웹 서비스 보안 기술

김 배 현*　　유 인 태**

◈ 목 차 ◈

1. 서 론

웹 서비스는 전자상거래 어플리케이션에 의한 기업 상호간 거래의 흐름에서 사람이 개입하지 않고 자동으로 웹상에서 서비스를 찾아서 요청하고 서비스하기위한 모듈화된 어플리케이션 프로그램이다.

현재, 여러 업체에서 XML을 기반으로 한 기업간 전자상거래 어플리케이션이 개발되고 있으며, 서로 다른 목적을 위해 다수의 어플리케이션으로 처리되고 있다. 이것은 개별 기업 내에서 구축할 때에는 큰 문제가 아닐 수 있으나 산업별로 아직 표준화가 완전히 진행되지 않았기 때문에 다른 산업과의 거래나 동일 산업내의 서로 다른 업체가 거래할 경우 서로 다른 포맷의 데이터베이스로 인해 상호간 거래가 어려워진다. 이러한 흐름에서 사람이 개입하지 않고 자동으로 웹상에서 서비스를 찾아서 요청하고 서비스하기 위한 웹 서비스가 차세대 인터넷 표준으로 향후 e-비즈니스를 비롯한 IT 산업의 환경변화에 큰 영향을 미칠 것이다[1-3,7]. 그러나 웹 서비스가 실제적으로 운영되기 위해서는 표준 정립, 상호 운용성, 그리고 보안문제 등 여러 가지 해결해야 할 문제점들이 있다[4,7,9].

특히 보안문제가 해결되지 않고는 웹 서비스의 발전에도 커다란 장애가 될 것이다. 이를 해결하기 위해 선진 관련 업체들과 국제 표준화기구를 중심으로 많은 노력을 기울이고 있다. 본 논문에서는 앞에서 언급한 것처럼 웹 서비스가 실제적으로 운영되기 위한 몇 가지 문제점들 가운데 보안에 관련된 문제점을 해결하기 위한 웹 보안 기술의 개발 동향을 기술하고 발전 방향을 분석하여 제시 한다.

본 논문의 구성은 2장에서 웹 서비스를 이해하기위한 웹 서비스 정의와 특징 그리고 구조를 소개하고, 3장에서는 웹 서비스를 안전하게 하기위한 웹 서비스 보안모델과 요소기술을 분석한다. 그리고 4장에서는 분석된 웹 서비스 보안모델과 요소기술의 문제점과 방전방향을 제시하고 5장에서 결론을 기술한다.

2. 웹 서비스

2.1 웹 서비스의 정의와 특징

2.1.1 웹 서비스의 정의

웹 서비스의 정의를 기술적인 측면과 비즈니스 측면으로 접근할 수 있다. 기술적인 측면의 정의를 살펴보면 웹 서비스는 분명 인터넷상에서 표준화된 기술을 사용하여 정의된 소프트웨어 어플리케이션임에 틀림없다. 하지만 비즈니스 차원에서의 웹 서비스는 소프트웨어의 기술을 통해 기업들이

* 한신대학교 정보통신학과 겸임교수
** 경희대학교 전자정보대학 부교수

다양한 비즈니스를 발견하여 운영할 수 있게 해줌으로써 웹 서비스자체를 하나의 비즈니스 로직으로 받아들일 수 있다. 이러한 두 가지 관점을 종합하여 일반적인 웹 서비스를 정의하면 다음과 같다. 웹 서비스는 인터넷과 같이 공개된 네트워크 및 관련 표준을 통해 단일한 기업내부 또는 다수의 기업간에 기존의 어플리케이션을 OS 및 프로그램 언어에 상관없이 상호운영이 가능하도록 해주는 표준화된 소프트웨어 기술로서 거래기업간의 필요한 서비스를 발견, 제공하여 다양한 비즈니스를 가능케 해 주는 것이다.

웹 서비스 제공을 위한 4가지 개념적인 필수조건은 다음과 같다.

