XML 문서 보안을 위한 새로운 XML-Signcryption scheme 설계 및 구현

한 명 진† · 이 영 경†† · 신 정 화†† · 이 경 현†††

요 약

XML이 UN공식 표준 언어로 승인됨에 따라 이를 보완하기 위한 움직임들이 활발히 진행 중에 있다. 본 논문에서는 이러한 움직임에 맞추어 이기종간의 시스템에서 동작 가능하고 XML 문서를 보호할 수 있는 보안 메커니즘으로 "XML-Signcryption"을 설계하고 구현한다. W3C의 표준스펙인 전자서명과 암호화 기법은 전자서명과 암호화에 대한 처리를 각각 따로 구현해야 하고, 전자서명과 암호화 수행 시 총 4번의 모듈라 멱승 연산을 요구하지만, 본 논문에서 설계한 "XML-Signcryption"은 전자서명과 암호화를 동시에 실시할 수 있는 signcryption 기법을 사용하여 총 3번의 모듈라 멱승 연산을 수행함으로 계산비용 측면에서 효율성을 가진다. 그리고, 전자서명과 암호화를 XML 형식을 통해 단일 포맷으로 수행함으로써 사용자 편리성을 제공하고, 구현에 있어 기존의 W3C XML 전자서명, 암호화 규격을 이용하여 각각을 따로 구현하는 경우보다 문서 내에 파싱해야 할 태그 수가 적기 때문에 파싱 시간을 절약할 수 있는 장점도 함께 가진다. 또한, 본 논문에서는 웹 서비스 보안에 대한 연구를 기반으로, 웹 서비스 시스템에서 서비스 호출을 담당하는 SOAP(Simple Object Access Protocol) 메시지에도 XML-Signcryption을 적용하여 보안 서비스를 제공할 수 있도록 구현한다. 본 논문에서 구현한 XML 보안 툴은 XML-Signcryption 기법을 기반으로 일반 XML 문서와 웹 서비스 시스템에 기밀성, 무결성, 인증, 송신 부인봉쇄 등의 보안 서비스를 동시에 제공할 수 있다.

Design and Implementation of a new XML-Signcryption scheme to protect the XML document

Myung-Jin Han† · Young-Kyung Lee†† · Jung-Hwa Shin†† · Kyung-Hyune Rhee†††

ABSTRACT

As the XML is approved standard language by the UN, the progress which complemented the XML security has being processed rapidly. In this paper, we design and implement the "XML-Signcryption" as a security mechanism to protect the XML document that can operate between other platforms. The signature and encryption which is the standard specification in W3C needs to be able to proceed them separately. Generally the signature and encryption require four times modular exponential operation, however the signcryption only needed three times modular exponential operation. This will benefit overall system effectiveness in terms of cost. And this scheme offers to convenient the user, because the signature and encryption implement as a single XML format. This tool can save the parsing time as a number of tags is few within a document. And also, in this paper, based on a research of Web Services security, we can apply XML-Signcryption to the SOAP message to provide the security services. Based on the XML-Signcryption scheme which provides confidentiality, integrity, authentication and non-repudiation to the XML document and Web Service security simultaneously.

키워드 : XML-Signcryption, 웹 서비스(Web Services), SOAP, 보안(Security)

1. 서 론

인터넷과 네트워크 환경의 발달로 분산 처리 환경이 발전하면서 여러 가지 어플리케이션들은 네트워크에 연결된 여러 호스트에서 실행 가능하다. 현재 사용되고 있는 웹 환경의 특성상 어플리케이션에 대한 공격이 쉽기 때문에 보안의 중요성은 점점 커지고 있다. 현재 어플리케이션 계층에서 동작하는 웹 서비스가 차세대 e-Business를 주도할 것으로 많은 주목을 받으면서 웹 서비스를 기술하는 WSDL (Web Service Description Language)과 웹 서비스의 검색을 위한 UDDI(Universal Description, Discovery and Integration), 그리고 웹 서비스의 서비스 호출을 책임지는 SOAP (Simple Object Access Protocol)등의 XML 형식의 표준기술 사용이 폭넓게 성장해 가고 있다. 그러나 안전한 서비스 제공 측면에서 웹 서비스는 폐쇄 환경에서 존재하지 않았던 새로운 보안 고려사항들을 부각시키고 있다. 그 중 개발

※ 이 논문은 2002학년도 두뇌한국21 사업에 의하여 지원되었음.
† 정 회 원 : 삼성전자 CTO 전략실 소프트웨어센터 연구원
†† 준 회 원 : 부경대학교 대학원 전자계산학과
††† 종신회원 : 부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2003년 3월 18일, 심사완료 : 2003년 6월 16일

