Web Services and Interoperability : Security Challenges

S K Sharma G K Sharma P N Srivastava

Abstract

The Web services framework intends to provide a standards-based realization of the service-oriented architecture (SOA) over Internet, which has emerged in response to a fundamental shift from program-to-consumer (B2C) to program-to-program (B2B) interactions. Fully Interconnected enterprises are being replaced by business networks in which each participant provides the others with specialized services. This new service architecture defines a set of requirements that distinguish SOA from other services architecture. Security and Web services are consistently reported among the top technologies of interest to business. Concerns about the security technology are major deterrent to companies considering use of the technology. This paper attempts to explain the new Web Services security and mentions the main initiatives and their respective specifications.

Keywords : Web Services, XML, XML-Signature, XML-Encryption, WS-Security.

0. Introduction

Everyone knows roughly what a "Web Service" is, but there is no universally accepted definition. The definition of web service has always been under hot debate within the W3C Web Services Architecture Working Group[1]. The precise definition of Web Services is still evolving as witness by the various definitions in the literature. One such the definition is that a Web Services is convergence between SOA and web with at least the following additional constraints:

- Interfaces must be based on Internet protocols such as HTTP, FTP, and SMTP.
- Except for binary data attachment, messages must be in XML format.

It is also defined as the web applications that are self-contained, self-describing, modular that can be published, located, and invoked across the web. They perform functions, which can anything from simple request to complicated business process [2]. W3C (World Wide Web consortium) define a Web service as a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [3].

At a minimum, web services can be any piece software that makes itself available over the Internet using standardized web services messaging system and interface [4]. This paradigm has the potential to deliver many benefits for business. Some of these are:

- Developer will be able to respond quicker with the demanding business needs to link up partners or to provide access to existing business assets within the firewall.
- Functionality from the heterogeneous development platform (.NET, CORBA, J2EE) can be quickly integrated into business applications.

- ✍ The web services paradigm, being platform agnostic, solves the basic EAI (Enterprise application integration) problem of having uniform interface to business applications. In the future, packaged software application will expose service-oriented interface based on web service standard.
- ✍ Investments made in your internal infrastructure can be leveraged to provide new channel for providing service to your customers.
- ✍ Current applications that have developed in any language can be exposed as web services quickly using web services tools, web services tools will be available on all development platforms.
- ✍ The services-oriented paradigm that web services builds on promoters the use of clean interfaces that allow a business asset being wrapped by an interface to be replaced, changed, or outsourced as business needs requires without affecting customers or business process that depend on the assets.
- ✍ Developers will be able to use number of web services provided be third parties to deliver more powerful and integrated business solutions.

Due to these immediate benefits, most of IT department are implementing this technology with the higher-priority objective of making them operable leaving aside, at least until later stages, the problem related to security. XML, eXtensible Markup Language, lies at the core of web services interoperability, and enables it to provide a language-neutral and platform-independent way of linking applications [5].

However, these characteristics also present new security threats and challenges. Security is being considered as the biggest roadblock facing widespread implementation of web services technologies.

The security solutions that are commonly implemented in today's web-based applications, such as SSL (Secure Socket Layer), do not provide a sufficient security infrastructure for web services [6]. In contrast to web site applications that require a security solution between the client browser and the web server, the web services applications can involve calling one or more intermediary services, thus requiring more comprehensive security solution. If a transaction passes through intermediary systems the integrity of the data and security of information that flows with it may be lost [7]. Also, the user credentials cannot be easily passed through each stop in the transaction chain. To solve the security issues in the above scenarios, web service security architecture requires mechanism that provides security for the entire transaction.

The Web Services reference Model

Interactions among Web Services involve three types of participants: **Service Providers, Service registry, and service users** (Fig1.0).

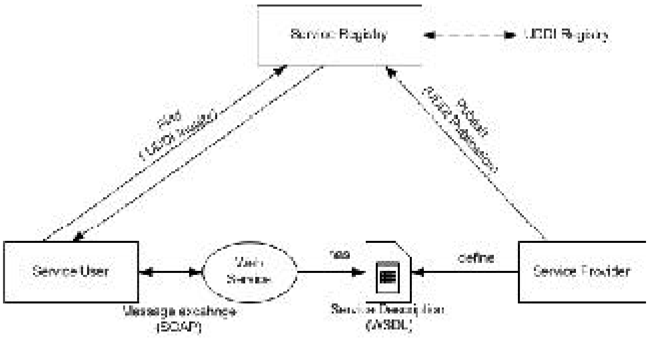

Fig 1.0

Service Providers are the parties that offer services. They define descriptions of their services and publish them in the service registry, a searchable repository of service descriptions. Each description contains details about the corresponding service such as its data types, operations, and network location. Service users use a find operation to locate services of interest. The registry returns the description of each relevant service. The user uses this description to invoke the corresponding web service.