- 인터넷상에서 서비스된다.
  웹 서비스는 ASP처럼 인터넷상에서 제공되지만, ASP와 다른 점은 사용자가 자신이 웹 서비스를 통해 서비스를 사용하고 있는지를 인식하지 못 한다는 점이다.
- 인터넷 표준을 지원한다.
  웹 서비스는 HTTP, TCP/IP 등의 표준뿐만 아니라, 차세대 인터넷 표준인 XML, SOAP, UDDI, WSDL 등을 지원한다. 이를 통해 플랫폼에 독립적이며, 상호운용성(Interoperability)이 높은 서비스 제공이 가능하다.
- 비즈니스 로직을 포함하고 있다.
  웹 서비스는 기업의 가치사슬 (Value Chain) 내에서 발생할 수 있는 특정 태스크 (예 : 재고관리, 주문관리 등)의 비즈니스 로직을 포함하고 있다. 비즈니스 로직은 특정 기업을 위해 customizing된 것이 아니라 모든 기업이 공통적으로 사용할 수 있는 표준화된 비즈니스 로직이다. 비즈니스 로직을 보유하고 있기 때문에 웹 서비스 관련 요소가 변경되었을 때 프로그래밍이 아닌 단순 조작으로 변화에 대처할 수 있다.
- 객체기술이 기반으로 된 컴포넌트이다.
  컴포넌트이므로 산업에 구별 없이 어떤 기업의 비즈니스 시스템에도 적용될 수 있으며, 기존의 패키지 소프트웨어나 자체 개발(Custom-Developed) 시스템뿐만 아니라 다른 웹 서비스와의 커뮤니케이션도 가능하다.

2.1.2 웹 서비스의 특징

웹 서비스의 가장 중요한 특징은 loosely-Coupled 어플리케이션이라는 것이다. 종래의 Tightly-Coupled 어플리케이션에서는 어플리케이션 간의 통신에 관련된 모든 것이 미리 개발 프로그래머에 의해 정해 졌다. 즉 개발 프로그래머가 정한 방식대로 어플리케이션들은 정해진 어플리케이션과 정해진 방식대로 통신을 하였으며 Connection 중에는 계속적인 관리가 요구되었다. 이러한 두 어플리케이션 간에는 Quality-of-service, 보안, Privacy, 데이터 무결성, Complex Transaction Processing 등에서 장점을 갖는다. 반면에 Loosely-Coupled 어플리케이션에서는 어플리케이션간의 통신이 표준화된 인터페이스를 통해서 이루어지기 때문에 개발 프로그래머가 미리 정의할 필요가 없으며 Registry 서비스에 의해서 원하는 어플리케이션이 자동적으로 검색된다. 따라서 개발자들의 부담이 줄어들며 관리가 쉬워진다. 또한 유연성과 상호운영성이 제공된다. 이 두 구조의 장단점은 서로 Trade-off 관계이나 향후 분산 컴퓨팅 환경이 점차적으로 이기종 간의 상호운영성이 강조되어가는 추세이기 때문에 Loosely-Coupled 구조의 단점을 보완해가면서 이 구조를 활성화시키려는 움직임으로 가고 있다. 또한 웹 서비스의 특징으로 Dynamic Look-up이 있다. Dynamic Look-up은 어플리케이션이 자동적으로, 동적으로 자신이 필요한 어플리케이션을 찾아서 원하는 기능을 수행하는 것을 의미한다. 즉 어플리케이션은 UDDI 서비스를 이용해서 자동적으로 원하는 상대 어플리케이션을 찾고 자동적으로 서로 통신하는 방법을 맞추고, 기존의 상대 어플리케이션을 찾지 못할 때는 동적으로 다른 어플리케이션을 찾아서 원하는 기능을 수행한다. 이 모든 것이 인간의 개입 없이 이루어지게 된다. 그리고 웹 서비스는 Cross-Platform, Program-to-Program 통신이라는 특징을 갖는다. 웹 서비스는 표준화된

SOAP, WSDL을 사용해서 인터페이스를 하기 때문에 웹 서비스는 Heterogeneous 환경, 즉 Cross-Platform에서의 상호운영성을 제공한다. 종래의 다른 분산 컴퓨팅 구조들도 이러한 점을 지향하려고 했지만 결국 표준화되지 않는 인터페이스로 비용과 관리, 확장성 문제로 인해서 모두 실패하였다.

웹 서비스의 대표적인 특징을 정리하면 다음과 같다[8].

