XKMS를 이용한 Web Service 보안 모듈 설계

김현희°, 차석일, 정종일, 신동규
세종대학교 컴퓨터공학과

Design of Web Service security module using XKMS system

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

요 약

현재 대부분의 XML 기반 e-Business 프레임워크의 연구 및 지원은 개별적인 구성 컴포넌트의 세부 설계 기법 및 구현 방법 개발에만 집중되어 있는 것이 현실이다. 과거와는 달리 보안 요소가 부가적인 요구 사항이 아닌 핵심 개발 요소라는 인식이 확대되어 가는 과정에 있다. 인터넷 보안기술 중 공개키 암호화 시스템을 이용해 향상된 보안 수준을 제공하기 위한 기반 기술인 PKI는 각종 응용 시스템 및 전자상거래 기반 기술로서 현재 가장 중요한 기술로 인식되고 사용되고 있다. 본 논문에서는 XML기반 차세대 PKI기술인 XKMS 보안 모듈 설계를 통해 웹 서비스에 보다 안전한 보안 서비스를 제공하기 위한 방안을 연구한다.

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 요구 사항이 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. 본론

2.1 웹 서비스 프레임워크

웹 서비스는 이기종 플랫폼에 탑재된 서로 다른 애플리케이션들간에 데이터 통신기능을 이용하여 작업을 자동화할 수 있는 서비스 통합 기술이라 말할 수 있다 [1]. 네트워크상에서 동적으로 상호 작용하는 다양한 이기종 플랫폼 환경이 보편화됨에 따라 관심이 증가되고 있다. 웹 서비스는 별도의 플랫폼에 별도의 언어로 작성된 프로그램들이 표준 기반으로 서로 통신할 수 있도록 상호운용성을 보장해 준다. 기존의 CORBA에도 같은 개념이 존재하지만, SOAP을 이용한 프레임워크는 확장성과 유연성이 뛰어나다. 또한, 표준 웹 프로토콜인 XML, HTTP 및 TCP/ IP와 작동함으로써 통신 프로토콜을 위한 제반 비용도 현저히 작아질 수 있다는 장점이 있다.

기존 웹은 컴퓨터의 브라우저를 통해 요청된 문서 또는 값이 서버 측에서 수행되어 그 반환 값으로

HTML 문서를 HTTP로 전달하는 방식에 이지만, 웹 서비스가 실현되면 기기와 프로그램에 상관없이 SOAP [2]을 통해 원하는 기능을 원격에서 수행할 수 있게 된다. 즉, SOAP을 통해 원격 컴퓨터에 구현된 기능을 마치 자신의 기능처럼 사용할 수 있게 된다.

현재까지의 웹은 사용자와 프로그램간의 상호 작용이 주를 이루었으나 에이전트 기술 및 개인화 서비스의 보편화로 인해 프로그램과 프로그램간 자동적인 상호 작용이 많아져 웹 서비스 표준화가 절실히 요구되고 있다.

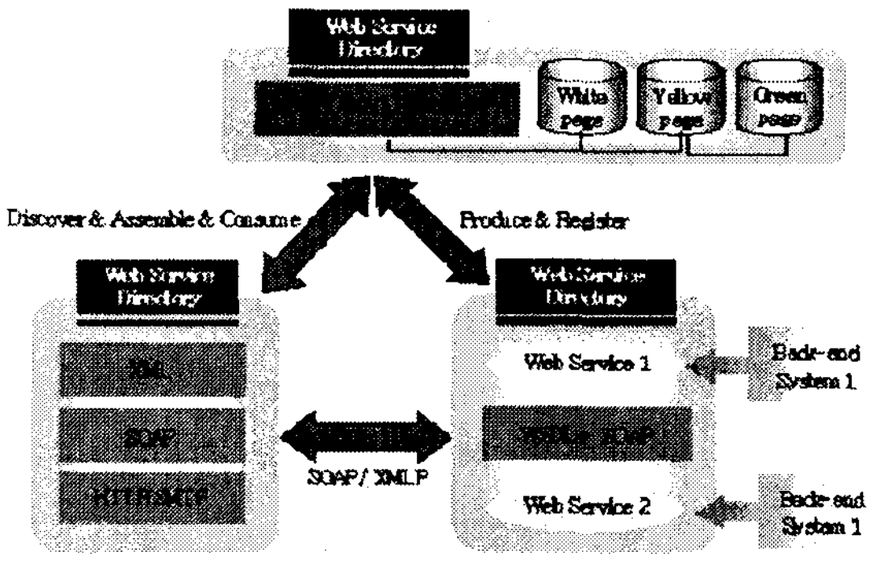
웹 서비스를 검색 및 액세스하는 일련의 과정을 살펴보면 다음과 같다.

