Journal of Advanced Computing and Communication Technologies ***(ISSN:* 2347 - 2804*)***
Volume No.2 Issue No. 1,Februry 2014

Review Paper On Web Service Security

By

1st Prachi Labhane, 2nd Prof. Khushboo Saxena,
1st Department of Information Technology, Technocrats Institute of Technology, Bhopal, India
2nd Department of Information Technology, Technocrats Institute of Technology, Bhopal, India
prachi.3010@gmail.com, sxn.khushboo@gmail.com,

ABSTRACT:

A web service is defined as a software system designed to support interoperable machine-to-machine interaction over a network. Put in another way, Web services provide a framework for system integration, independent of programming language and operating system. Web services are widely deployed in current distributed systems and have become the technology of choice. The suitability of Web services for integrating heterogeneous systems is largely facilitated through its extensive use of the Extensible Markup Language (XML). Thus, the security of a Web services based system depends not only on the security of the services themselves, but also on the confidentiality and integrity of the XML based SOAP messages used for communication. Recently, Web services have generated great interests in both vendors and researchers. A web service, based on existing Internet protocols and open standards, and provides a flexible solution to the problem of application integration. This paper provides an overview of the web services, web service security and the various algorithms used for encryption of the SOAP messages.

Keywords:

Web service, Web services security, Web services security standards.

1. INTRODUCTION

A *web service* is a network accessible interface to application functionality, built using standard Internet technologies illustrated in Figure 1.

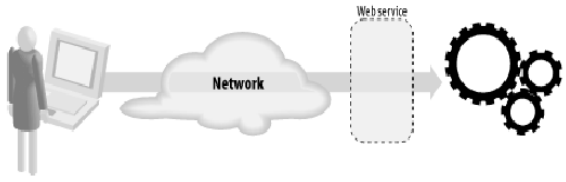

Figure 1. A web service allows access to application code using standard Internet Technologies

In other words, if an application can be accessed over a network using a combination of protocols like HTTP, XML, SMTP, or Jabber, then it is a web service. Despite all the media hype around web services, it really is that simple. **A**WEB service is defined as a software system designed to support interoperable machine-to-machine interaction over a network. Put in another way, Web services provide a framework for system integration, independent of programming language and operating system.
A web service is an interface positioned between the application code and the user of that code as shown in figure 2. Web service acts as an abstraction layer, separating the platform and programming-language-specific details of how the application code is actually invoked. This standardized layer means that any language that supports the web service can access the application's functionality.

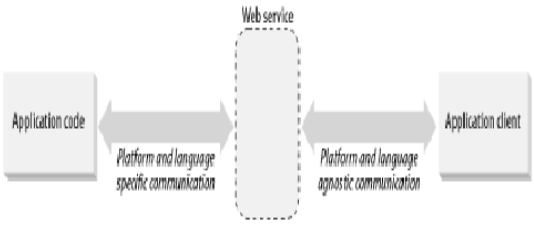

Figure 2. Web services provide an abstraction layer between the application client and the application code

The web services that we see deployed on the Internet today are HTML web sites. In these, the application services—the mechanisms for publishing, managing, searching, and retrieving content—are accessed through the use of standard protocols and data formats: HTTP and HTML. Client applications (web browsers) that understand these standards can interact with the application services to perform tasks like ordering books, sending greeting cards, or reading news. Because of the abstraction provided by the standards-based interfaces, it does not matter whether the application services are written in Java and the browser written in C++, or the application services deployed on a Unix box while the browser is deployed on Windows. Web services allow for cross-platform interoperability in a way that makes the platform irrelevant.

Interoperability is one of the key benefits gained from implementing web services. Java and Microsoft Windows-based solutions have typically been difficult to integrate, but a web services layer between application and client can greatly remove friction. There is currently an ongoing effort within the Java community to define an exact architecture for implementing web services within the framework of the Java 2 Enterprise Edition specification. Each of the major Java technology providers (Sun, IBM, BEA, etc.) are all working to enable their platforms for web services support.
Many significant application vendors such as IBM and Microsoft have completely embraced web services. IBM for example, is integrating web services support throughout their Web Sphere, Lotus, and DB2 products, And Microsoft's new .NET development platform is built around web services. Web services are a messaging framework. The only requirement placed on a web service is that it must be capable of sending and receiving messages using some combination of standard Internet protocols. The application code holds all the business logic and code for actually doing things (listing books, adding a book to a shopping cart, paying for books, etc.).