자들의 입장에서 우려하는 부분은 비 인가된 권한의 사용, 서비스 거부, 데이터 노출 또는 변경, 송수신 부인 등에 관한 보안 사항들이다. 이에 대한 해결을 위해 본 논문에서는 XML을 기반으로 하는 웹 서비스의 특성에 맞추어, 플랫폼과 구현 언어에 독립적인 특징을 가지는 XML을 Zheng[6]이 제안한 전자 서명과 암호화를 한번에 수행할 수 있는 signcryption에 적용하여 "XML-Signcryption" 구조를 설계하고 구현한다. 본 논문에서 설계하고 구현한 "XML 기반 웹 서비스를 위한 signcryption 보안 툴"은 XML-Signcryption 뿐만 아니라 XML-Signature[1]와 XML-Encryption[2]의 기능도 제공하고, 메시지 인증과 데이터 무결성, 기밀성, 송신 부인봉쇄 등의 보안 서비스를 제공한다. 또한, 구현 툴은 전자서명과 암호화를 한번에 수행하므로 계산 비용 측면에서 효율성을 제공하고 XML 기반 웹 서비스 보안 담당 개발자들에게는 구현 측면에 있어 편리함을 제공한다. 또한, 본 논문에서는 웹 서비스 보안에 대한 연구를 바탕으로, 웹 서비스 시스템에서 서비스 호출을 담당하는 SOAP(Simple Object Access Protocol) 메시지에도 "XML-Signcryption"을 적용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구로 웹 서비스 보안과 이를 위하여 W3C에서 제공하고 있는 XML 보안 메커니즘들을 살펴보고, 3장에서는 기존 signcryption 기법에 대해서 살펴본 후 웹 서비스 보안에 적용할 수 있는 XML-Signcryption 구조를 설계한다. 4장에서는 3장에서 설계한 XML-Signcryption 구조를 기반으로 "XML 기반 웹 서비스를 위한 signcryption 보안 툴"을 소개하고, 5장에서 결론과 향후 연구방향을 서술한다.

2. 관련 연구

2.1 XML 기반 웹 서비스 보안

웹 서비스의 보안정책은 전송 계층과 어플리케이션 계층 두 부분에 적용되고 있으며, 분산 어플리케이션간 통신을 할 경우 데이터에 대한 암호화와 인증 처리가 가장 중요한 부분이 될 것이다. 이러한 전송계층에서의 보안은 웹 서비스가 이용하는 프로토콜의 전송 계층 자체의 보안을 말하며, 전송계층은 또 다시 point-to-point와 end-to-end 관점에서 보안을 다룰 수 있다[14].

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

또한, SOAP 모델에 있어서 중요한 개체인 end-to-end 간의 메시지를 재전송 해주는 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 암호화 할 수 있는 XML 기반 보안 기술을 포함하는 보안 방법을 이용한다. 어플리케이션 계층에서의 보안을 위해서 SOAP을 직접 수정할 수 있으며 수정한 SOAP 메시지는 어떠한 프로토콜을 통해서도 전송 가능하다. 이러한 측면에서 보안이 필요한 정보에만 end-to-end간에 보안 서비스를 해주는 어플리케이션 계층의 보안 방법이 어플리케이션간 프로세서가 연결되어 상호 작용하는 웹 서비스와 더 잘 어울리는 면이 있다. 즉, XML 기반 웹 서비스의 어플리케이션 계층의 호출을 책임지는 SOAP 메시지에 암호화와 전자서명을 해주어 기밀성, 메시지 인증, 무결성, 부인봉쇄 등의 보안 서비스를 제공해 줄 수 있다. SOAP 1.1부터 지원되는 SOAP 보안 확장 모듈은 SOAP envelope에 W3C의 XML-Signature와 XML-Encryption을 포함해서 보안 서비스를 해 줄 수 있도록 SOAP header 엔트리의 문법과 처리 규칙을 다루고 있다[4].

2.2 W3C의 XML 보안 기술 메커니즘

XML은 문서의 데이터 표현 형식을 향상시키는 데 중점을 두고 만들어진 것으로 문서의 위·변조, 데이터 삭제 등의 보안에는 취약하다. XML이 웹 서비스의 표준으로 정해짐에 따라 XML 보안은 과거 웹 보안의 일부분에서 독립하여 새로운 분야로 분류되었다. 현재 XML 보안에 대한 명세는 W3C에서 제정하고 있으며 그 중 대표적인 XML-Signature, XML-Encryption, SOAP-SEC에 대해 간단히 소개한다.

2.2.1 XML-Signature

W3C는 IETF(Internet Engineering Task Force)와 공동으로 XML 트랜잭션에 이용할 수 있도록 설계된 디지털 서명을 정의하여 XML-Signature로 표준화했다. 이 표준에서는 디지털 서명 오퍼레이션의 결과를 가져올 수 있는 스키마를 정의하고 메시지 인증과 무결성, 서명된 데이터에 대한 부인 봉쇄 등을 지원하기 위한 내용이 포함되어 있다. XML-Signature의 장점은 특정 문서 전체에 대한 서명 또는 부분적인 서명이 가능하다. XML-Signature가 가지고 있

는 이러한 유연성은 서버의 부담을 덜어주고, 사용자의 필요에 따라 서명 할 부분을 선택할 수 있는 융통성을 제공해준다. 이는 2002년 2월 12일 recommendation 상태로 표준화가 완료된 상태이다[1].

2.2.2 XML-Encryption

XML 문서에 기밀성을 제공할 수 있는 XML-Encryption은 W3C에서 2002년 12월 10일자로 recommendation 상태로 표준화가 완료되었으며, XML 기반 데이터의 기밀성을 보장하기 위해 암호화와 복호화 및 결과 표시를 위한 처리를 명시하고 있다. <표 1>은 XML-Signature와 XML-Encryption의 기본 구조를 나타낸 것이다[2].