1. 웹 서비스 공급자가 서비스를 만들어 WSDL(Web Services Definition Language) [3]로 정의한 다음 웹 서비스 디렉토리에 게시한다.

2. 이용자가 웹 서비스 디렉토리로 질의한다. 이에 대한 응답은 요청된 서비스의 WSDL 기술자를 포함한다.

3. 이용자가 WSDL 기술자의 정보를 사용하여 SOAP메시지를 구성하는 서비스를 호출한다.

일반적으로 웹 서비스 활용을 위해서는 먼저, 웹 서비스를 찾고, 찾았다면 그 이용 방법을 알아 실제로 이용하는 과정을 거치게 된다. 여기서, 웹 서비스를 찾기 위해 UDDI를, 서비스 이용 방법을 알기 위해 WSDL을, 그리고 실제로 이용하기 위한 객체간의 통신규격으로 SOAP이 사용되고 있다 [4].



<그림 1> 웹 서비스 프레임워크

2.1.1 UDDI(Universal Description, Discovery and Integration)

일반적으로 인터넷에서 기하급수적으로 증가하는 무수한 정보 가운데 필요로 하는 정보를 얻기 위한 방법 중의 하나로 Yahoo나 Google과 같은 검색 사이트를 이용한다. 웹 서비스의 정보를 얻기 위해서도 이러한 검색 기능은 필수적일 것이다. 이러한 기능을 제공하는 것이 UDDI이다.

UDDI는 전자상거래 및 웹 서비스를 온라인 디렉토리에 등록, 공개하기 위해 개발된 규약으로, 데이터는 UDDI 레지스트리(registry)에 보관되며, UDDI 레지스트리에 있는 서비스를 조회하는 경우 SOAP메시지 형식을 취하고 있는 조회 API(inquiry API)를 사용하여 검색하면 된다. UDDI는 Yahoo와 같은 디렉토리 검색엔진과 닮은 서비스라 할 수 있으며, 웹 서비스 공급자를 검색하기 위해 다음 3개의 레지스트리로 구성되어 있다.

① 화이트 페이지: 서비스 제공자의 기업명, 주소, 전화번호 등 기업명으로부터 검색하기 위한 레지스터리

② 옐로우 페이지: 서비스 제공자의 서비스 분류코드 등 업종/서비스 종류로부터 검색하기 위한 레지스터리

③ 그린 페이지: 웹 서비스를 이용하기 위한 기술정보가 등록된 레지스터리

이러한 데이터는 모두 XML기반으로 관리 및 제공되고 있다.

2.1.2 WSDL(Web Service Description Language)

UDDI 등을 통해 발견한 웹 서비스를 이용하기 위해서는 해당 서비스를 이용하기 위한 인터페이스 사양을 알아야 하며, 이 사양을 컴퓨터가 이해할 수 있는 형식으로 기술하기 위한 XML 형식 언어가 WSDL이다. 즉, WSDL은 SOAP메시지 집합 및 해당 메시지가 교환되는 방법을 설명하는 XML문서라고 할 수 있다. WSDL은 XML이기 때문에 읽고 편집할 수 있지만, 대분의 경우에는 소프트웨어에 의해 작성되고 사용된다.

이렇게 XML 스키마 표준을 사용함으로써 다양한 플랫폼과 프로그래밍 언어에서 액세스할 수 있는 웹 서비스 인터페이스를 정의할 수 있다.

WSDL은 메시지 컨텐트를 설명할 뿐만 아니라 서비스를 사용할 수 있는 위치 및 서비스와 대화하는데 사용되는 통신 프로토콜을 정의한다. 즉 WSDL은 XML 웹 서비스와 함께 작동하는 프로그램을 이용하는 데 필요한 모든 사항을 정의한다.