- 분산 컴퓨팅 기술 측면에서 플랫폼 독립적이다.
  웹 서비스는 매우 유연한 어플리케이션(loosely coupled application) 구조를 가지고 있다. 따라서 웹 서비스는 유연한 어플리케이션 구조를 가지고 있기 때문에 서비스 공급자, 수요자가 특별한 기능을 추가하기 위해 새로운 플랫폼을 사용하지 않아도 되며, 플랫폼 선택도 자유롭다.
- 디바이스 및 위치 독립적이다.
  웹 서비스를 통해 PC, PDA, 핸드폰 등 다양한 유무선 디바이스를 통해 시간 및 장소에 상관없이 웹 서비스에 접근이 가능하다.
- 동적인 기능(dynamic function)이다.
  동적 기능이란 자동적으로, 동적으로 어플리케이션이 자신이 필요한 어플리케이션을 찾아서 원하는 기능을 수행하는 것이다. 이 모든 기능에는 인간이 개입하지 않고 이루어진다. 기업에서 요구되는 다양한 기능들을 적절한 서비스 제공자로부터 찾을 수 있고, 실시간으로 연계될 수 있으며, 서비스 제공자와 고객의 역할이 고정되어 있지 않다. 그러므로 웹 서비스를 통한 자사의 필요한 기능 또는 공급자들을 자유롭게 선택할 수 있어서 보다 비용 효율적인 기능으로 대체가 가능하며, 새로운 비즈니스 모델로 변화하는데도 편의성을 제공해 줄 수 있다.
- 상호운영성을 제공
  표준화된 SOAP, WSDL을 사용하여 인터페이스하기 때문에 이기종 환경에서 상호운영성을 제공할 수 있다. 따라서 웹 서비스를 기존 시스템이 적용이 가능하다. 기존에 투자되었던 IT 어플리케이션 및 인프라 등 기존의 시스템에 특별한 웹 서비스 프로세스를 포함시켜 운영할 수 있다.

2.2 웹서비스의 구조

웹 서비스의 구조는 3가지의 빌딩 블록과 각 빌딩 블록의 기능을 수행하기 위한 요소기술로 이루어진다[7]. 웹 서비스의 주요 빌딩 블록은 Discovery, Description, Invocation의 개념으로 이루어진다. Discovery는 XML 웹 서비스를 사용하기위해, 사용자 어플리케이션 프로그램이 필요한 웹 서비스를 발견하는 것이다. Description는 사용자에게 XML 웹 서비스가 어떤 것인지 설명하는 것이다. Description 빌딩 블록은 XML 웹 서비스의 의미를 나타내거나, XML 웹 서비스를 설명하는 메타데이터로 생각할 수 있다. Invocation은 사용자가 웹 서비스에 필요한 입력요소를 넘긴 다음, 적절한 결과 데이터를 반환 받을 수 있도록 웹 서비스를

(표 1) 웹 서비스 구조 및 기술요소

빌딩블록	기술요소	기술표준
Invocation	• Message Exchange	SOAP
	• Security	SAML, XKMA, SOAP Security Extions (WS-Security)
	⇒Message Encryption	XML Encryption
	⇒Digital Signature	XML Digtal Signature
	• Binary Attachment	SOAP with Attachment, SOAP 1.2
	• Reliable Messaging	-
	• Transaction	BTP
	• Routing	-
	• scalability	-
Description	• Service Description	WSDK, WSCL
	• process Flow	BPML, WSFL, XLang
Discovery	• Inspection	WSIL(WS-Inspection)
	• Discovery	UDDI

호출(invole)하는 것이다. 이러한 invoke 블록은 확장 형태의 SOAP 프로토콜을 포함하고 있다. invoke 빌딩 블록은 전송 프로토콜(일반적으로 HTTP, SMTP등)로 구성되어 있는 전송계층의 최상위에 위치한다.