The suitability of Web services for integrating heterogeneous systems is largely facilitated through its extensive use of the Extensible Markup Language (XML). The interface of a Web service is for instance described using the XML based Web Services Description Language (WSDL). Furthermore, communication is performed using XML based SOAP messages. Thus, the security of a Web services based system depends not only on the security of the services themselves, but also on the confidentiality and integrity of the XML based SOAP messages used for communication. The Organization for the Advancement of Structured Information
Standards (OASIS) and the World Wide Web Consortium (W3C) have over the last years standardized several specifications related to security in Web services and XML. Recently, Web services are emerging as a systematic and extensible framework for application-to-application interaction, built on top of existing Web protocols and open XML standards. Web services are a new breed of Web applications. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions that can be anything from simple requests for information to creating and executing complicated business processes. Once a Web service is deployed, it can be discovered and invoked by other applications (or other Web services). The key advantage of using Web services is the ability to create applications on the fly through the use of loosely coupled, reusable software components. This has fundamental implications in both technologies and business applications. Business services can be completely decentralized and distributed over the Internet and accessed by a wide variety of communications devices. Businesses can be released from the burden of complex, low and expensive software integration and focus instead on the value of their offerings and mission critical tasks. Then, the Internet will become a global common platform where organizations and individuals communicate with each other to carry out various commercial activities and to provide value-added services. The barriers to providing new offerings and entering new markets will be lowered to enable access for small and medium-sized enterprises. The dynamic enterprises and dynamic value chains become achievable and may be even mandatory for competitive advantages.
The Web services framework is divided into three areas — communication protocols, service descriptions, and service discovery — and specifications are being developed for each. The following specifications are currently most salient and stable in each area:

1. The simple object access protocol (SOAP) that enables communications among Web services. **It** is fundamentally a stateless, one-way message exchange paradigm that enables applications to create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining one-way exchanges with features provided by an underlying protocol and/or application-specific information.
2. The Web Services Description Language (WSDL) that provides a formal, computer-readable description of Web services. **It** provides a model and an XML format for describing Web services. WSDL defines services as collections of network endpoints or ports.
3. The Universal Description, Discovery and Integration (UDDI) directory that is a registry of Web services descriptions. It provides a mechanism for clients to find Web services. Web services are meaningful only if potential users may find information sufficient to permit their execution.

2. LITERATURE SURVEY

The amazon web services provided an overview of the various security processes they have used for providing security to web services [1].

Joe M. Tekli, Ernesto Damiani, Richard Chbeir and Gabriele Gianini gave an overview of current research related to SOAP processing performance enhancement, focusing on similarity-based approaches, as well as WS-Security optimizations, and XML parallel processing architectures. Most methods build on the observation that SOAP message exchange usually involves highly similar messages. They identified the common parts of SOAP messages, to be processed once, only repeating the processing for parts which are different, and substantially reducing SOAP processing overhead [2].

Nils Agne Nordbotten has provided an overview of current security standards for XML and Web services. Together these standards provide a flexible framework for fulfilling basic security requirements such as confidentiality, integrity, and authentication, as well as more complex requirements such as, authorization, and federated identities. Mechanisms such as those provided by Web Services Policy and the Web Services Description Language (WSDL) may also provide valuable sources of information to an attacker trying to find weaknesses in a system. In addition to more common security issues, there are also some attacks/vulnerabilities that are specific to XML. Although XML firewalls may be able to detect messages trying to exploit these vulnerabilities, the use of end-to-end encryption may effectively prevent such detection [3].

Hongbing Wang, Joshua Zhexue Huang, Yuzhong Qu, Junyuan Xie have presented Web services, an emerging technology for the Web. They presented three aspects of Web services: the service security, the service composition, and the service semantics. They are critical to the successful deployment of Web services [4].

Doug Tidwell, James Snell, Pavel Kulchenko has mentioned in their book that a critical insight is that web services do not replace existing technology infrastructures. Rather, they help to integrate existing technologies. In other words, if you need

a J2EE application to talk to another application, web services makes it easier. Web services won't completely replace that 30-year-old mainframe system in the back closet that nobody ever thinks about anymore. But web services might provide cross-platform automated access to the mainframe's applications, thus opening new channels of business [5].

3. SYSTEM OVERVIEW

The Web Services architecture is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions involve the publish, find and bind operations. Together, these roles and operations act upon the Web Services artifacts: the Web service software module and its description. In a typical scenario, a service provider hosts a network-accessible software module (an implementation of a Web service). The service provider defines a service description for the Web service and publishes it to a service requestor or service registry. The service requestor uses a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and invoke or interact with the Web service implementation. Service provider and service requestor roles are logical constructs and a service can exhibit characteristics of both. Figure 3 illustrates these operations, the components providing them and their interactions.