웹 서비스 제공자는 서비스의 이름, 호출 방법, 호출때의 파라미터, 서비스 이용 결과로 얻을 수 있는 정보 등을 정의하며 이는 다음 7가지 요소로 구성된다.

- 타입(type): 교환되는 메시지를 기술하기 위해 사용하는 데이터 타입 정의
- 메시지(message): 1개 이상의 논리적 파트로 구성된 각각의 데이터 포맷

- 오퍼레이션(operation): 전송방식을 지원하기 위한 입/출력 메시지 참조 정의
- 포트타입(port type): 논리적 오퍼레이션 세트
- 바인딩(binding): 특정 port type으로 정의된 인터페이스의 논리적 모델과 물리적 모델 연결
- 포트(port): 개별 엔드 포인트 주소를 정의
- 서비스(service): port세트를 그룹화 해 서비스명을 적어 서비스의 구체적 액세스 포인트 정의

2.1.3 SOAP

SOAP은 앞에서 설명한 바와 같이 웹 서비스를 실제로 이용하기 위한 객체간의 통신규격이다. 과거에도 COM(Component Object Model)이나 CORBA(Common Object Request Broker Architecture)와 같은 메시징이나 RPC의 기술이 있었다. 그럼에도 불구하고 현재 SOAP이 주목을 끄는 이유는 매우 간단하고 융통성이 크다는 장점 때문이다.

예를 들어, CORBA와 COM 등을 이용하여 통신을 하기 위해서는 「IIOP(Internet Inter-Orb Protocol)」로 불리는 특수한 프로토콜이 필요하지만, SOAP를 사용할 경우에는 기존의 이진(binary) 방식의 프로토콜을 이용해서 분산 객체를 사용하는 것이 아니라 텍스트 방식의 XML기반 프로토콜을 만들어서 사용함으로써 다양한 응용 프로그램간 분산객체 사용이 쉽게 이루어 질 수 있다.

또한, HTTP(Hyper Text Transfer Protocol), SMTP(Simple Mail Transfer Protocol) 프로토콜에 바인딩 해 사용하면 통신 대상은 한정되지 않고, 대부분의 파이어 월을 통과할 수 있다.

SOAP은 액세스 요구나 리턴되는 결과값으로 XML를 사용함으로써 특정의 포맷의 제약이 없고, 유연성 높은 범용의 액세스 기능을 제공하고 있다. XML에서는 데이터와 함께 데이터 명, 데이터 속성을 나타내는 태그도 동시에 포함할 수가 있기 때문에 단순한 수치형이나 문자형 뿐만 아니라 배열과 같은 반복형의 데이터나 복잡한 구조를 표현할 수 있다.

SOAP의 메시지 구성은 [그림 3]과 같이 헤더와 본체로 구성되며 각각의 기능은 다음과 같다.

- SOAP(SOAP envelope): SOAP 이 메시지를 전달하기 위해 필요한 header와 body를 정의
- SOAP 헤더: 모든 SOAP header속성들은 SOAP header의 자식엘리먼트에 적용SOAP actor 속성 정의
- SOAP 본체: SOAP메시지로서 최종수신자에게 전달될 정보를 저장RPC calls과 error reporting에 사용
- SOAP Encoding Rule: 특정 datatype값들의 교환에 사용되는 serialization mechanism 정의
- SOAP RPC Representation: 원격지 프로시저 요청이나 응답표현에 사용되는 규칙정의

2.2 XKMS 내부 구조

XKMS [5]는 크게 X-KISS(XML Key Information Service Specification)와 X-KRSS(XML Key Registration Service Specification)의 두 영역으로 구성되어 있다.

X-KISS는 XML전자서명 요소 내에 포함된 공개키 정보를 해석하는 Trust Service에 대한 프로토콜을 정의하고, <ds:KeyInfo> 내에 포함된 데이터 처리를 위한 요구사항에 대한 서비스 등을 클라이언트에게 제공한다. 이 프로토콜 설계의 핵심적인 목표는 기본적인 PKI에서의 XML구문의 확립과 복잡성의 제거로 인해, 응용 시스템에 대한 실현을 최대한 단순화하는 것이다.