3. 웹서비스 보안

3.1 웹 서비스 보안 요구사항

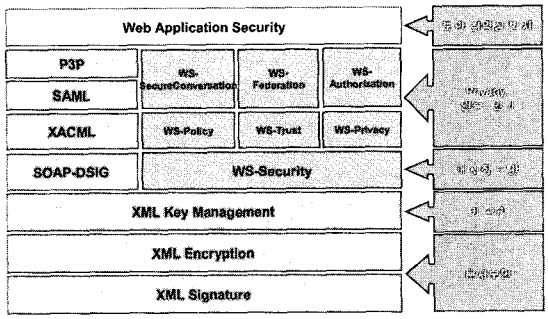
웹 서비스가 이루어지기전에 요청자와 웹 서비스는 상호간에 서로를 확인해야하며, SOAP 메시지에 대한 무결성, 기밀성, 부인봉쇄 등의 필수적인 보안 기능이 만족되어야한다. 웹 서비스는 각각의 보안 정책을 가지고 있는 서비스 주체(subject)간의 trust, Federation설정을 통한 상호 협력적인 방식으로 서비스가 이루어지기 때문에 웹 서비스 보안 아키텍처 는 보안 기술 측면뿐만 아니라 비즈니스 프로세스 측면의 보안을 효율적이고 안전하게 지원하기 위해 유연성 및 확장성이 있는 구조이어야 한다. 또한 요청자와 웹 서비스 사이에 여러 Intermediary가 존재하는 Multi-Hop Topology이기 때문에 양단간의 End-to-End 보안이 지원되어야 한다.

웹 서비스에서 요구하는 보안 서비스를 정리하면 다음과 같다[7,8].

- 인증
- 권한
- 무결성
- 기밀성
- privacy
- 가용성
- 부인봉쇄

이외에도 End-to-End 보안, Challenge/Response 형태의 보안 Context 설정, 키 교환 및 Derived key, Multiple Trust Domains 환경에서의 Trust/Federation의 설정 및 관리 등이 요구된다.

3.2 웹 서비스 보안기술



(그림 1) XML 보안기술 유형별 분류

그림 1은 MS와 IBM에서 제안하고 표준화를 위해서 작업 중인 웹 서비스 보안 Specification의 Roadmap이다. 그림 1에서 알 수 있듯이 보안에 관련된 Specification은 요구 사항별로 모듈화 되었으며 완성 단계별로 계층화되어 있다[3,7-8].

WS-Security는 SOAP 메시지 무결성과 기밀성, 인증을 포함하는 메시지 보호수준(Quality of Protection)을 제공하기 위하여 MS와 IBM이 Verisign과 함께 만든 웹 서비스 보안의 기반이 되는 Specification이다. WS-Security는 SOAP 메시지에 대한 어플리케이션 단계 보안에 대한 내용을 기술하고 있는 것으로 바이너리 보안 토큰을 인코딩하는 방법과 X.509인증서나 커버러스(Kerberos) 티켓 등을 사용하는 방식 등을 정의하고 있다. 특히, 이 specification은 OASIS에 제안되어 WSS(Web Service Security)라는 명칭으로 표준화가 진행되고 있다.

WS-Policy는 수신자와 송신자가 보안 요구사항과 지원 가능한 정도를 명시하는 방법을 제공하는 것으로 다음의 네 가지 문서가 포함되어 있다.

- Policy Framework(WS-Policy) 문서: 웹 서비스 정책을 표현하는 문법 정의.
- Policy Attachment (WS-Policy-Attachment) 문서: 정책들을 웹 서비스에 어태치하는 방법 정의.
- 일반적인 정책 선언 (WS-Policy-Assertions).
- 보안 정책 선언 (WS-Security Policy)

WS-Trust는 보안 토큰 서비스가 보안토큰의 발행, 교환, 유효성검사를 제공하는데 사용되는 인터페이스를 정의하는 것으로 당사자간에 직접 신뢰관계를 형성하는 방법과 신뢰할 수 있는 중간 계층을 통해 신뢰관계를 형성하는 방법을 소개한다. 이것은 다양한 인증 및 권한 메커니즘을 수용하는 여러 개의 보안 토큰 포맷의 생성을 지원하도록 설계되었다.

웹 서비스를 개발, 관리, 사용하는 조직들은 종종 자신들의 프라이버시 정책을 명확하게 표명하고, 들어오는 요청들이 발신자에게 이러한 정책을 따르도록 요구하도록 할 필요가 있다. 따라서 WS-Policy, WS-Security 및 WS-Trust를 결합하여 사용함으로써 조직들은 프라이버시 정책을 명시하고 이를 따르도록 지시할 수 있다.

WS-Privacy는 프라이버시 용어가 WS-Policy 설명에 어떻게 포함될 수 있는지, 그리고 프라이버시 클레임을 메시지와 연결시키는데 WS-Secrutity을 어떻게 사용할 수 있는지를 설명할 것이다. 마지막으로, 이 사양은 사용자 선호와 조직적인 실행 요구에 대해 이들 프라이버시 클레임을 평가하는데 WS-Trust 메커니즘이 어떻게 사용될 수 있는지를 설명한다.