Three major standardization initiatives have been submitted to the W3C consortium to support interactions among Web Services.

- WSDL (Web Service Description Language) : WSDL [8] is an XML-based language for describing operational features of Web Services. WSDL descriptions are composed of interface and implementation definitions. The interface is an abstract and reusable service definition that can be referenced by multiple implementations. The implementation describes how the interface is implemented by a given service provider.
- UDDI (Universal Description, Discovery and Integration) : UDDI [9] defines a programmatic interface for publishing and discovering Web Services. The core component of UDDI is the business registry, an XML repository where businesses advertise services so that other businesses can be find them. Conceptually, the information provided in a UDDI business consists of white pages (contract information), yellow pages (industrial categorization), and green pages (technical information about services).
- SOAP (Simple Object Access Protocol) : SOAP [10] is a lightweight messaging framework for exchanging XML formatted data among Web Service. SOAP can be used with a variety of transport protocols such as HTTP, SMTP, and FTP. A SOAP message has a very simple structure: an XML element, the header includes features such as security and transactions. The second element, the Body includes the actual exchanged data.

2. Interactions in Web Services

Web Services allow interactions at the communication layer by using SOAP as a messaging protocol. The adoption of an XML-based messaging over well-established protocols (e.g. HTTP, SMTP, and FTP) enables communication among heterogeneous systems.

At the content layer, Web Services use WSDL language. WSDL recommends the use XML Schema as a canonical type system (to associate data types to message parameter). However, the current version of WSDL does not model semantic features of Web Services. For example, no constructs are defined to describe document types (e.g. whether an operation is a request for quotation or a purchase order).

Web Services are still at a maturing stage. Hence, they still lack the support for interactions at the business process layer. To date, enabling interaction among Web Services has largely been an ad hoc process involving repetitive low-level programming. Standardization efforts such as BPEL4WS (Business Process Execution Language for Web Services) are underway for enabling the definition of business process through Web Services composition.

WSDL does not currently include operations for monitoring Web Services such as checking the availability of an operation or the status of a submitted request. Additionally, neither UDDI nor WSDL currently define quality of service parameters such as cost and time. In terms of adaptability, changes may occur in operation signatures (e.g. name), messages (e.g. number of parameters, data type), service access (e.g., port address), and service and operation availability. The process with changes is currently ad hoc and manually performed.

3. Main Web Services security Issues

Security in Web Services needs to be addressed at different levels including communication, description, and firewall. Some of the major security issues that web services technologies must address:

- Authentication: Any Web Services that participate in an interaction may be required to provide authentication credentials by the other party (e.g. a pair username/password or an X.509 certificate).
- Authorization: Web Service should include mechanisms that allow them to control access to the services being offered. They should be able to determine who can do what and how on their resources.
- Confidentiality: Keeping the information exchanged among Web Services nodes secret is another of the main properties that should be guaranteed in order to consider the channel secure.
- Integrity: This property guarantees that the information received by a Web Service remains the same as the information that was sent from the client.
- Non-repudiation: In the Web services world, it is necessary to be able to prove that a client utilized a services and that service processed the client request. This security issues is covered by Digital Signatures.
- Availability: The need to take care of the availability aspects for preventing denial-of-service attacks or to arrange redundancy systems is a crucial point in Web Services technology.
- End-to-End Security: Network topologies require end-to-end security to be maintained all across the intermediaries in the message's path. " When data is received and forwarded on by an intermediary beyond the transport layer, both the integrity of the data and any security information that flows with it may be lost" [7]. This forces any upstream message processors to rely on the security evaluations made by previous intermediaries and to completely trust their handling of the content of messages.