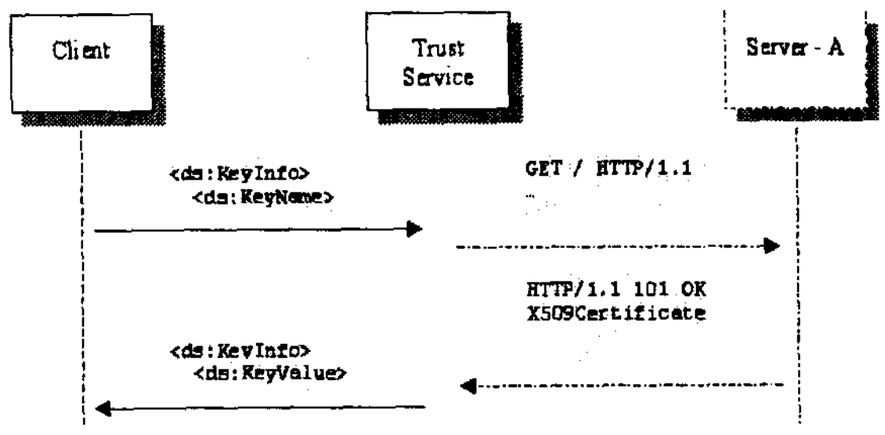
XKMS의 적용 범위는 각 구현 어플리케이션마다 상이하다. 이러한 이유로 인해 XKMS에서는 계층적 서비스 모델로 세분화시킴으로써, 업무에 따른 정확한 처리 계층을 선택할 수 있도록 정의하고 있는데 3가지로 구분 할 수 있다.

· Tier 0 : <ds:RetrievalMethod>의 처리

XML 전자서명 명세서에 따라 응용 프로그램에 의해 처리되며, Trust Service가 없이 처리된다.

· Tier 1 : Locate Service

<ds:KeyInfo>가 포함하는 데이터의 처리는 Trust Service에 위임되며, 공개키를 가진 <ds:KeyInfo>를 반환하고, <ds:KeyInfo>의 유효성은 클라이언트에 의해 수행된다.



<그림 2> Tier1 Protocol의 구조

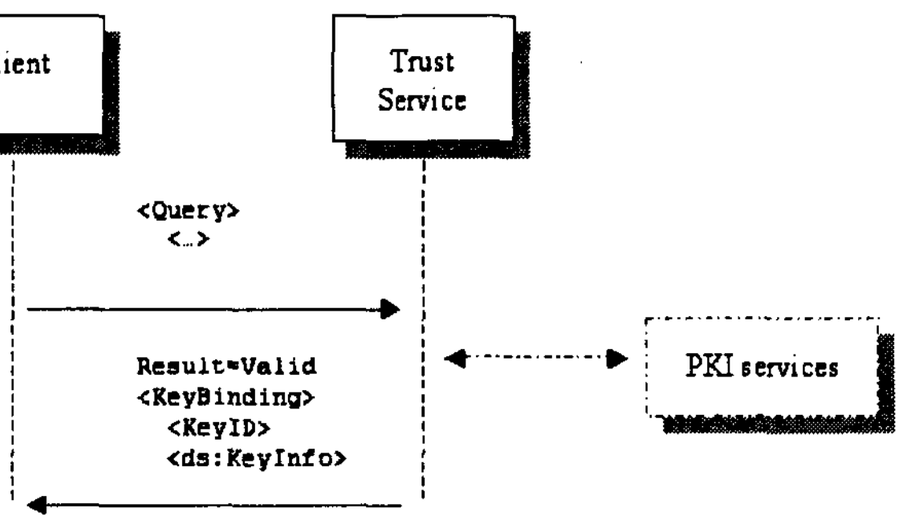
· Tier2 : Validate Service

Tier2에서는 <ds:KeyInfo>내에 포함된 데이터들이 한층 더 많은 정보를 전달해준다. 클라이언트는 공개키와 다른 데이터들 사이의 조합상태에 대한 정보들을 획득할 수 있다. 또한, 공개키와 연계하여 응답 받은 데이터의 유효성을 제공한다.

각각의 계층적인 구조 내부에서 Trust Service는 다음과 같은 기능을 클라이언트에게 제공한다.

- 복잡한 구문과 의미의 효율적인 조작

- 디렉토리와 데이터 저장 하부 구조로부터의 정보 검색
- 상태확인 및 철회
- 신뢰관계의 생성과 처리



<그림 3> Tier 2의 키 검증 서비스 구조

X-KRSS는 공개 키 정보의 등록을 처리하는 웹 서비스에 대한 프로토콜을 정의한다. X-KRSS는 키 등록, 키 폐기 및 키 복구의 전체 인증 처리과정을 단순한 단일 명세서에서 지원한다.