WS-SecureConversation은 웹 서비스가 요청자 메시지의 신원을 어떻게 확인하고 요청자가 서비스의 신원을 어떻게 확인하며 상호 신원 확인된 보안 문맥을 어떻게 구축하는지를 설명한다.

WS-Federation은 WS-Security, WS-Policy, WS-Trust 및 WS-SecureConversation을 사용하여 사이트 또는 조직간 ID 연동을 위한 specification이다. 현재 WS-Federation이 SAML을 채택하여 사용할지에 대해서는 미지수이다. WS-Authorization은 웹 서비스에 대한 접근 정책이 어떻게 지정되고 관리되는지를 설명한다.

3.3 웹 서비스 보안 모델

웹 서비스 보안은 다음과 같이 2가지 단계에 적용할 수 있다[7].

- 전송 단계(point-to-point level) 보안
- 어플리케이션 단계(End-to-end level) 보안

전송 단계 보안은 특정 웹 서비스가 사용하고 있는 전송 또는 네트워크 계층 프로토콜에 이미 구축된 보안 특성을 사용하는 것으로 이루어진다. 전송 단계 보안은 웹 서비스와 관련해서는 점대점 보안(point-to-point security)이다. 점대점 보안은 컴퓨터나 어플리케이션 프로그램과 같은 하나의 점(Point)으로부터 다른 점으로 직접 안전하게 통신하는 것을 의미한다. 이것은 둘 사이에 다른 SOAP 매개물이 존재하지 않고 직접 접속된다는 것을 내포하고 있다. 전송 단계 보안 모델은 간단하고 이해하기 쉬우며 주로 인트라넷 기반의 여러 시나리오에 적합하다. 네트워크 계층에서 전송 단계 보안을 구현하는 것은 IPSec(Internet Protocol Security), SSL, VPN, 방화벽와 같은 기술을 사용하여 IP 트래픽에 대한 보안을 구현하는 것으로 이루어진다.

어플리케이션 단계 보안은 전송단계 보안의 지원 없이 어플리케이션의 메시지 자체에서 내부적으로 보안 메커니즘을 가지고 있는 것이다. 어플리케이션 단계 보안에서 중요한 점은 종단간(End-to- End) 보안을 제공한다는 것이다. 이와 같은 장점 때문에 최근의 웹 서비스 보안의 흐름은 어플리케이션 단계 보안으로 가고 있다. 웹 서비스는 근본적으로 요청자와 웹 서비스 상호간의 SOAP 메시지 교환이라고 볼 수 있기 때문에 가장 기본적인 웹 서비스 보안은 XML, 즉 SOAP 메시지 보안부터 시작해야 한다. 어플리케이션 단계에서는 보안을 적용할 때에는 SOAP 메시지 자체를 수정한다. 이러한 방식으로 수정한 메시지는 어떠한 프로토콜로도 전송할 수 있으며, 서버나 클라이언트 시스템 소프트웨어를 특별히 설정해줄 필요가 없다. 그러나 웹 서비스 메시지를 교환하는 클라이언트와 서버에서 동시에 지원하도록 하기위해 어플리케이션 단계의 특정한 구현이 필요하다.

4. 웹 서비스 보안기술 발전 방향

웹 서비스 보안을 효율적이고 안전하게 지원하기

위해서는 유연성 및 확장성이 있는 구조이어야 하며 양단간의 End-to-End 보안이 지원되어야 한다. 따라서 웹 서비스에서 요구하는 보안 서비스는 인증, 권한, 기밀성, 무결성, 가용성, 부인방지, End-to-End 보안 등 이다. 웹 서비스 보안을 위해 기존의 보안 기술을 그대로 적용할 경우, 기존 보안기술에서 일부 문제점이 지적되고 있는 상황이며, 웹 서비스 특성에 맞는 보안요구사항을 만족할 수 없다.

웹 서비스 보안 접근 방법은 전송단계 보안과 어플리케이션 단계 보안으로 구분할 수 있다. 전송단계 보안에서는 Point-to-Point 보안으로 기존의 네트워크 보안 기술인 IPsec, SSL, VPN, S/MIME 등을 사용하여 무결성과 기밀성을 제공한다. 어플리케이션 단계 보안은 End-to-End 보안으로 어플리케이션의 메시지 자체에 보안 메커니즘을 가지고 있는 것이다. 어플리케이션 단계 보안을 위해서는 XML 명세들을 사용하는데 대표적인 것으로 WS-Security을 사용하여 무결성, 기밀성, 인증을 제공한다.