In addition, the above mentioned issues, which are inherited from the distributed computing classical scheme, Web Services should also address the issues arises from the new threats created by its own nature such as:

- Availability of higher number of standard specifications;
- Most of specifications are in draft state;
- XML standard format needed to structure the security data;
- Application-level, end-to-end and just one-context-security communications;
- Interoperability of the requirement and online security elements;
- Audit, automatic and intelligent contingency processes aimed at being machine-to-machine interactions not controlled by humans;
- Online availability management in critical business process;

4. Core Web Services Security Specifications

The core Web Services specifications are XML, SOAP, WSDL, UDDI. These specifications have been broadly adopted by the industry, and constitute the basic building blocks on which Web Services are being designed and implemented. The bad news is that these four operative services specification allows creation of Web Services but they do not say anything about how to secure them.

XML and SOAP both specifications do not say anything about how to obtain integrity, confidentiality, and authenticity of the information that they respectively represent and transport.

Numbers of questions are associated with the UDDI and WSDL specifications such as: “Is the UDDI registry located in a trustworthy location? How can we be sure that the published data has not been maliciously manipulated? Was the data published by the business it is supposed to have been published by? Can we rely on the business that published the services? Are the services available at any moment? Can we trust the transactions that are produced from the execution of the services?” As we can see from all these questions, an in-depth analysis of the security problems that UDDI and WSDL architecture implies is needed [9].

Two new security initiatives designed to both account for and take advantage of the special nature of XML data are XML Signature and XML Encryption. Both are currently progressing through the standardization process. XML Signature is a joint effort between the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF), and XML Encryption is solely W3C effort. In addition, XML-key Management System standard is associated with these two standards.

<<Specification>>
XML Encryption
<<Specification>>
XML Digital
Signature
<<Specification>>
XML Key
Management system

Fig 2.0

4.1 XML-Digital Signature

It defines how to digitally sign XML content and hot to represent the resulting information to an XML schema. A digital signature grants information integrity and non-repudiation [11]. Thus, for example entity cannot deny the authorship of digitally signed documents.

According to the XML Digital Signature specification, digital signature can be applied any kind of digital content, including XML. It has ability to sign only specific portion of the XML tree rather than the complete document. This is important when a single XML document may need to be signed by multiple times by a single or multiple parties. This flexibility can ensure the integrity of certain portion of an XML document, while leaving open the possibility for other portions of the document to change. The signature validation mandates that the data object that was signed be accessible to the party that interested in the transaction. The XML signature will generally indicate the location of the original signed object.

4.2 XML Encryption

It provides a model for encryption, decryption, and representation of full XML documents, single XML elements in an XML document, contents of an XML element in an XML document, and arbitrary binary content outside and XML document [11].

XML encryption solves the problem of confidentiality of SOAP messages exchanged in Web Services. It describes the structure and syntax of the XML elements that represent encrypted information and it provides rules for encrypting/decrypting an XML document (or part of it).

The specification states that encrypted fragments of a document should be replaced by XML elements specifically defined in the recommendation. In order to recovery the original information, a decryption process is also specified.

4.3 XML Key Management Specification (XKMS)

It is an XML-based way of managing the public key infrastructure (PKI), a system that uses public-key cryptography for encrypting, signing, authorizing and verifying the authenticity of information in the Internet [12]. It specifies protocols for distributing and registering public keys, suitable for use in conjunction with the proposed standard for XML Signature and XML Encryption.

XKMS allow implementers to outsource the task of key registration and validation to a “trust” utility. This simplifies implementation since third party does the actual work of managing public and private key pairs and other PKI details.

5. WS-Security Family Specification

IBM and Microsoft with other major companies have defined Web Services security models that guarantee end-to-end communication security. The center of these specifications is composed of WS-*Policy*, Ws-*Trust*, WS-*Privacy*, WS-*SecureConversation*, WS-*Federation*, WS-*Authorization*, and WS-*Security*.