<표 1> XML Signature(좌), Encryption(우) 기본구조

< Signature ID? > < SignedInfo > < CanonicalizationMethod /> < SignatureMethod /> < Reference URI? > (< Transforms >)? < DigestMethod > < DigestValue > </ Reference >)+ </ SignedInfo > < SignatureValue > (< KeyInfo >)? (< Object ID? >)* </ Signature >	< EncryptedData > < EncryptionMethod /> ? < ds : KeyInfo > < EncryptedKey > ? < AgreementMethod > ? < ds : KeyName > ? < ds : RetrievalMethod > ? < ds : * > ? </ ds : KeyInfo > ? < CipherData > < CipherValue > ? < CipherReference URI? > </ CipherData> < EncryptionProperties > ? </ EncryptedData >

2.2.3 SOAP-SEC

SOAP-SEC은 2001년 2월 IBM과 MS가 W3C에 제출한 규격이다. 이는 현재 2001년 2월 6일자로 W3C에 NOTE 상태로 있다. SOAP1.1 메시지에 SOAP-SEC : Signature라는 새로운 헤더 엔트리를 추가하는 방식으로 디지털 서명을 지원한다[4].

3. 웹 서비스 보안을 위한 XML-Signcryption 설계

현재 W3C에서는 XML 규격을 지원하는 서명과 암호화 기술을 제공한다. 이러한 기술을 활용하여 본 논문에서는 서명과 암호화를 동시에 처리할 수 있는 signcryption기법을 기반으로 "XML-Signcryption"을 설계하고 구현한다. 3.1절에서는 signcryption의 기존 연구를 통해 서명 후 암호화와 signcryption의 계산량을 비교하여 어떠한 이점을 가지는지 살펴보고, signcryption의 생성, 검증 과정을 기술한다. 3.2절에서는 이러한 signcryption을 기반으로 "XML-Signcryption"의 기본구조와 스키마를 설계한다.

3.1 Signcryption에 대한 기존 연구

Signcryption은 1997년 Zheng이 처음 제안한 것으로 SDSS (Shortened Digital Signature Standard)와 대칭키 암호를 결합시킨 방법이다. <표 2>는 기존 RSA 서명 후 암호 기법과 Zheng[6]의 signcryption 기법의 계산 비용을 나타낸 것이다. 이는 암호 알고리즘 계산 비용에 가장 부담을 주는 모듈라 멱승(exp)연산을 비교할 때, RSA 기반 서명 후 암호화 기법에서는 송신측과 수신측에서 2번의 모듈라 멱승 연산으로 총 4번의 멱승 연산이 요구되는 반면, Zheng의 기법에서는 signcryption시 1번, unsigncryption시 2번으로 총 3번의 멱승 연산이 요구되므로 멱승 연산시 효율성이 높아진다[7, 9].

<표 2> 계산 비용 비교표

기 법	계산 비용	참 고
RSA 서명 후 암호 기법	exp = 2, hash = 1, enc = 1 (exp = 2, hash = 1, dec = 1)	mod p 상에서 • exp : 멱승연산 • div : 역원연산 • mul : 곱셈연산 • hash : 해쉬연산 • add : 덧셈, 뺄셈 • enc : 암호연산 • dec : 복호연산
Zheng's signcryption 기법	exp = 1, mul = 0, div = 0, add = 1, hash = 1, enc = 1 (exp = 2, mul = 2, div = 0 add = 0, hash = 1, dec = 1)	

3.1.1 Zheng의 Signcryption

(1) 표기법

- p, q : 큰 소수(단, $q \mid p-1$)
- g : 법 p상에서 위수 q인 정수
- $hash$: 일방향 해쉬함수
- KH : keyed-해쉬함수
- $k_1 \| k_2$: k_1, k_2의 연접
- E, D : 암, 복호화 함수
- x_A, x_B : 사용자 A, B의 개인키
- y_A, y_B : 사용자 A, B의 공개키

(2) Signcryption 과정은 다음과 같다(사용자 A).

① $x \in_R [1, 2, \cdots, q-1]$

② $k = hash\ (y_B^x \bmod p)$

③ $k_1 \| k_2 = k$

④ $c = E_{k_1}(m)$

⑤ $r = KH_{k_2}(m)$

⑥ $s = x/(r + x_A) \bmod q$

사용자 A는 랜덤 비밀값 x와 사용자 B의 공개키 $y_B = g^{x_B}$를 이용하여 세션키 k를 생성한다. 세션키 k를 k_1, k_2로 분리한 다음 메시지를 암호화하기 위한 키와 keyed-해쉬함수를 위한 키로 사용한다. 사용자 A는 랜덤 비밀값 x, keyed-해쉬값 r, 자신의 비밀키 x_A를 사용하여 생성한 서명값 s, (c, r, s)을 사용자 B에게 전송한다.

(3) Unsigncryption 과정은 다음과 같다(사용자 B).