우선 전송단계 보안의 경우, 웹 서비스에서는 전송단계에서 일반적으로 HTTP를 사용한다. HTTP는 인증 위주의 보안기술을 주로 적용한다. HTTP의 Basic 인증은 ID/패스워드만으로 인증을 하는 가장 간단한 형태의 보안 방식이다. 그러나 이 방식은 사용자의 패스워크가 평문 형태로 전송되기 때문에 공격자에게 노출될 위험성이 크다. 다른 방법으로는 패스워드에 대한 Digest를 생성하여 이를 전송하는 방식이 있지만 이것 역시 Digest가 평문 형태로 전송되기 때문에 안전하지 못하다. 이를 해결하기 위해서 검증된 가장 널리 쓰이는 방식이 SSL을 사용하여 Line 암호화를 하는 방식이다. 그러나 전송되는 모든 데이터가 전송 노드사이에서 암호화/복호화되기 때문에 Multi-Hop 토플로지에서 End-to-End 보안을 지원하지 못한다. 또한 SSL은 웹 서비스에서 성능에 부담을 준다. 따라서 SSL은 중요한 웹 서비스에 강력히 권고되지만, 보안 수준이 낮은 상황이나, 인트라넷과 같은 네트워크를 쉽게 통제할 수 있는 상황에서는 다른 인증 방식이 더 좋은 성능을 발휘한다. 또한 SSL은 암호화 통신 시 타이밍 기반 공격(Timing-based Attacks)에 의한 비밀키 노출, Klima-Pokorny-Rosa 공격, SSL/TLS상의 CBC 암호시 타이밍기반 공격, 암호화 라이브러리 및 어플리케이션 프로그램에 대한 타이밍 공격 등 취약성을 가지고 있다. 다음으로 VPN과 방화벽을 사용하는 경우, VPN은 요청자의 IP가 미리 고정적으로 알려지는 웹 서비스에 적용 가능하며 Connection이 Long-Term인 관계로 성능 면에서 문제가 있을 수 있다. 방화벽은 웹 서비스에서 적용될 때는 IP Blocking으로 허용되지 않은 접근을 차단한다. 방화벽은 외부에 주로 서버를 운용하는 웹 사이트와는 달리 웹 서비스는 기업 내 어플리케이션과 다른 기업 내 어플리케이션간의 통신이 필요하기 때문에 방화벽을 통과하면서 보안을 지원해야 하기 때문에 웹 서비스에는 적용이 곤란하다.

어플리케이션 단계 보안은 WS-Security에서 PKI, Kerberos, 전자서명 등을 사용할 수 있다. 그러나 PKI는 사용자 쪽의 부담 때문에 웹 사이트의 인증 방식으로 널리 쓰이지는 않으며, Kerberos는 상호호환성이 없기 때문에 Cross-Platform 환경에 적용하기는 힘들고 패스워드 사전 공격에 취약성을 가지고 있다. 전자서명은 이미 사용된 서명 값을 재사용하는 Replay공격에 대한 취약점이 있다.

따라서 기존 보안기술을 SOAP 메시지에 적용할 때도 역시 발생할 수 잇기 때문에 이에 대한 해결책이 고려되어야 한다. 서명과 함께 난수 값, Timestamp, 순서번호, Expirations, 메시지 Correlation 등의 정보를 같이 전송하는 방안을 고려해야 한다. 웹 서비스 보안 모델의 방향이 어플리케이션 단계 보안 모델 쪽으로 방향을 잡아가는 흐름 속에서 웹 서비스 보안의 요구 사항을 만족시켜 줄 수 있는 웹 서비스 보안 아키텍처의 모델을 기반으로 보안 요구 사항을 충족시켜 주는 표준 Specification의 필요성이 확산되었다. 표준 Specification 없이 기업들이 나름대로의 웹 서비스 보안 솔루션을 적용한다면, 상호운영성이 떨어지게 되고 이를 맞추기 위해서 또 추가적인 작업이 소요되는 경우가 발생하게 된다. 이러한 상황을 인식하고 MS와 IBM은 위에서 언급한 웹 서비스 보안 요구 사항을 반영한 웹 서비스 보안 아키텍처를 제안하였다.

