keying Material), 보조벡터(AV:Auxiliary Vector)를 얻기 위해 X.501과 X.509의 디렉토리 시스템을 이용하는 특징을 가지고 있다.

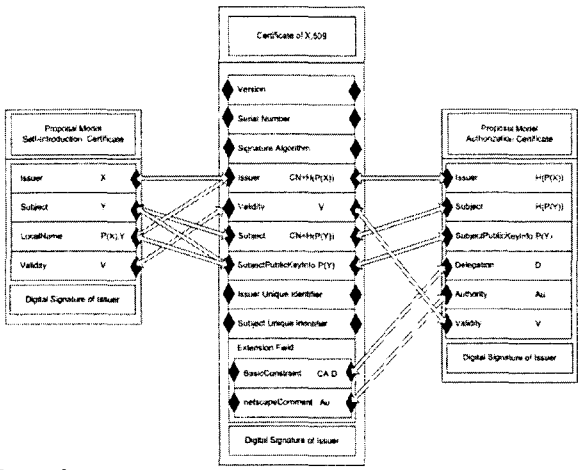
2.2. SPKI/SDSI

웹 환경에서 웹 서버와 웹 클라이언트 간 보안서비스를 제공해 주는 기존 X.509 공개키 기반구조의 문제점을 해결하기 위해 단순화시키고 유연하게 보완한 SPKI/SDSI기반구조가 연구되었다. SPKI/SDSI기반구조는 발행자(Issuer)가 공개키를 쉽게 생성하고, 지역이름공간(Local Name Space)을 도입해서 누구든지 자신의 공개키에 기반하여 주체의 이름이나 공개키와 연관된 식별자을 주어 지역이름에 연결하는 이름인증서(Name Certificate)를 발행하거나, 자신의 공개키를 지닌 주체들에게 권한을 주는 권한인증서(Authorization Certificate)들을 발행하고, 이를 이용해 서버가 클라이언트의 접근제어를 제공할 수 있도록 하였다. 예를 들면 기존 X.509 공개키 기반구조의 인증서는 통합방식인 X.509v3 인증서로 한 사용자의 권한이 바뀔 때마다 매번 그 사용자에게 새로운 인증서를 발행해야 하는 불편함이 있다. 이런 단점을 보안하고자 권한관리 기반구조(PMI:Privilege Management Infrastructure)를 두어 속성인증서(Attribute Certificate)를 정의해 사용자에게 권한을 부여하고 있지만 여전히 전역이름을 사용하고 있고, 구성요소가 또한 복잡하다. 하지만 SPKI/SDSI는 주체의 구별을 이름이 아닌 공개키로 함으로써 주체의 신상정보에 따라 수시로 변경될 수 있는 주체의 전역이름을 그 주체의 구별 대상으로 하는 X.509v3 인증서를 이용하는데 따른 문제점을 해결할 수 있다.

3. 제안 메시지 보안 프로토콜의 설계

3.1 제안 메시지 보안 프로토콜의 인증서

메시지 보안 프로토콜에서 제공하는 송·수신자간 메시지보안서비스를 제공하기 위해 [그림 1]와 같이 SPKI/SDSI 인증형식을 갖춘 자기소개인증서(SelfIntroduction Certificate)와 권한인증서(Authorization Certificate)를 XML 표현형식으로 정의하고, 제안 메시지 보안 프로토콜에서 이들 특수목적 인증서들을 이용하도록 하였다.



[그림 1] X.509v3 인증서와 제안 메시지 보안 프로토콜 인증서 비교

3.1.1. 자기소개인증서

자기소개인증서는 자신이 보유한 주체(Subject)의 공개키에 자신의 공개키에 기반을 둔 지역이름을 연결한 인증서이다. 자기소개인증서는 아래와 같이 4-튜플(tuple)로 구성한다.

$$Cert_{ID}Subject = \langle Issuer,\ localname,\ Subject,\ Vailidity \rangle_{S(Issuer)}$$

자기소개인증서의 각 튜플의 의미는 다음과 같다.