① $k = hash((y_A g^r)^{s \cdot x_B} \bmod p)$

② $k_1 \| k_2 = k$

③ $m = D_{k_1}(c)$

④ $KH_{k_2}(m) = r$ 인지 검증

사용자 B는 사용자 A의 공개키 y_A와 자신의 비밀키 x_B를 이용하여 해쉬함수를 취한 세션키 k를 계산한다. k_1을 이용하여 암호화된 메시지 c를 복호화 한다. 그러나, Zheng이 제안한 signcryption기법은 송신부인이 발생하여 제 3자에게 송신부인에 대한 검증을 위탁해야 할 경우, 수신자의 비밀키를 제 3자에게 공개하지 않으면 제 3자가 검증을 할 수 없는 문제점을 가지고 있다[6].

이러한 문제는 웹 서비스와 같이 서로 빈번한 상호작용을 하는 분산 어플리케이션들 간에 노출된 키를 복구하는데 있어 오버헤드를 증가시키는 문제를 야기하게 된다. 이는 F.Bao의 변형 signcryption을 통해 해결할 수 있다[6, 7].

3.1.2 F.Bao의 변형 signcryption

(1) Signcryption 과정은 다음과 같다(사용자 A).

① $x \in_R [1, 2, \cdots, q-1]$

② $k_1 = hash(y_B^x \bmod p)$

③ $k_2 = hash(g^x \bmod p)$

④ $c = E_{k_1}(m)$

⑤ $r = KH_{k_2}(m)$

⑥ $s = x/(r + x_A) \bmod q$

사용자 A는 사용자 B에게 위의 방법과 마찬가지로 (c, r, s)를 전송한다. 이는 사용자 A가 송신 부인을 했을 경우 사용자 B가 제 3자에게 (m, r, s)를 전송하여 사용자 B의 개인키를 노출시키지 않고 사용자 A가 서명한 것임을 검증할 수 있도록 한다.

(2) Unsigncryption 과정은 다음과 같다(사용자 B).

① $t_1 = (y_A g^r)^s \bmod p$

② $t_2 = t_1^{x_B} \bmod p$

③ $k_1 = hash(t_2)$

④ $k_2 = hash(t_1)$

⑤ $m = D_{k_1}(c)$

⑥ $KH_{k_2}(m) = r$ 인지 검증

F.Bao의 변형 signcryption은 제 3자에게 서명 검증을 위탁했을 경우, $k_2 = hash(y_A \cdot g^r)^s \bmod p$를 계산해서 검증을 요청한 사용자로부터 받은 r과 ⑥번의 $r = KH_{k_2}(m)$과 동일한지 검사하면 누구에 의해 서명되었는지 검증 가능하다. 그러므로, B의 개인키 x_B를 노출시키지 않고 송신부인에 대한 검증이 가능하다[8]. 또한, 제 3자에게 검증을 위탁할 경우 수신자의 개인키를 노출시키지 않고 송신부인을 막을 수 있다는 측면에서 더 나은 기능을 가지고 웹 서비스 환경에 적용될 수 있다.

기존의 W3C에서 정의한 XML-Signature와 XML-Encryption 명세서에는 전자서명과 암호화를 따로 정의하고 있어, 서명과 암호화에 대한 분석과 처리를 위한 모듈 개발 시 각각을 별도로 개발해야 하므로 서명과 암호화를 동시에 적용하고자 하는 개발자 입장에서는 두 가지 서비스를 동시에 효율적으로 처리할 수 있는 방법을 필요로 할 것이다. 이에 본 논문에서는 XML 기반 웹 서비스 보안에 활용할 수 있는 보안 메커니즘으로 전자서명과 암호화를 동시에 효율적으로 처리할 수 있는 "XML-Signcryption"을 설계하고, 이를 위한 스키마를 설계하여 개발자들이 공통적으로 XML-Signcryption을 사용할 수 있도록 문서 구조를 정의한다.

3.2 XML-Signcryption 설계

3.2.1 XML-Signcryption 기본 구조 설계

<표 3>은 XML-Signcryption의 기본 구조로 전체를 감싸고 있는 루트 엘리먼트는 <XML_Signcryption>이다. 이것은 하부에 signcryption의 생성과 검증, 복호에 필요한 정보를 담고 있는 <SigncryptionInfo>와 XML-Signcryption의 실제 데이터 값을 포함하고 있는 <SigncryptionData>를 포함하고 있다. <SigncryptionData>는 다시 <Signcryption Value>를 자식 노드로 가지며, 키 정보를 가지고 있는 <KeyInfo>가 선택적 엘리먼트로 함께 구성되어 있다. Signcryption의 실제 파라미터 값들인 keyed hash 값은 <SigncryptionDigestValue>에 포함되고, 암호문은 <Signcryption CipherValue> 값으로 저장된다. 또한, signcryption 서명은 <SigncryptionSign>에 저장되어 XML 데이터의 기밀성, 메시지 송신자 인증, 메시지 무결성 보장, 송신 부인봉쇄 등의 서비스를 위해사용된다. XML-Signcryption은 루트 엘리먼트에 다수의 속성을 포함하고 있는데, 그 중 "Mode" 속성을 이용해 순수 signcryption 뿐만 아니라, 암호화와 서명을 각각을 독립적으로 구현할 수 있는 선택을 가능하도록 하였다. 또한, "Type"을 통해 문서 전체나 엘리먼트 혹은 값만을 선택해서 각각을 부분적으로 signcryption 할 수도 있다.