5. 결 론

웹 서비스는 인터넷 표준 프로토콜을 이용하여 원격지에 있는 웹 객체를 XML 기반으로 접근, 이용, 재사용을 할 수 있는 웹 분산 환경의 분산 컴포넌트 모델이다. 따라서 웹 서비스는 전자상거래, 에이전트 시스템 등 다양한 어플리케이션에서 널리 사용될 수 있다. 웹 서비스는 Cross-Platform 환경에서의 기업 내 또는 기업간의 어플리케이션을 원하는 방식으로 서로간의 기능을 공유하는 것이 필수적이다. 그러나 이러한 기능을 제공하기 위해서는 보안문제가 필수적으로 대두된다. 웹 서비스 보안을 효율적이고 안전하게 지원하기 위해서는 유연성 및 확장성이 있는 구조이어야 하며 양단간의 End-to-End 보안이 지원되어야 한다. 따라서 웹 서비스에서 요구하는 보안 서비스는 인증, 권한, 기밀성, 무결성, 가용성, 부인방지, End-to-End 보안 등 이다. 웹 서비스 보안 접근 방법은 전송단계보안과 어플리케이션 단계 보안으로 구분할 수 있다.이두가지 방법 모두 장단점이 있으나 웹 서비스의 특성상 전송단계 보안 보다는 어플리케이션 단계의 보안으로 가는 추세이다. 그러나 기존의 인터넷 보안 기술 등을 단순하게 적용하는 것으로 웹 서비스 보안 요구사항을 만족할 수 없다. 따라서 웹 서비스의 특성에 맞는 보안기술이 따로 개발되어야 한다. 또한 웹 서비스의 상호운영성을 위해 표준화가 필요하다.

참 고 문 헌

[1] Martin Naedele, "Standards for XML and Web Services Security" computer April 2003 p.96~98

[2] Yuichi Nakamur, Satoshi Hada and Ryo Neyama, "Towards the Integration ofWeb Services Security on Enterprise Environments", SAINT, 2002

[3] Francisco Curbera, Matthew Duftler, Rania Khalaf,William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana, "Unraveling the Web Services Web" IEEE INTERNET COMPUTING, MARCH.APRIL 2002

[4] J.D. Meier, Alex Mackman, Michael Dunner, and Srinath Vasireddy, "Web Service Security", http://www.microsoft.com/korea/msdn/library/dnnetsec/html/SecNetch10.asp

[5] Giovanni Della-Libera, Brendan Dixon, "WS-SecureConversation", http://www.microsoft.com/korea/msdn/library/dnglobspec/html/ws-secureconversation.asp, 2002

[6] Giovanni Della-Libera, Phillip Hallam-Baker, "WS-SecurityPolicy", http://www.microsoft.com/korea/msdn/library/dnglobspec/html/ws-securitypolicy.asp, 2002

[7] SOAP Version 1.2 Part 0: Primer, W3C Recommendation 24 June 2003, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

[8] Blake Dournaee, "XML Security", McGraw-Hill, 2002

[9] Patric Cauldwell, Rejesh Chawla, Vivek Chopra,"Professional XML Web Services" Wrox Press, September 2001

[10] 이해규, 이상수, 김문규, "웹 서비스 보안" 정보처리학회지 2002년 제9권 4호, pp.36~45

◐ 저 자 소 개 ◑

김 배 현

1995년 호원대학교 전자계산학과 졸업(학사)
1997년 수원대학교 대학원 전자계산학과 졸업(석사)
2003년 경희대학교 대학원 컴퓨터공학과 박사과정 수료
2004년 9월~현재 한신대학교 정보통신학과 겸임교수
관심분야 : Mobile IP, 차세대 네트워크, 이동통신, 네트워크 보안

유 인 태

연세대학교 (공학사, 공학석사, 공학박사)
1997년 The University of Tokyo
1997년 10월~1999년 3월 삼성전자 선임연구원
1999년 3월~2003년 2월 경희대학교 전자정보학부 조교수
2003년 3월~현재 경희대학교 전자정보대학 부교수
관심분야 : 차세대 네트워크/인터넷, 무선 및 이동통신, 멀티미디어 트래픽 관리, 네트워크 QoS, 정보보호