-발행자(Issuer):인증서 발행자를 말하며, 발행자는 자신의 개인키로 인증서를 서명한다.
-지역이름(local name):발행자의 공개키와 한 개 이상의 식별자로 구성되어 있다.
-주체(Subject):인증서를 받는 대상이다.
-유효기간(Validity):인증서의 유효기간을 표시한다.

[그림 2]은 자기소개인증서를 XML로 나타낸 것이다.

```
<!DOCTYPE Self-IntroductionCertificatecert SYSTEM "Self-Introduction
Certificate.dtd">
<Self-IntroductionCertificatecert>
    <issuer>
        <name>Alice</name>
    </issuer>
    <localname>
        <issuerpublickey>
            <public-key>
                <rsa-publickey>
                    <rsa-e>
                    |NFGq/E3wh9f4rJlQVXhS|
                    </rsa-e>
                    <rsa-n>
                |d738/4ghP9rFZlYZ5q9y6iskrEQpEQq8ZyMZelZzlAE=|
                    </rsa-n>
                </rsa-publickey>
            </public-key>
        </issuerpublickey>
        <subject>
            <name>Bob</name>
        </subject>
    </localname>
    <subject>
        <name>Bob</name>
    </subject>
    <validity>
        <notbefore>"2005-01-01_09:00:00"</notbefore>
        <notafter>"2006-01-01_09:00:00"</notafter>
    </validity>
</Self-IntroductionCertificatecert>
```

[그림 2] XML로 표현한 자기소개인증서

3.1.2. 권한인증서

권한인증서는 공개키를 지닌 주체(Subject)들에게 권한을 주는 인증서이다. 권한인증서는 아래와 같이 6-튜플(tuple)로 구성되어 있다.

$$Cert_{Au}Subject = \langle Issuer,\ Subejct,\ SubjectPublicKeyInfo,\ Delegation,\ Authorization,\ Vailidity \rangle_{S(Issuer)}$$

권한인증서의 각 튜플의 의미는 다음과 같다. 발행자와 주체, 그리고 유효기간 명세는 자기소개인증서의 명세하는 방법과 의미가 같다.

-위임비트(Delegation Bit) : 참 또는 거짓 값을 가지며, 권한의 전부 또는 일부를 다른 주체에게 위임할 수 있는지를 나타낸다.
-권한태그(Authorization Tag) : 발행자가 주체에게 허가하는 권한을 명세한다.

[그림 3]은 권한인증서를 XML로 나타낸 것이다.

```
<!DOCTYPE AuthorizationCertificate SYSTEM "Authroization
Certificate.dtd">
<AuthorizationCertificate>
    <issuer>
        <hash-of-key>
            <hash hash-alg="sha1">
                AMmGTeQjk65b82Jggdp+0A5MOMo=
            </hash>
        </hash-of-key>
    </issuer>
    <subject>
        <hash-of-key>
            <hash hash-alg="sha1">
                AsveX3D34gewS3SVEW3fdsfe₩ewVE
            </hash>
        </hash-of-key>
    </subject>
    <subjectPublicKey>
        <public-key>
            <rsa-publickey>
                <rsa-e>
                    |NFGq/E3wh9f4rJlQVXhS|
                </rsa-e>
                <rsa-n>
                |d738/4ghP9rFZlYZ5q9y6iskrEQpEQq8ZyMZelZzlAE=|
                </rsa-n>
            </rsa-publickey>
        </public-key>
    </subjectPublicKey>
    <delegation>
        <tags>true</tags>
    </delegation>
    <authorization>
        <entry>Top_secret</entry>
    </authorization>
    <validity>
        <notbefore>"2005-01-01_09:00:00"</notbefore>
        <notafter>"2006-01-01_09:00:00"</notafter>
    </validity>
</AuthorizationCertificate>
```

[그림 3] XML로 표현한 권한인증서

3.2 제안 메시지 보안 프로토콜을 이용한 메시지 전송 시스템

본 논문에서 제안하는 메시지 보안 프로토콜을 이용한 메시지 전송 시스템의 구조는 [그림 4]와 같다.

