http://dx.doi.org/10.14400/JDPM.2013.11.12.361

웹서비스 보안성 강화 방안

이성훈
백석대학교 정보통신학부

A Security Enhancement Method for Web Service

Seong-Hoon Lee
Div. of Information Communication, Baekseok University

요 약 최근 인터넷이 발전함에 따라 월드와이드웹(world wide web) 기반의 서비스 규모는 기하급수적으로 증가하였다. 또한 최근 기업간 비즈니스 로직의 구현에 웹 서비스를 이용하고 응용 간 통신 및 상호 응용의 사례가 많아지고 있으며 이를 위한 기업 내의 기반 시스템 구축에도 웹 서비스의 이용이 활발해지고 있다. 이에 따라 인터넷을 이용한 웹서비스에서 다루어지는 정보의 품질에 대한 중요성이 증대되고 있으며, 다양하고 방대한 정보에 대한 보안 역시 점점 더 중요성을 띄고 있다. 따라서 본 연구에서는 웹서비스에 대한 표준 동향들을 고찰하고, 사용자들의 정보를 보호하기 위한 다양한 보안정책들을 분석하였다. 이를 기반으로 웹서비스의 보안성을 강화하기 위한 방안을 기술하였다.

주제어 : 웹, 웹 서비스, SOAP, 보안, WSDL

Abstract As the Internet has been growing, WWW(World Wide Web) based services were popularized and users using the service were increased excessively. Recently, the instances of communications between the applications and interaction applications using the Web services in the implementation of the business logics among the enterprises are spread widely. Therefore, it has been emphasized quality and security of web services. In this paper, we described standard trends for web servises. And we analyzed the security policies to protect user's informations. Eventually, We described a security enhancement method for web service.

Key Words : Web, Web service, SOAP, WSDL, Security

1. 서론

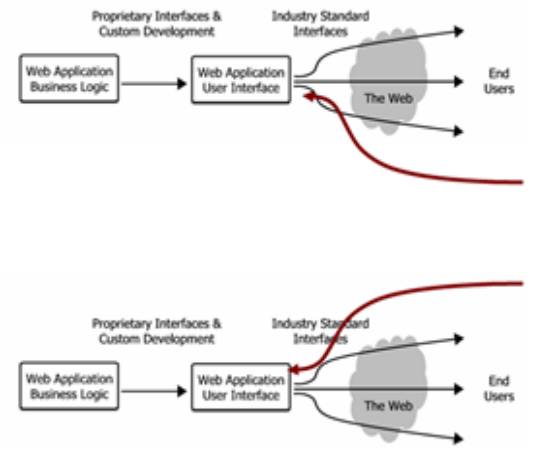
웹서비스는 서로 다른 컴퓨팅 환경에서 사용되는 모든 애플리케이션들이 직접 소통하고 실행될 수 있도록 동적 시스템 환경을 구현해 주는 소프트웨어 컴포넌트. 단순 객체 접근 프로토콜(SOAP), 웹 서비스 기술 언어(WSDL), 전역 비즈니스 레지스트리(UDDI) 등의 표준 기술을 사용하여 네트워크에 연결된 다른 컴퓨터 간의 분산 컴퓨팅을 지원하는 소프트웨어 및 기술들이다[2, 3]. 또한, 웹 서비스는 논리적 응용 프로그램의 단위로 데이터와 서비스를 다른 응용 프로그램에 제공하고, 응용 프로그램의 작성 시 HTTP, XML, SOAP와 같은 표준화된

Received 21 October 2013, Revised 15 November 2013
Accepted 20 December 2013
Corresponding Author: Seong-Hoon Lee(Baekseok University)
Email: shlee@bu.ac.kr

ISSN: 1738-1916

웹 프로토콜과 데이터 형식을 사용함으로써 운영 체계(OS) 등 특정 플랫폼과 상관없이 모든 컴퓨터 간 원활한 데이터의 흐름을 보장해 준다. 웹 서비스를 활용하면 어려운 프로그램 작성 언어를 배우지 않고도 간단하게 여러 가지 웹 서비스들을 조합하여 새로운 애플리케이션을 만들어 낼 수 있다

최근 기관 내부의 다양한 정보 시스템을 연계 및 통합하는 소프트웨어 인프라 구축에 높은 상호 운용성을 제공하는 웹서비스 기술이 적용되고 있다. 국내에서는 IT839의 융통합에 웹서비스가 채택되어 활용될 예정이다. 웹서비스를 통해 외부에 개방된 비즈니스 로직의 개별요소들을 적절히 조합함으로써 응용을 개발할 수 있으며, 이 과정에 응용과 서비스는 작은 기능의 많은 웹서비스가 조합되어 구현됨으로 동일한 기능을 수행하는 다양한 응용이 구성될 수 있다.