공개키는 등록된 즉시 X-KISS를 포함하는 다른 웹 서비스와의 결합으로 사용되어질 수 있다.

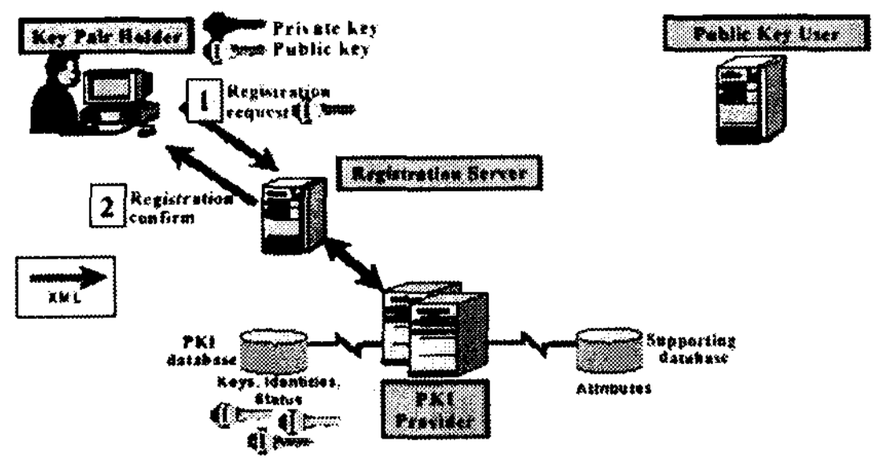
두 프로토콜은 XML Schema Language, WSDL(Web Services Definition Language v1.0)에 의해 정의된 메시지 사이의 관계를 정의하고, SOAP(Simple Object Access Protocol v1.1)을 채택하는 프로토콜 내부에서 표현되는 구조를 정의한다. 또한, 다른 응용에 부합되는 객체 인코딩 구조에서 내부적인 XKMS의 표현 역시 가능하다 [5].

X-KRSS는 아래의 전체 인증 라이프사이클을 단순한 단일 스펙에서 지원한다.

· X-KRSS : Registration

키 쌍 소유자(Key pair holder)는 등록 단계에서 자신의 공개키를 등록 서버(Registration Server)에 신뢰 기반구조(trusted infrastructure)에 등록한다. 공개키는 이때 X-KRSS에 명시된 전자 서명을 수행한 요청서에 포함되어 전송된다. 이때 요청에는 이름과 속성 정보, 인증 정보, 개인 키 소유 증명(Poof-of-possession)의 정보가 포함 될 수 있다. 등록 서버는 요청을 수신 한 뒤, XML형식의 응답을 전송한다. 이 응답 문서에는 요청에 대한 처리 결과(수락, 거절, 대기 등)와 공개키와 함께 등록되어 있는 이름 및 속성 정보를 전송한다. 요청 거절의 경우를 제외하고는 추후 참조될 키 쌍 식별자를 전송한다.

일반적인 요청 및 응답은 아래의 <그림 3>과 같은 순서를 가진다.



<그림 4> X-KRSS내에서의 키 등록

· X-KRSS : Revocation

키 폐지(Key Revocation) 기능은 클라이언트가 이전에 요청했던 키 등록 정보를 폐지 할 수 있게 한다. KeyBinding 또는 KeyAssertion 프로토타입의 Status는 Invalid로 설정되는 것과 만약 Registration Service에 요청에 대한 기록이 존재하지 않는다면 결과 코드는 NotFound가 전달되는 것을 제외한다면, 키 폐지 작업은 최초 키 등록 작업과 동일하다.

· X-KRSS : Recovery

아래의 부분을 제외하고 키 복구(Key Recovery) 작업은 최초 키 등록 작업과 동일하다. 키 복구 요청은 요청에 대한 응답시간이 필요할 수 있으며, <ResultCode>의 값이 Pending이 될 수도 있다.

만약 Registration Service상에 요청에 대한 기록이 존재하지 않는다면 결과 코드는 NotFound가 전달한다. 키 복구 요청 순서는 아래와 같다.