"Mode"의 설정에 따라 <SigncryptionInfo>와 <Signcryption Data>의 구성도 각각의 환경에 맞게 다른 형태를 가지게 된다. 예를 들어, 암호화만을 수행하는 "Enc" mode로 설정한 경우에는 알고리즘 정보를 명시하고 있는 <SigncyptionInfo>의 <SignatureMethod>, <KeyedDigestMethod>, <CanonicalizationMethod> 등은 생략하고 <Encrypted

Method>만을 포함시킨다. 또한, 실제 수행 값을 포함하는 <SigncryptionData>안에는 <SigncryptionCipherValue>만을 포함시키고 다른 엘리먼트 들은 제외시켜 사용한다.

전자서명만을 수행하는 "Sig" mode로 설정한 경우에는 <SigncyptionInfo>안의 <SigncryptionMethod>와 <EncryptedMethod>는 제외한 서명 알고리즘 <SignatureMethod>와 해쉬 알고리즘<KeyedDigestMethod>, 그리고 정규화 알고리즘 <CanonicalizationMethod>를 포함시켜 사용하고 실제 수행 값을 포함하는 <SigncryptionData>에는 <SigncryptionDigesValue>와 <SigncryptionSign>만을 포함시킨다. 이러한 방법으로 하나의 XML-Signcryption 기본 구조 내에서 여러 XML 보안 메커니즘들을 하이브리드(hybrid)한 형태로 사용할 수 있다.

〈표 3〉 제안하는 XML-Signcryption 기본 구조

```
< XML_Signcryption Id? MimeType?
                Mode = "Enc | Sig | Signcryption"?
                Type = "Document | Element | Content"? Encoding? >
    < SigncryptionInfo >
        < SigncryptionMethod />
        < EncryptedMethod />
        < SignatureMethod />?
        < KeyedDigestMethod />
        < CanonicalizationMethod />
        (< Transforms >)?
    </ SigncryptionInfo >
    < SigncryptionData >
        < SigncryptionValue >
          < SigncryptionDigestValue >
          < SigncryptionCipherValue >
          < SigncryptionSign >
        </ SigncryptionValue >
    </ SigncryptionData >
    < KeyInfo >?
</ XML_Signcryption >
```

3.2.2 XML-Signcryption의 스키마 설계

<표 4>는 XML-Signcryption을 정의하는 스키마의 일부로 루트 엘리먼트인 <XML-Signcryption>의 스키마 구조를 표현한 것이다.

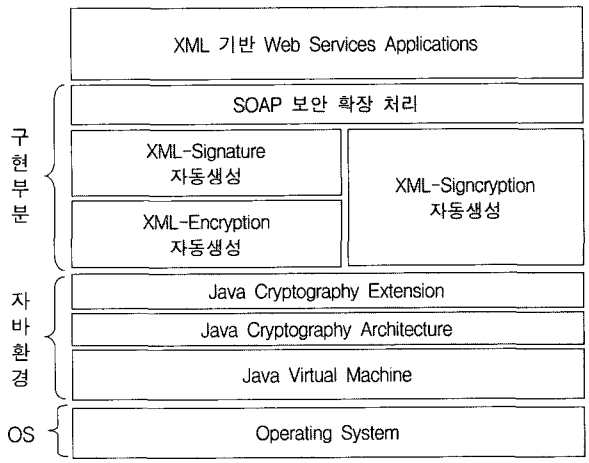
〈표 4〉 루트 element인 XML-Signcryption 스키마

```
< element
  name = "XML_Signcryption" type = "SigncryptionType" />
< complexType name = "SigncryptionType" >
   < sequence >
       < element ref = "SigncryptionInfo" />
       < element ref = "SigncryptionValue" />
       < element ref = "KeyInfo" minOccurs = "0" />
   </ sequence>
   < attribute name = "Id" type = "ID" use = "optional" />
   < attribute name = "MimeType" type = "MIME" use = "optional" />
   < attribute name = "Mode" type = "MODE" use = "required" />
   < attribute name = "Type" type = "TYPE" use = "required" />
   < attribute name = "Encoding" type = "CODING" use = "optional" />
</ complexType >
```

4. XML 기반 웹 서비스를 위한 signcryption 보안 툴 개발

4.1 XML Signcryption tool 개발

(그림 1)은 "XML 기반의 웹 서비스를 위한 Signcryption 보안 툴"의 전체 구조이고, 구현환경은 다음과 같다.

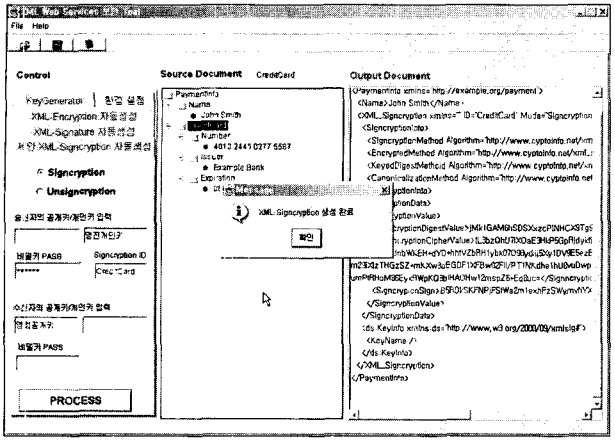


(그림 1) XML 보안 툴의 전체 구조

- Java 2 Platform Standard Edition SDK(JDK 1.3)
- JCA(Java Cryptography Architecture)
- JCE(Java Cryptography Extension)
- DOM(Document Object Model)
- CPU : Pentium IV 1.5GHz
- Memory : 256M
- OS : Windows 2000 Server, Linux 7.2