[Fig. 1] Web Service Structure

공공 부문 및 기업의 전산 시스템은 제어가 가능하여야 하나 웹서비스를 기반으로 응용을 구축하는 경우 서비스를 제공하는 제 3 공급자에 대한 제어가 쉽지 않으므로, QoS을 기반으로 한 예측 가능성을 제공하기 위해서는 웹서비스 품질에 대한 평가는 필수적으로 요구된다[1]. 또한 여러 형태로 구성되는 이종 서비스 모듈들의 통합과 더불어 일관된 보안 인터페이스를 제공할 수 있는 정책 및 방법에 대한 연구가 필요하다. 웹서비스 보안에 있어서 전자상거래의 신뢰성을 높이기 위해서는 인증(authentication), 허가(authorization), 디지털 서명(digital signature) 기능의 제공이 매우 중요하다[2, 7]. 웹서비스 환경과 같이 자원이 분산되어 있고 원격 실행 및 포워딩이 산재한 환경에서는 개인 정보의 노출 및 이를 이용한 각종 사기 등이 발생할 위험이 매우 높다. 시스템 접근, 서비스 요청 또는 상거래를 위한 신분 확인 절차를 투명하고 신뢰성 있게 구축하기 위해서 생체 정보를 포함한 사용자 인식을 위한 기술을 접목하는 것이 필수적이다.

웹서비스의 보급에 따라 서비스 품질은 서비스의 사용자에 중요한 선택 기준으로 등장하고 있으며, 웹서비스에 대한 품질에 대한 평가가 요구된다[1, 2, 3]. OASIS를 중심으로 웹서비스 품질에 대한 평가를 위한 모델에 대한 표준이 진행되고 있으며 결과로서 WSQM TC가 운영되고 있다[6, 7]. WSDL은 웹서비스의 기능적인 정보만을 서비스 사용자에게 제공하고 있으며[4], 웹서비스의 성능이나 안정성에 대한 요소가 중요해짐에 따라 WSDL의 확장을 통해 서비스 명세 내에 타임아웃, 응답시간 등의 QoS에 대한 정보를 포함하고자 하는 연구가 진행되고 있다.

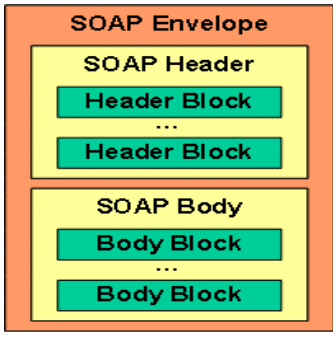
2. 웹서비스 동향

2.1 표준 동향

웹서비스에서는 WSDL 인터페이스에 의해서 기술된 서비스들 간에 메시지를 논리적으로 전달하는 메커니즘으로 SOAP을 사용하는 것이다. SOAP(Simple Object Access Protocol)은 웹서비스에 있어서 가장 오래된, 충분히 성숙된, 가장 중요한 프로토콜이다[3, 5, 8]. SOAP 메시지는 모든 SOAP 모델이 기반하고 있는 개념적 기초를 제공한다. 또한 SOAP 메시지는 웹서비스 간에 설정되어 있는 전달 프로토콜(transport protocol)을 통해서 전송된다.