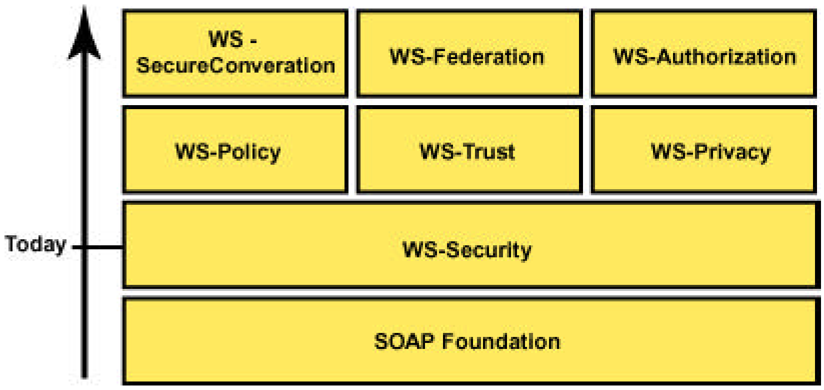

Fig 3.0[4]

5.1 WS-Security

The most important work in this area is *WS-Security* specifications from IBM, and Microsoft, and VeriSign [13]. The three companies jointly developed the new specification, known as *WS-Security*, and have submitted it to two major standardization organizations: W3C, World Wide Web Consortium, and the OASIS, Organization for the Advancement of Structured Information Standards.

WS-Security describes enhancements to SOAP messaging to provide *quality of protection* through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

WS-Security is placed at the base of the security specification pile. Other specifications that directly relate to security issues are being developed based on WS-Security. In the protocol stack and left on top of the WS-Security, we find WS-policy, WS-trust, and WS-privacy. WS-Policy will describe how senders and receivers can specify their requirements and capabilities. WS-Trust defines XML Schema as well as protocols that allow security tokens to be accessed, validated, and exchanged. WS-Privacy will describe how organizations state the privacy policy so that incoming request make claim about the sender's adherence to these policies.

The top layer consist three protocols, which are follow-on specifications. WS-SecureConversation will describe how a Web service can authenticate requester messages, how requesters can authenticate services, and how to establish mutually authenticated security contexts. It is designed to operate at the SOAP message layer so that the messages may traverse a variety of transports and intermediaries. This does not preclude its use within other messaging frameworks. In order to further increase the security of the systems, transport level security may be used in conjunction with both WS-Security and WS-SecureConversation across selected links. WS-Federation specifications define how to construct federated trust scenarios using the WS-Security, WS-Policy, WS-Trust, and WS-SecureConversation. It also defines the mechanism for managing trust relationships. WS-Authorization specifications describe how access policies for a Web service are specified and managed. In particular it will describe how claims may be specified within security tokens and how these claims will be interpreted at the endpoint.

5.2 Security Assertion Markup Languages (SAML)

SAML is an Extensible Markup Language standard (XML) [2] that supports Single Sign On. SAML allows a user to log on once to a web site and conduct business with affiliated but separate web sites. SAML can be used in B2B and B2C transactions.

There are three basic SAML components: Assertions, protocol, and binding. Assertion can be one of three types: authentication, attribute, and authorization. Authentication assertion validates the identity of the user. The attribute assertion contains specific information about the user. While, the authorization assertions identifies what the user is authorized to do.

The protocol defines how SAML request and receives assertions. There are several available binding for SAML. There are bindings that define how SAML message exchanges are mapped to SOAP, HTTP, SMTP and FTP among others. OASIS is the body developing SAML.

5.3 XACML: Communicating Policy Information

XACML is an Extensible Markup Language standard (XML) based technology, developed by OASIS for writing access control policies for disparate devices and application. It includes an access control language and request/response language that let developers write policies that determine what users can access on a network or over the Web. XACML can used to connect disparate access control policy engines.

5.4 Liberty Alliance Project

The Liberty Alliance Project is led by Sun Microsystems, and it s purpose is to define a standard federation framework that allows services such as Single Sign-On [14].

Thus, the intention is to define an authentication distributed system that allows intuitive and seamless business interactions. This purpose is the same as those of the WS-Federation specifications and Passport's .NET technology. Once again, this is another example of the previously so-called overlap problem in Web Services Security solutions.

6. Summary of the Current Web Services Standards

Authentication	WS-Security, WS-Trust (Draft), XKMS, SAML, Liberty Alliance Project, WS-Federation (Draft)
Authorization	XACML, WS-Authorization, (Draft)
Confidentiality	XML-Encryption, WS-Security
Integrity	XML-Digital Signature
Non-repudiation	XML-Digital Signature, WS-Security
Security Policy	WS-Policy, WS-SecurityPolicy (Draft), XACML
Trust authority	WS-Trust (Draft); XKMS
Security Context	WS-SecureConversation (Draft)
Delegation/Proxy	WS-Trust (Draft), Delegation has not yet been fully addressed
Privacy	WS-Privacy

7. Conclusion

In spite of the amount of specifications, there are many unresolved security issues that will have to be addressed in the future. The explosion of specifications and concepts, and lacking of a global standardization initiative is causing overlapping solutions to similar problem. In, addition, the problems relating to security vulnerabilities, which would be introduced in complex WS implementation using different security tokens, have not been sufficiently addressed. This fact will require an extra effort in the future not only for the specifications to unify and make themselves interoperable but also for industry to adopt and easy implement them.