· 기 등록된 공개키의 개인 키를 분실.

· 오프라인(컴퓨터 네트워크이외의 방법)으로 키 관리자에게 복구 신청.

· 관리자는 오프라인으로 키 복구 인증코드(Key recovery authorization code) 신청에게 전달.

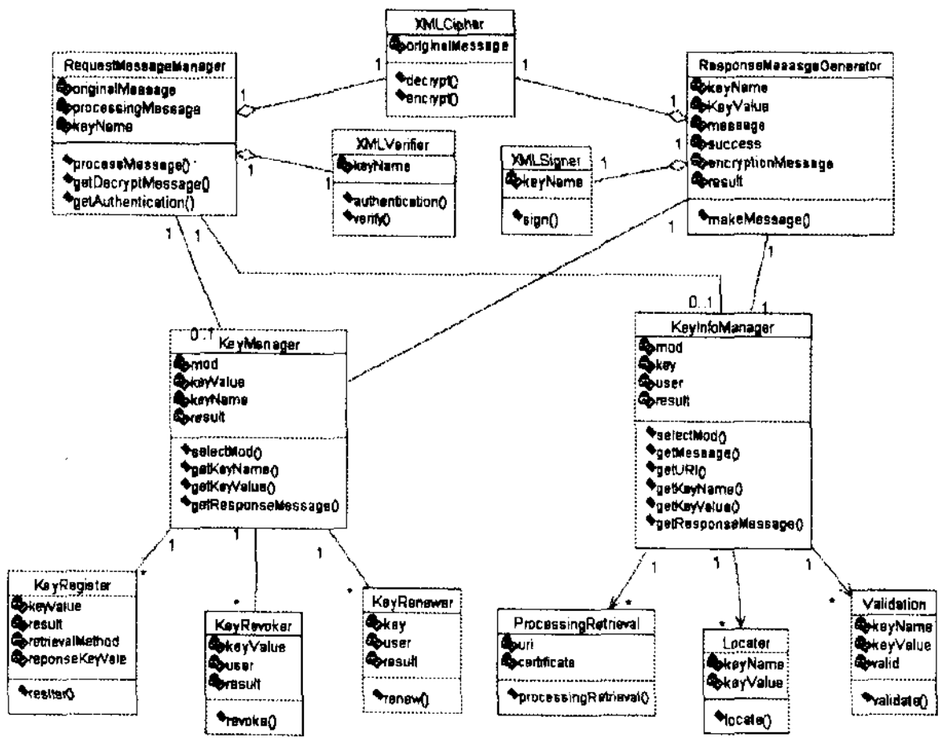
3. XKMS 시스템 설계

XKMS 모듈은 크게 키 관리 요청과 키 정보 요청의 두 가지의 UseCase로 분류 할 수 있다. 이것은 각각 X-KISS와 X-KRSS의 역할을 수행한다. 이 메인 유즈케이스를 처리하기 위한 메시지 처리와 처리 후의 메시지 처리는 메시지처리클래스에서 수행한다. 웹 서비스를 전제로 하고 있으므로 클라이언트에서 XKMS 서비스로의 요청 메시지는 SOAP 메시지 형태이며, 사용자 서명과 함께 암호화되어 있다. 반대로 클라이언트로의 응답 메시지는 메시지를 서명해서 암호화하는 과정을 거쳐야 한다.

XKMS 모듈의 사용을 시나리오를 통해서 알아보면 다음과 같은 순서로 처리된다.

(1) 클라이언트가 XKMS 웹 서비스에 인증서 정보의 타당성 검사를 요청한다.
(2) KeyInfoManager는 암호화 되어 있는 사용자 메시지의 처리를 위해 RequestMessageManager에 클라이언트의 메시지를 전달한다.
(3) XMLCipher클래스에서 클라이언트의 암호화된 메시지를 복호화한다. 복호화된 메시지는 KeyInfoManager에 반환된다.
(4) XMLVerifier클래스에서 복호화된 메시지에서 추출한 사용자를 검증한다.
(5) 인증에 성공하면 keyName과 keyValue의 인증을 위해 Validation 클래스에 인자를 전달한다.
(6) Validation은 PKI service에 정보를 요청해서 validate를 체크한다.
(7) Validation에서 검사한 결과를 ResponseMessageGenerator에 보내서 클라이언트에 보낼 메시지 형태로 변환한다.
(8) 메시지에 서버의 전자서명을 한다.
(9) 메시지를 암호화한다.
(10) 클라이언트를 결과를 전송한다.