애플리케이션 페이로드는 메시지의 바디 부분에 저장되어 이동되고, 부가적인 프로토콜 메시지(애플리케이션 데이터만 전달될 때에는 표시되지 않거나 없어도 되는 임의 사항임)는 헤더 블록에 저장되어 이동한다. 이는 SOAP 처리 수준에서 애플리케이션 수준의 메시지와

SOAP의 헤더 공간에 저장되어 이동하는 좀 더 상위 수준의 웹서비스 프로토콜(예, 트랜잭션 처리나 보안)을 구분하도록 해준다. SOAP은 웹서비스의 바인딩(binding) 프로토콜로서 XML로 인코딩한 자료를 전송하는 일정한 방법을 정의하는 표준 프로토콜이다. XML 포맷을 사용하여 웹서비스의 메소드를 호출할 수 있는 기능을 제공한다. SOAP은 분산 환경에서 정보 교환에 사용하는 경량(lightweight)의 프로토콜로서 텍스트 기반의 XML을 프로토콜로 사용함으로써 하드웨어 플랫폼, 운영체제, 프로그래밍 언어 및 네트워크 하드웨어 플랫폼에 종속적이지 않은 장점을 갖는다.



[Fig. 2] SOAP Message Structure

SOAP 메시지는 단방향의 메시지의 전송 기능을 제공한다. SOAP 코딩은 모든 단순형은 XML 스키마 명세로 구성되어지며 이들 자료 형의 표현을 위한 두 개의 문법을 제공한다.

두 번째 모델인 WSDL(Web Service Description Language)은 웹서비스가 제공하는 서비스에 대한 개요를 설명하는 XML 문서로서 IBM의 NASSL과 마이크로소프트의 SDL의 명세가 통합되어 만들어졌으며 W3C에 의해 표준화가 주도되고 있다[3, 4].

모든 WSDL 인터페이스의 기본은 인터페이스 뒤에 있는 서비스가 보내고 받고자 하는 메시지의 집합에 있다. 비록 WSDL이 다른 스키마 언어를 사용하는 것을 허용하긴 하지만, 하나의 메시지는 XML 스키마 유형을 사용하여 정의되는 것이 일반적이고, 내용에 대한 접근성을 높이기 위해서 몇 개의 논리적인 부분으로 나눠진다. portType은 웹서비스가 모양을 갖추기 시작하는 곳이라고 생각되는 장소이다. portType은 웹서비스가 될 것이라고 생각하는 연산의 집합이다. 그러나 이 시점에서 연산은 여전히 추상적 언어로 정의되어 있고, 몇 개의 메시지 교환 집합이 연산으로 묶여지게 된다.

WSDL 인터페이스의 binding 부분은 추상적으로 정의된 메시지와 연산을 어떻게 물리적인 반송 프로토콜에 매핑하는가를 서술한다. 특정한 프로토콜로 바운드될(그래서 결국은 네트워크에서 사용 가능하게 되는) portType의 연산은 WSDL 명세의 바인딩 부분에서 바인딩 관련 정보를 추가하여 확장된다. 즉, WSDL은 SOAP, HTTP GET과 POST, MIME을 지원하며, 본래의 portType 선언에 대한 프로토타입 특화된 버전을 제공하는 것이다. 마지막으로, port는 특정한 바인딩을 참조하는 정보와 어드레싱 정보가 함께 쌓여져 service 요소를 구성하고 물리적으로 네트워크 접속 가능한 최종 형태의 웹서비스가 된다.

WSDL 서술의 추상적 부분은 types, message, portType 요소이고, 실제 요소는 binding과 service이다. 추상과 실재 부분 사이의 분리는 유용한 개념인데, 인터페이스 디자인을 궁극적 배포(deployment) 환경과 분리할 수 있으며 WSDL에서는 추상 정의만을 사용하기 때문이다.

마지막 모델로서, UDDI은 웹서비스를 출판하고 검색하기 위한 웹서비스의 레지스트리이며 프로토콜을 의미한다. 웹서비스는 다른 기업에서 제공하는 소프트웨어 기능을 플랫폼에 독립적이면서 개방적으로 접근 활용할 수 있도록 하는 표준화된 수단이기 때문에, UDDI 또한 개방적이어야 하고, 플랫폼에 독립적이어야 할 뿐만 아니라 표준화 되어야 한다. UDDI 내의 정보는 XML로 표현된 자료 구조의 인스턴스들로 구성된다. UDDI 노드에 의하여 저장되고 관리되는 자료란 바로 이 자료 구조에 의하여 생성된 인스턴스들이다. 이 자료 구조를 엔티티라고 하는데, UDDI 명세에는 네 개의 엔티티가 정의되어 있다.