먼저 서버는 서버의 공개키와 개인키 키 쌍을 생성한다. 서버는 Issue Agent에 서버의 존재를 알리고, 서버의 공개키를 등록하기 위해, 서버의 공개키가 포함된 서버자기소개인증서를 생성하고, Issue Agent에 전송한다. 서버자기소개인증서를 받은 Issue Agent는 Issue Agent의 공개키와 개인키 키 쌍을 생성하고, Issue Agent의 공개키가 포함된 Issue Agent자기소개인증서를 생성하여 서버에 전송한다. Issue Agent로부터 Issue Agent자기소개인증서를 받은 서버는 Issue Agent에게 서버의

권한을 위임하기 위해 서버권한인증서와 Issue Agent권한인증서를 생성하여 Issue Agent에 전송한다.

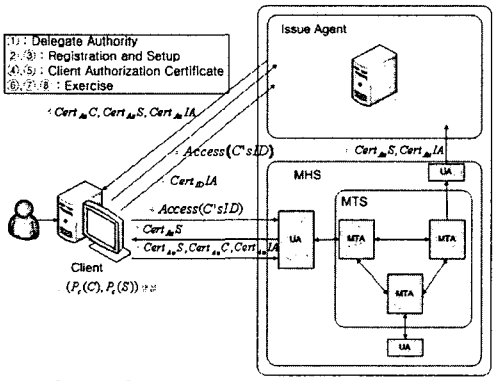
클라이언트는 클라이언트의 개인키와 공개키 키 쌍을 생성하고, 클라이언트의 공개키를 포함한 클라이언트자기소개인증서를 생성하여 Issue Agent에게 전송한다. Issue Agent는 클라이언트로부터 받은 클라이언트자기소개인증서에서 클라이언트 ID와 클라이언트의 공개키를 Issue Agent에 저장하고, 서버로부터 받은 서버권한인증서와 Issue Agent권한인증서를 가지고 클라이언트의 서버에 대한 권한을 결정하여 저장한다.

다음은 클라이언트가 서버에 대한 클라이언트의 권한을 확인하기 위해, 클라이언트는 Issue Agent에게 클라이언트 ID를 전송하여 클라이언트권한인증서를 요청한다. Issue Agent는 클라이언트의 권한을 확인한 후, 클라이언트권한인증서를 생성하고, Issue Agent는 클라이언트권한인증서와 함께 서버로부터 받은 서버권한인증서, Issue Agent권한인증서를 함께 클라이언트에 전송한다.

이제 클라이언트가 서버에 메시지를 전송하기 위해 서버에 접속하면 서버는 클라이언트에게 클라이언트 권한인증서를 요구하고, 클라이언트는 Issue Agent로부터 받은 클라이언트권한인증서, Issue Agent권한인증서, 서버권한인증서를 서버에 전송한다. 서버는 클라이언트로부터 받은 클라이언트권한인증서를 검증하고, 클라이언트의 권한을 확인하고, 클라이언트에게 request한다.

이제 클라이언트는 메시지에 메시지보안서비스를 적용하여 서버에게 전송하기 위해 클라이언트는 메시지에 지원할 보안서비스를 결정하고, 송신자의 권한과 수신자의 권한을 부여하여 접근제어를 결정한다. 그리고 메시지암호화와 서명을 하고 각각의 수신자에 대한 생성된 토큰을 메시지 보안 프로토콜 헤딩을 생성하여 포함시키고 서명한 후, 메시지를 전송한다.

수신한 메시지는 메시지 보안 프로토콜 구조가 서명되어 있는지를 먼저 조사한다. 서명이 되어 있다면 서명이 검증된 이후 메시지 보안프로토콜은 메시지에 어떤 보안 서비스가 적용되어 있는지를 알기 위해 메시지 보안 프로토콜 구조를 검사한다. 각각의 맞는 수신자토큰을 선택하고, 송신자와 수신자의 권한을 결정한 후 메시지처리를 시작한다.