위의 과정은 X-KRSS의 과정이 선행되었음을 전제로 한 상황이다. 즉, 사용자는 이미 자신의 공개키를 XKMS에 저장해 놓은 상태이다. 그리고, 위의 과정은 X-KISS 서비스 중 Tier2인 Validate 서비스에 해당하는 과정이다. Tier0은 Trust Service 없이 요청문서를 전달하는 서비스이며, Tier1은 <ds:KeyInfo>가 포함하는 데이터의 처리는 Trust Service에 위임되며, 공개키를 가진 <ds:KeyInfo>를 반환하고, <ds:KeyInfo>의 유효성은 클라이언트에 의해 수행된다. Tier2에서는 클라이언트는 공개키와 다른 데이터들 사이의 조합상태에 대한 정보들을 획득 할 수 있고 공개키와 연계하여 응답 받은 데이터의 유효성을 확인 할 수 있다 [6] .



<그림 5> XKMS 웹 서비스 클래스 다이어그램

위의 시나리오를 수행 할 수 있는 모듈은 <그림 5>에서 클래스 다이어 그램을 통해서 살펴 볼 수 있다. 중요한 클래스의 기능은 다음과 같다.

- RequestMessageManager: 클라이언트로부터의 서명 후 암호화된 메시지를 전달받아 복호화한 후 서명을 검증하는 기능을 수행한다.
- XMLCipher: 메시지를 암호화/복호화한다.
- XMLVerifier: client의 메시지에 서명된 사용자를 검증하는 기능을 수행한다. 참, 거짓 값을 반환한다.
- ResponseMessageGenerator: 요청을 서버에서 처리 후 그 결과를 client로 전송하기 위해서 알맞은 메시지 형태로 변환하는 기능을 수행한다.
- XMLSigner: 서버에서 처리한 결과를 담은 메시지에 서버의 개인키로 전자 서명한다.
- KeyManager: 사용자로부터의 요청 중 키 관리에 관한 요청을 수행한다.
- KeyInfoManager: 클라이언트가 요청한 사용자에 키 정보를 처리한다.
- ProcessingRetrieval: Tier0 기능을 수행한다.
- Locater: Tier1 기능을 수행한다.
- Validation: Tier2 기능을 수행한다.

3. 결론

국내외적으로 가장 주목받고 있는 웹 서비스의 현재 상황은 새로운 기술로써 과장하는 시기에서 벗어나 실제로 기업의 컴퓨팅 환경에 맞게 구현되는 상황에 도달해 있다. 이러한 시점에 실제 구현 시 가장 고려되어야 하는 부분인 보안 관련 부분에 대한 중요성이 증대되고 있다.

본 논문에서는 웹 서비스에서 기본 기술인 PKI의 차세대 모델인 XKMS를 통해서 전자상거래 사이트 상에서 협정되는 안전한 거래와 계약을 지원하며, 개발자들이 전자 인증과 다른 온라인 보안 기능을 전자상거래 응용시스템에 쉽게 접목할 수 있도록 할 수 있도록 할 수 있는 XKMS모듈을 설계했다. XKMS를 통해서 응용시스템을 공개키 내부구조에 결부시킴으로서 소프트웨어 개발자들일 PKI를 좀더 쉽게 사용 할 수 있게 될 것이며, PKI가 보편화 될 것으로 기대된다.

향후에는 안전한 문서 교환을 위한 보안 시스템에 대한 연구를 진행해서 웹 서비스가 보다 안전한 서비스를 제공 할 수 있는 시스템을 구축할 수 있는 연구가 진행 되야 할 것이다.

XKMS는 현재의 PKI 시스템과의 연동을 전제로 설계되어 있다. 사용자에게 편리함과 안전성을 보장하지만, 이러한 시스템을 제공하기 위해서는 보다

효율적인 기존 PKI 시스템과의 연동이 과제로 남아 있다. 기존 시스템에서 새로운 XKMS에 연동되기 위한 부분의 구현에 많은 어려움이 예상된다. 따라서 이 부분에 많은 연구가 선행되어야겠다.

[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.