2.2 보안 필요성

최근 기업에서 고객의 애플리케이션에 사용 사례들이 늘어나게 되면서 보안과 관련된 비기능적 요구사항들이 요구되고 있다. 이러한 요구사항으로는 다음과 같은 것들이 있다.

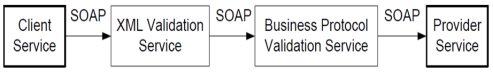
- 고객과 비즈니스 파트너 간 통신이 인터넷 상에 흐를 때 제 삼자는 이를 볼 수 없어야 한다.

- 메시지가 어디에서 왔는지를 결정할 수 있고 보낸 사람이 승인된 것인지를 확인할 수 있다.
- 전송되는 데이터가 조작되는 일이 없도록 해야 한다.

첫 번째 요구사항은 HTTPS/SSL 전송 보안을 사용하여 해결될 것이다. SSL(Secure Sockets Layer)은 웹 사이트의 암호화를 위해 사용되는 기술로서 웹 서버의 요청 및 응답을 위한 메시지를 암호화하여 전달하고 이를 복호화해서 사용한다. SSL 프로토콜은 두 사용자 사이에서는 점대점 데이터 보호를 제공하지만 다음과 같은 단점이 있다.

- 다수의 사용자 사이에서는 종단간(end-to-end) 데이터 보호를 지원하지 않는다.
- 메시지를 세그먼트 단위로 선택적으로 암호화 할 수 없다.

웹서비스 기반의 응용에서 정보 또는 데이터가 근원지에서 목적지로 이동할 때 데이터가 도난당하거나 변경될 위험은 언제나 존재한다. 이와 같은 보안 위험은 웹서비스 트랜잭션에도 적용된다. SOAP 개발 초기에는 SOAP이 HTTP 기반 프로토콜로 표시되어 HTTP 보안이 SOAP에 충분하였으나, 여러 중개자(intermediary)를 통해 서비스가 운영되는 웹서비스에 활용됨에 따라 보안을 위한 확장이 필요하게 된다. 현재의 SOAP 모델은 중간 계층에서 완전한 신뢰성 보장이 어렵고, 통신 링크 중 어느 하나라도 안전하지 않을 경우, 종점간의 보안 취약점을 갖는다.



[Fig. 3] Weakness of web service protection

[그림 3]는 웹서비스의 보안 취약성을 보인 것으로 SOAP 메시지의 요청을 받고 있는 XML 메시지를 검증하는 서비스(XML Validation Service)는 메시지 내의 요소에 대한 자료의 유형에 대한 검증을 수행하거나 비즈니스 프로토콜 검증 서비스에서 자료에 대한 처리를 위한 검증 서비스(Business Protocol Validation Service)에서는 해당 메시지 내의 개인의 사적인 정보가 노출되는 위험을 갖는다.

이러한 웹서비스 보안 취약성의 해결을 위해서는 보안을 요구하는 민감한 정보에만 보안 서비스를 적용하는 어플리케이션 계층의 보안이 필요하다. 다수의 사용자들 사이에서 전체적인 관점에서 보안을 제공하지 못하는 것이 다수의 엔티티가 관계되는 웹서비스 환경에서의 SSL의 단점이다.