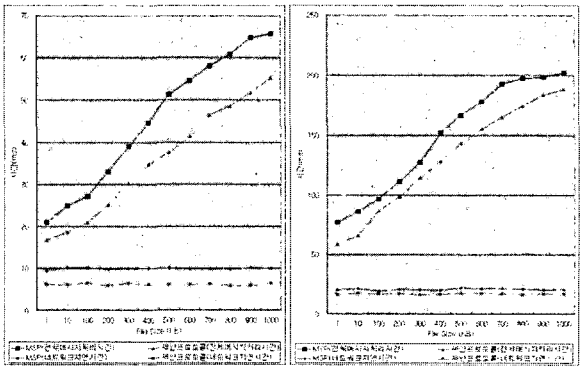


[그림 4] 제한 메시지 전송 시스템의 구조

4. 구현 및 성능분석

성능평가는 기존 X.509 공개키 기반구조의 메시지 보안 프로토콜을 사용하는 메시지 전송 시스템과 제안 메시지 보안 프로토콜을 사용하는 메시지 전송 시스템을 비교하였다. 실험은 전송회수(100회,1000회)별로 메시지의 크기를 1KByte에서 1000KByte까지 100KByte씩 증가시키며, 전체메시지전송시간과 인증서를 요청하고 발급받는데 걸리는 네트워크지연시간을 측정하였다. 메시지암호화알고리즘인 DES, 전자서명알고리즘인 RSA, 해쉬알고리즘인 MD5 알고리즘을 Microsoft Crypto Library인 CryptoAPI에서 제공하는 함수를 사용하여 구현하였다.

[그림 5]은 각각 100회와 1000회 메시지전송을 했을 경우 메시지 크기의 증가에 따른 전체메시지전송시간과 네트워크지연시간을 나타낸다. 제안된 메시지 보안 프로토콜이 기존 메시지 보안 프로토콜보다 두 경우 모두 전체시간은 20%, 네트워크지연시간은 18% 줄어졌다.



[그림 5] 100회, 1000회 메시지전송

5. 결론

본 논문에서는 X.509 공개키 기반구조의 메시지 보안 프로토콜에서 사용되고 있는 인증서를 SPKI/SDSI 기반구조의 인증형식을 갖춘 지역이름공간을 사용하는 자기소개인증서와 권한인증서를 정의하여, 기존 인증구조를 개선하였다. 그리고 이를 X.400 메시지 처리 시스템에 기반을 둔 메시지 보안 프로토콜에 적용함으로써 전송되는 메시지에 기밀성, 무결성, 접근제어, 데이터의 송수신 부인봉쇄 등의 송수신자간 메시지 보안 서비스를 제공하도록 하였다.

6. 참고문헌

[1] National Security Agency, "Secure Data Network System : Message Security Protocol(MSP), SDN.701, Revision 4.0", January 1996.

[2] Matthew H. Fredette, "An Implementation of SDSI-the simple distributed security infrastructure". Master of thesis, M.I.T, EECS, May 1997.

[3] Andrew J.Maywah, "An Implementation of a Secure Web Client Using SPKI/SDSI Certificates", Master of thesis, M.I.T, EECS, May 2000.

[4] Dwaine E.Clarke, "SPKI/SDSI HTTP Server/Certificate Chain Discovery in SPKI/SDSI", Master of thesis, M.I.T, EECS, May 2000.

[5] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, "SPKI Certificate Theory", RFC2693, September 1999.

[6] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, "Simple Public Key Certificate", Internet Draft, 31 January 2000.

[7] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, "SPKI Examples", Internet Draft, March 1998.

[8] R. Rivest, "S-Expressions", Internet Draft, May 4, 1997.

[9] J.Paajarvi, "XML Encoding of SPKI Certificates", Internet Draft, September 2000.

[10] X.Orri, "SPKI-XML Certificate Structure", Internet Draft, November 2001.

[11] Saito, T., Umesawa, K., Kito, T., Okuno, H.G., "Privacy-enhanced SPKI access control on PKIX and its application to Web server", IEEE 17th International Conference on Advanced Information Networking and Applications, 2003(AINA 2003), pp.696-703, March 2003.