XML 규격을 지원하는 포맷과 암호 보안 메커니즘의 연동을 가능하게 하는 중간 단계에서의 프로세서역할은 Java에서 지원하는 문서 객체 모델(DOM)이 제공해 준다. DOM (Document Object Model)은 XML 문서와 관련된 작업을 할 수 있게 해주고, 본 논문에서 설계하여 구현한 Java 프로그램 코드 사이에 인터페이스를 제공해 준다. 자바 기반의 암호 라이브러리는 SUN사의 JCA 표준을 따르고, XML은 W3C 스펙의 표준을 따라 해쉬, 암·복호화를 수행하며, 실제 생성된 해쉬값이나 암호값 등은 DOM을 통해 새로운 형태의 XML 문서로 만들어진다. 이때, XML 문서 형태로 만들어진 서명이나 암호화된 결과는 XML 기반 응용 시스템에 통합이 용이하고 부분 서명과 암호화 또한 가능하다. 본 논문에서 구현한 "XML-Signcryption"의 경우 서명과 암호화를 동시에 수행하기 때문에 작업을 위한 XML 문서 파싱에 있어 서명 후 암호화를 수행하는 XML 문서보다 엘리먼트의 수가 작으므로 문서 파싱 시간을 줄일 수 있다. 또한 unsigncryption 수행시에도 엘리먼트의 수가 적기 때문에 수행 시간을 줄일 수 있는 장점을 가진다.

(그림 2)는 XML encryption, signature, signcryption 작업을 선택하여 수행할 수 있도록 구성된 메뉴를 나타내고 있다.



(그림 2) 제안 XML-Signcryption 보안 툴

구현 툴에서 제공하는 메뉴는 다음과 같다.

① KeyGenerator : RSA와 DSA 공개키 쌍 생성
② 환경설정 : 웹 서비스 기반 SOAP 메시지의 자동 인식 유무 선택
③ XML-Encryption 자동생성 : 일반 XML 문서와 SOAP 메시지에 암호화 수행
④ XML-Signature 자동생성 : 일반 XML 문서와 SOAP 메시지에 전자서명 수행
⑤ 제안 XML-Signcryption 자동생성 : 암호화와 전자서명 기능을 함께 수행

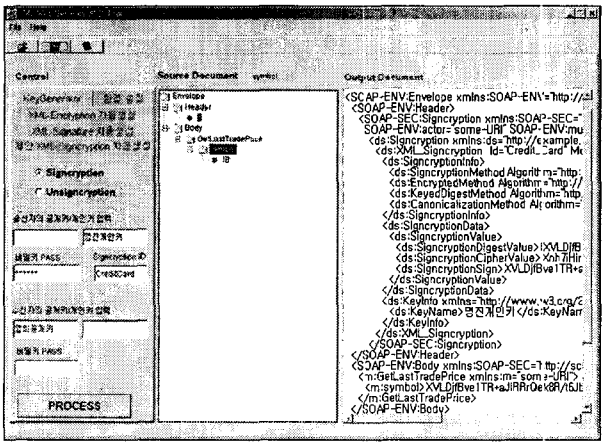
구현 모델에서 암호화와 전자서명은 RSA와 DSA 공개키 쌍과 X.509 형식의 인증서를 사용하여 수행하며, 일반 XML 문서와 SOAP 메시지에 대한 인식은 옵션으로 지정하여 자동으로 인식할 수 있도록 하고, SOAP 보안 확장 규칙에 따라 각각의 XML 보안 메커니즘을 수행할 수 있는 기능 또한 제공된다. 또한, 사용자가 XML 문서의 엘리먼트 단위나 값 단위를 선택하여 필요한 정보에만 부분적으로 서명, 암호화, signcryption을 적용할 수 있는 기능을 제공하므로 보안 관련 지식이 없는 개발자라도 본 XML Signcryption 보안 툴을 사용하여 서명, 암호화 및 signcryption을 실행할 수 있도록 한다. 그리고, 이러한 기능을 사용할 경우 XML 기반 웹 서비스 환경에 end-to-end 간의 보안 서비스를 도입할 수 있다[11, 12].

기존 W3C SOAP 보안 확장에서는 송신자의 개인키와 알고리즘을 이용해 전자서명하고 W3C signature 스펙에 의해 XML-Signature를 생성하여 SOAP header에 추가한다. 수신측에서 SOAP 요청을 처리할 때 공개키로 header 부분에 서명된 서명을 검증하도록 한다.

환경설정을 (그림 3)과 같이 해주면 SOAP 메시지를 로딩하면서 자동으로 SOAP 메시지를 인식하게 된다.