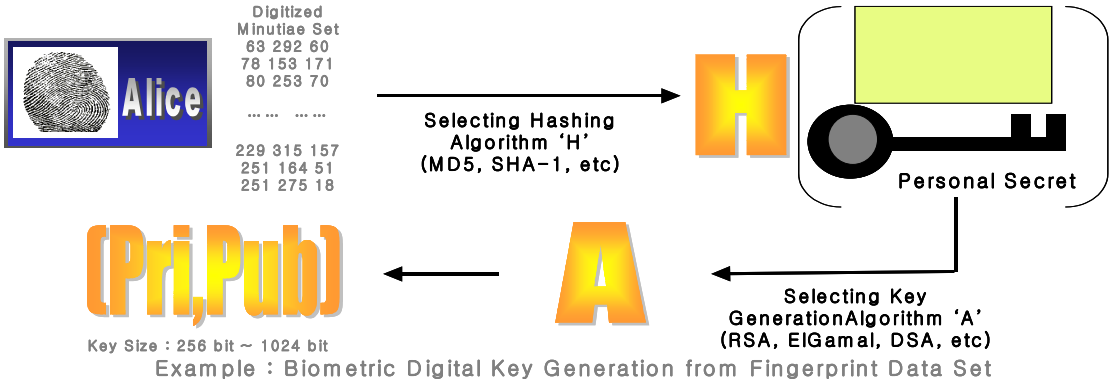
3. 보완성 강화 방안

본장에서는 바이오 정보를 이용한 사용자 인식의 필요성과 관련 연구를 살펴본다. 기존의 사용자 인식 방법은 크게 두 가지로 구분할 수 있다. 패스워드 기반의 인증방식 및 카드 기반의 사용자 인식 방법은 해킹 및 분실, 도난 등의 위험에 대해서 안전하지 못하다. 또한, 악의적인 의도를 갖는 사용자에 의한 카드 변조 및 위조 등의 위험요소가 존재한다[9, 10]. 따라서, 사용하기 편하고 보안성이 높은 새로운 사용자 인식 방법에 대한 연구가 필요하며 사용자의 고유한 바이오 정보를 이용한 인식 방법이 주요 대안으로 연구되고 있다.

바이오 인식(biometric recognition)은 개개인 고유의 바이오 정보를 인식하기 위한 방법을 의미하며, 기술적으로 지문, 정맥, 홍채, 망막, 얼굴, 서명 및 음성 등 다양한 개인 고유의 정보를 추출하여 미리 등록되어 있는 데이터와 비교하는 기술을 의미한다. 따라서 개인의 식별(identification) 혹은 인증(verification)을 위한 목적으로 사용되며, 더 나아가서 접근 제어를 위한 기술로 활용될 수 있다.

바이오 인식 시스템은 사용자의 바이오 데이터를 추출하여 특징점을 구하고 이를 이용하여 데이터베이스에 저장된 템플릿들과 비교하는 패턴 인식 시스템이다. 이러한 바이오 인식 시스템의 각 구성 요소는 다음과 같은 역할을 수행한다.

1) 센서(sensor): 센서를 이용하여 사용자의 바이오 데이터를 캡춰하고 디지털 이미지로 저장한다.
2) 특징점 추출(feature extraction): 이미지 처리 및 가공 모듈에서는 센서로 입력받은 원시 이미지의 노이즈 제거, 지문 융선 복원, 세선화 등의 작업을 한

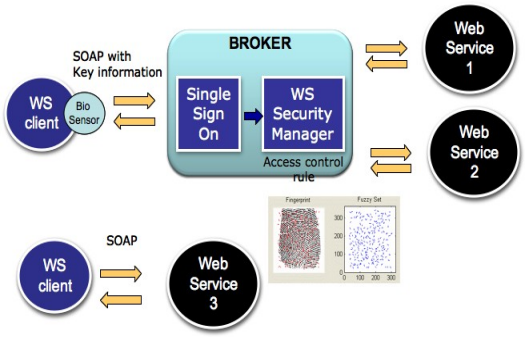


[Fig. 4] Bio-based Identification

다. 가공된 이미지로부터 좌표 및 각도 정보를 갖는 특징점을 추출한다. 지문(fingerprint)의 경우에 특징점은 지문 융선이 끝나는 단점과 갈라지는 분기점으로 구성된다.

3) 매칭(matcher): 인증 받고자 하는 사용자의 특징점과 데이터베이스에 저장된 템플릿(특징점)을 비교하여 매칭 스코어를 계산한다. 매칭 스코어를 토대로 사용자에 대한 인증 및 식별을 수행한다.

4) 데이터베이스(stored templates): 사용자들의 바이오 정보를 입력받아 특징점들로 구성된 템플릿을 저장하고 검색하기 위한 용도로 사용된다.