8. References

1. WSAS Web Services Architecture Draft 8 August 2003 (2003). Seehttp://www.w3.org/TR/2003/WD-ws-arch-20030808/

2. SAML Specifications V1.1 –http://www.oasis-open.org/committees/download.php/791/sstc-acml-1.1-cs-02.zip

3. Web Services architecture W3C working group note 11th Feb, 2004 See http://www.w3.org/TR/ws-arch/.

4. WS-Security Specifications V1.0 – Chris Kaler (Editor). *WS-Security, Version 1.0*. An IBM, Microsoft and VeriSign joint specification. April 5, 2002.http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

5. Gottschalk K., Graham S., Kreger H., and Snell J., Introduction to Web Services architecture IBM System journal, Volume 41, Number 2, 2002 http://www.research.ibm.com/journal/sj/41/gottschalk.html.

6. Mark Curphey (OWASP), et. al. A Guide to Building Secure Web Applications and Web Services. Version 1.1. Sep. 22, 2002. http://www.owasp.org/guide/

7. Security in a Web Services World: A Proposed Architecture and Roadmap, IBM/Microsoft White paper – http://www-106.ibm.com/developerworks/security/library/ws-secmap/?dwzonw=security

8. WSDL Web Service Description Language (WSDL) 1.1 – W3C Note Mach 2001. See http://www.w3.org/TR/wsdl

9. UDDI version 3.0.1 – UDDI spec Technical committee Specification 14 October 2003. See. http://uddi.org/pubs/uddi-v3.0.1-20031014.htm.

10. Don Box (DevelopMentor), et. al. SOAP: Simple Object Access Protocol 1.1 W3C Note. May 8, 2000.http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

11. Ed Simon, Paul Madsen and Carlisle Adams. An Introduction to XML Digital Signatures. August 8, 2001. http://www.xml.com/pub/a/2001/08/08/xmldsig.html

12. XML Key Management Specification 2.0 (XKMS) W3C Working Draft 18 March 2002, http://www.w3.org/TR/xkms2/

13. Chris Kaler (Editor). WS-Security, Version 1.0. An IBM, Microsoft and VeriSign joint specification. April 5, 2002. http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

14. Liberty Alliance Project Specification Archive V.1.1 –http://www.projectliberty.org/specs/archive/v1_1/index.html

About Authors

Mr. S K Sharma is Scientist-B, leader of Networking, testing and Quality control group in Information and Library Network Centre, UGC-INFLIBNET Ahmedabad. He has a M.Sc. in Physics and Master of Computer Application (MCA). He has nearly 7 years rich experience in IT and published papers in national conference/journals. Currently he is doing research in Web Services Security. His major research interests are distributed computing, Network security and management.

E-mail : sksharma@inflibent.ac.in

Dr. G. K. Sharma is professor and Head of Information Technology Group at Indian Institute of Information Technology & Management (IIITM), Gwalior. He has more than 20 years of work experience in both the research and academic world. Prior to IIITM, he was Professor & Head of the Department of Computer Science & Engineering at Thapar Institute of Engineering & Technology (TIET), Patiala. Dr. Sharma contributed more than 27 publications at the national and international levels. He did his Master's and Ph.D. in Electronics & Computer Engineering from University of Roorkee (now, Indian Institute of Technology), Roorkee.His area of interest is Distributed Computing.

Prof. P. N. Srivastava is Head of Dept. Mathematical Science and Computer Application and Director of Institute of Information Technology, Bundelkhand University, Jhansi. Dr. Srivastava has 40 years of rich teaching and 35 years research experience. He has guided 15 research scholars and contributed more than 50 research papers at National and International Level. His research areas are special functions, operations research, cryptography etc.

E-mail : pn_shrivastava@yahoo.com

.