(그림 3) SOAP 인식

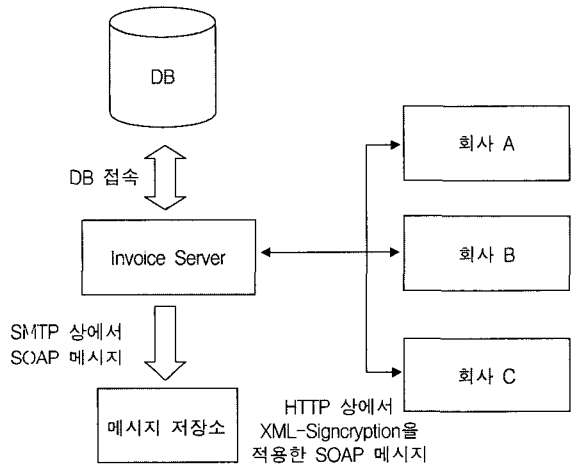
SOAP은 네트워크 상에 존재하는 각종 컴포넌트간의 서비스 호출을 효율적으로 실현하기 위한 방법을 제시하는 규약으로 크게 추가적인 정보를 담는 "header" 부분과 메시지의 실제적인 정보를 담는 "body" 부분으로 구성된다. 본 논문에서는 SOAP 메시지에 대한 signcryption을 생성할 때, SOAP 메시지에서 다른 요소의 컨테이너 역할을 하는 <envelope> 루트 엘리먼트가 추가되고, 네임 스페이스를 정의한다. 앞서 설명된 signcryption 방법과 동일한 과정을 통해 SOAP 메시지를 signcryption하고, HTTP를 통해 메시지를 전달하기 위해 <Signcryption> 엘리먼트를 자동으로 생성하여 SOAP header에 부가하는 기능이 제공된다. SOAP header에는 서명과 암호 알고리즘, 다이제스트 알고리즘에 대한 정보와 다이제스트 값들이 추가되고 body 부분에는 signcryption 된 값이 들어간다. (그림 4)는 SOAP 요청 메시지에 서비스 제공자가 가진 서비스 중에서 "GetLastTrade Price" 정보를 요청하는 SOAP 메시지에 signcryption을 적용한 예를 보여주고 있다. 이와 같이 보안을 필요로 하는 SOAP 메시지에 서명과 암호를 동시에 처리해 줌으로써 안전한 웹 서비스를 제공할 수 있다.



(그림 4) SOAP 메시지에 signcryption 적용 예

4.2 XML-Signcryption을 이용한 웹 서비스 응용 시나리오

다음 (그림 5)는 HTTP와 SMTP를 통해 호출하는 invoice(송장) 웹 서비스를 간단하게 구현한 그림이다. Invoice 서버는 HTTP를 통해 송장을 협력 회사에게 제출할 수 있는 웹 서비스를 가지고 있다. Invoice 서버는 그 자체로는 클라이언트의 역할을 하지만, 웹 서비스와 협력사의 사이트에 위치해 있는 다양한 클라이언트와 통신한다. 본 논문에서 제안한 방식을 이용하여 invoice 서버는 XML형식의 SOAP 메시지를 HTTP 상에서 XML-Signcryption을 사용하여 전송한다. 이렇게 보안으로부터 노출되어 있는 SOAP 메시지에 XML-Signcryption을 적용하면 협력사와 다양한 클라이언트 사이에서 데이터 기밀성, 송신자 인증, 메시지 무결성, 송신 부인 봉쇄 등의 보안 서비스를 한번에 모두 제공하는 안전한 웹 서비스를 제공할 수 있다.



(그림 5) 웹 서비스를 이용하는 구매 응용 프로그램

5. 결론 및 활용 방안

본 논문은 기존의 공개키를 표현하기 위해 ASN.1이나 BER을 이용한 것과 달리 XML 문법과 마크업을 이용하고 있다. XML은 어떤 형의 데이터라도 구조적인 방법으로 데이터를 기술할 수 있는 마크업 언어를 생성하기 위한 언어로 어플리케이션에서 쉽게 텍스트로 파싱(parsing)할 수 있는 장점을 가지고 있다. 한편, 바이너리 형식의 문서를 사용할 경우 XML을 사용할 때보다 문서의 크기를 좀 더 줄일 수 있는 장점은 있지만, 어플리케이션에서 바이너리 문서의 파싱이 어렵다. 이에 비해, XML은 암호화된 객체들에 명확한 의미를 부여할 수 있고, 어플리케이션에서 쉽게 파싱할 수 있는 이점을 가진다[13].

본 논문에서는 암호화와 전자서명을 따로 구현한 W3C의 XML-Encryption이나 XML-Signature와는 달리 전자서명과 암호화를 한번에 수행할 수 있는 "XML-Signcryption"을 설계하고, XML-Encryption과 XML-Signature, XML-Signcryption을 자동으로 생성하여 XML 기반 웹 서비스 환경의 어플리케이션 보호를 위해 사용할 수 있는 "XML 기반의 웹 서비스를 위한 Signcryption 보안 툴"을 구현하였다. XML 규격을 지원하는 본 툴은 원시 바이너리로 된 결과물이 아니라 서명된 내용이 XML형태로 문서 내에 포함 되어 있으므로 부가적인 정보가 필요없이 전자서명을 추가할 수 있고, 메시지 자체를 선택적으로 암호화할 수 있는 장점을 지닌다.