[Fig. 5] Bio recognition based SSO model

본 연구에서 제시하는 보안성 강화 방안은 [그림 4]과 같이 요약할 수 있다. 지문 정보에서 특징점에 해당하는 지문 템플릿을 추출하고 이를 이용하여 개인키와 비밀키 쌍을 생성하는 것이다. [그림 5]에서와 같이 바이오 기반 통합 로그인(Single Sign On)을 통하여 여러 가지 웹서비스들을 사용하는 구조에서는 사용자 인증 및 식별이 매우 중요한 요소가 된다. 바이오 정보는 사용자 마다 고유하기 때문에 다른 사용자에게 대여할 수 없는 특징을 가지며 웹서비스 환경과 같이 사용자를 대면할 수 없는 상황에서 가장 확실한 사용자 인증 수단이며 보안성이 높다고 할 수 있다.

4. 결론

웹서비스는 서로 다른 컴퓨팅 환경에서 사용되는 모든 애플리케이션들이 직접 소통하고 실행될 수 있도록 동적 시스템 환경을 구현해 주는 소프트웨어 컴포넌트. 단순 객체 접근 프로토콜(SOAP), 웹 서비스 기술 언어(WSDL), 전역 비즈니스 레지스트리(UDDI) 등의 표준 기술을 사용하여 네트워크에 연결된 다른 컴퓨터 간의 분산 컴퓨팅을 지원하는 소프트웨어 및 기술들이다. 웹서비스 환경과 같이 자원이 분산되어 있고 원격 실행 및 포워딩이 산재한 환경에서는 개인 정보의 노출 및 이를 이용한 각종 사기 등이 발생할 위험이 매우 높음으로 시스템 접근, 서비스 요청 또는 상거래를 위한 신분 확인 절차를 투명하고 신뢰성 있게 구축되어야 한다.

이에 본 연구는 웹서비스의 다양한 모델들에 대한 표준 동향을 알아보고, 웹서비스에서 다루어지는 데이터들의 중요성이 증대함에 따라 보안의 필요성 또한 증대하고 있는 상황을 감안하여 보안 필요성을 기술하였다. 마지막으로 웹서비스에서의 보안성 강화를 위한 방안을 다루었다.

REFERENCES

[1] M.P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," CACM, Vol. 46, No. 10, Oct 2003.

[2] W3C, Web Services Architecture, http://www.w3.org/TR/2003/WD-ws-arch-20030808/

[3] F. Curbera et. al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI," IEEE Internet Computing, Vol. 6, No. 2, pp. 86-93, March/April 2002.

[4] W3C, Web Services Description Language(WSDL) 1.1, 2001, http://www.w3c.org/TR/wsdl

[5] W3C, Simple Object Access Protocol (SOAP) 1.1. 2000, http://www.w3c.org/TR/SOAP.

[6] P. Baglietto, M. Maresca, A. Parodi and N. Zingirian, "Deployment of Service Oriented Architecture for a Business Community," In Proc. of the Sixth International ENTERPRISE DISTRIBUTED OBJECT COMPUTING (EDOC'02), 2002.

[7] T. Erl. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.

[8] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. Web Services Concepts, Architectures and Applications Series: Data-Centric Systems and Applications. Addison-Wesley Professional,, 2002.

[9] Jin-Su Park, "Enterprise Web Service Development", HongRung Publishing Company, 2004.

[10] A. K. Jain, A. Ross, and S. Pankanti, "A prototype hand geometry-based verification system," 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication, Washington D.C., pp. 161-171, March 22-24, 1999.

[11] N. Ratha, J. Connell, and R. Bolle, "Enhancing Security and Privacy in Biometrics-based Authentication Systems," IBM Systems Journal, Vol. 40, No. 3, pp. 614-634, 2001.

[12] N. Ratha, J. Connell, and R. Bolle, "Cancelable Biometrics," Biometric Consortium, 2000.

[13] J. Cambier, et. al., "Application-Specific Biometric Templates," Proc. of AutoID, pp.167-171, 2002.

이 성 훈(Seong-Hoon Lee)

- 1998년 2월 : 고려대학교 컴퓨터학과 (이학박사)
- 1998년 3월 ~ 현재 : 백석대학교 정보통신학부 교수.
- 관심분야 : 분산 시스템, 무선 통신, 유전 정보, 웹서비스
- E-Mail : shlee@bu.ac.kr