현재 웹 서비스 보안에 대한 필요성이 대두되고 있는 가운데 본 논문에서는 웹 서비스 환경을 기반으로 SOAP 메시지에 보안을 강화하는 것에 초점을 맞추고 있다. W3C에서 XML-Encryption, XML-Signature에 대한 스펙은 표준화가 이루어졌으나 XML-Signcryption에 대한 제안은 없으므로 이를 사용할 경우 계산량과 사용자 편리성 관점에서 볼 때 서명 후 암호화보다 효율성을 가질 것으로 판단된다. 본 툴을 이용하여 만들어진 XML 문서에 대한 서명, 암호화, signcryption의 결과가 XML 문서 형식이기 때문에 XML 기반의 응용 시스템에 통합이 용이하고, 전자상거래 플랫폼에 투명하게 접목이 가능할 것으로 예상된다. 또한, 구현 툴은 XML을 이용한 전자문서의 안전한 교환 및 유통 서비스, 그리고 B2B, B2C 등의 전자상거래에서의 안전한 주문서 교환에 요청되는 보안 서비스를 제공하기 위한 목적으로도 사용 가능하고, ebXML 기반의 전자상거래 보안 서비스 또는 EDI 서비스에서의 전자 문서 보호에도 활용 가능할 것으로 판단된다.

참 고 문 헌

[1] W3C, XML-Signature Syntax and Processing, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, 2002.

[2] W3C, XML Encryption Syntax and Processing, http://www.w3.org/TR/2002/CR-xmlenc-core-20020802/, 2002.

[3] SOAP security extensions http://www.trl.ibm.com/projects/xml/soap/wp/wp.html#SOAP, November, 2000.

[4] W3C, SOAP Security Extensions : Digital Signature, http://www.w3.org/TR/SOAP-dsig/, 2001.

[5] Proposed Federal Information Proceeding Standard for Digital Signature Standard(DSS), Federal Register, Vol.56, No.169, 1991.

[6] Y. Zheng, "Digital signcryption or how to achieve cost (signature and encryption) << cost (signature) + cost (encryption)," Advances in Cryptology, Proceedings of CRYPTO '97, LNCS, Vol.1294, pp.165-179, Springer-Verlag, 1997.

[7] Y. Zheng, "Signcryption and its application in efficient public key solutions," Proc. of Information Security Workshop(ISW '97), LNCS, Springer-Verlag, Vol.1396, pp.291-312, 1998.

[8] F. Bao and H. Deng, "A signcryption scheme with signature directly verifiable by public key," Proceeding of Public Key Cryptography (PKC '98), LNCS Vol.1431, pp. 55-59, 1998.
[9] A. J. Menezes P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography," 1997.
[10] Joe, Web Service Gotchas, IBM, 2002.
[11] Patrick, Professional XML Web Services, Wrox, 2001.
[12] Ben, Professional Java Web Services, Wrox, 2002.
[13] Blake, XML Security, Mc Graw Hill, 2002.
[14] Cauldwell, Professional XML Web Services, Wrox, 2001.
[15] Building Web Services with Java (Making Sense of XML, SOAP, WSDL, and UDDI), SAMS, 2002.
[16] 한명진, 이경현, "XML web services 보안을 위한 XML-Signcryption 설계", 한국정보과학회 추계학술발표논문집, 2002.

한 명 진

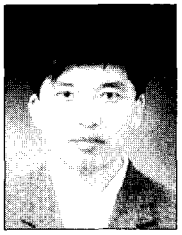
e-mail : mj7.han@samsung.com
2001년 부경대학교 컴퓨터공학과(학사)
2003년 부경대학교 대학원 전자계산학과 (이학석사)
2003년~현재 삼성전자 CTO 전략실 소프트웨어센터 연구원
관심분야 : 암호이론, XML 보안, DRM, TRS

이 영 경

e-mail : twinkle@lisia21.net
2002년 부경대학교 전자계산학과(학사)
2003년 부경대학교 대학원 전자계산학과 석사과정
관심분야 : XML 보안, Access control, authorization, SPKI

신 정 화

e-mail : shinjh@unicorn.pknu.ac.kr
1997년 한국방송통신대학교 전자계산학과 (학사)
2000년 부경대학교 대학원 전산정보학과 (이학석사)
2001년~현재 부경대학교 대학원 전자계산학과 박사과정
관심분야 : 암호이론, 네트워크 보안, 이동에이전트, XML 보안, P2P



이 경 현

e-mail : khrhee@pknu.ac.kr
1982년 경북대학교 수학교육과(학사)
1985년 한국과학기술원 응용수학과(이학석사)
1992년 한국과학기술원 수학과(이학박사)
1985년~1993년 한국전자통신연구소 선임연구원
1995년~1996년 Univ. of Adelaide, 응용수학과, Australia 방문교수
1999년 Univ. of Tokyo, 객원 연구원
2001년~2002년 Univ. of California at Irvine, USA, Visting Scholar
2002년~2003년 Intergovernmental Organization, Colombo Plan Staff College, Manila, Philippines Chair of Division of Information & Communication Technology
1993년~현재 부경대학교 전자컴퓨터정보통신공학부 교수
1997년~현재 한국멀티미디어학회 학술이사
2001년~현재 한국통신정보보호학회 논문지 편집위원
관심분야 : 정보보호론, 멀티미디어 정보보호, 네트워크 성능 평가, 그룹키 관리, 재시도 대기체계론