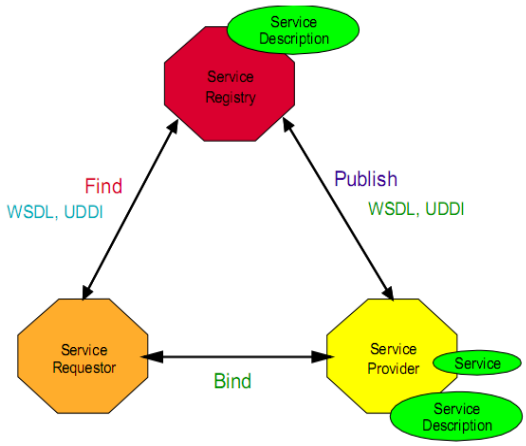

Figure 3: Web Services Architecture

Roles in Web Services Architecture

1. Service provider: From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service.

2. Service requestor: From a business perspective, this is the business that requires certain functions to be satisfied. From an architectural perspective, this is the application that is looking for and invoking or initiating an interaction with a service. The service requestor role can be played by a browser driven by a person or a program without a user interface, for example another Web service.

3. Service registry: This is a searchable registry of service descriptions where service providers publish their service descriptions. Service requestors find services and obtain binding information (in the service descriptions) for services during development for static binding or during execution for dynamic binding. For statically bound service requestors, the service registry is an optional role in architecture, because a service provider can send description directly to service requestors. Likewise, service requestors can obtain a service description from other sources besides a service registry, such as a local file, Web site.

Operations in Web Service Architecture

For an application to take advantage of Web Services, three behaviors must take place:

Publication of service descriptions, lookup or finding of service descriptions, and binding or invoking of services based on the service description. These behaviors can occur singly or iteratively. In detail, these operations are:

1. Publish: To be accessible, a service description needs to be published so that the service requestor can find it. Where it is published can vary depending upon the requirements of the application.

2. Find: In the find operation, the service requestor retrieves a service description directly or queries the service registry for the type of service required.

3. Bind: Eventually, a service needs to be invoked. In the bind operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service.

Artifacts of a Web Service

1. Service: Where a Web service is an interface described by a service description, its implementation is the service. A service is a software module deployed on network accessible platforms provided by the service provider. It exists to be invoked by or to interact with a service requestor. It can also function as a requestor, using other Web Services in its implementation.

2. Service Description: The service description contains the details of the interface and implementation of the service. This includes its data types, operations, binding information and network location. It could also include categorization and other metadata to facilitate discovery and utilization by service requestors. The service description might be published to a service requestor or to a service registry. The Web Services architecture explains how to instantiate the elements and implement the operations in an interoperable manner.

4. WEB SERVICES SECURITY

Web services context, security means that the recipient of a message should be able to verify the integrity of the message and to make sure that it has not been modified. WS-Security from OASIS defines the mechanism to include integrity, confidentiality, and single message authentication features within a SOAP message. WS-Security makes use of the XML Signature and XML Encryption specifications and defines how to include digital signatures, message digests, and encrypted data in a SOAP message. WS-Security is concerned with security for SOAP messages, thus, WS Security clearly builds on top of SOAP. In addition, WS Security also makes use of XML Signature and XML Encryption. The Web Services Security (WSS) specifications aim to provide a framework for building secure Web services using SOAP, and consist of a core specification and several additional profiles. The core specification, the Web Services Security: SOAP Message Security specification, defines a security header for

use within SOAP messages and defines how this security header can be used to provide confidentiality and integrity to SOAP messages. XML Encryption is utilized to provide confidentiality, while message integrity is provided through the use of XML Signature. Using these mechanisms, SOAP message body elements, selected headers, or any combination thereof may be signed or encrypted; potentially using different signatures and encryptions for different SOAP roles that because SOAP message headers may be subject to processing and modification by
SOAP intermediaries, lower layer security mechanisms such as SSL/TLS are often insufficient to ensure end-to-end integrity and confidentiality for SOAP messages. For such messages, the functionality provided by WS-Security is essential if confidentiality and integrity are required.

A major performance bottleneck resides in SOAP message processing. The reason for SOAP performance criticality is twofold: On one hand, SOAP communication produces considerable network traffic, and causes higher latency than competing technologies, like Java RMI and CORBA. On the other hand, and perhaps more importantly, the generation and parsing of SOAP messages and their conversion to and from in memory application data can be computationally very expensive.

Whereas the XML encryption does not provides security in these web services yet, and hence an algorithm can be used to provide security to web services. However, the current Web services architectures are confronted with a few stubborn problems that is, security and the algorithm is used for performing cryptographic operations with symmetric key based security tokens. Existing XML encryption used is symmetric key encryption origin and authenticity of message cannot be guaranteed. Public key encryption allows the use of RSA which enables the recipient of a message to verify that the message is truly from a particular source.The recipient should have received a message confidentially so that unauthorized users could not read it, know the identity of the sender and determine whether or not the center is authorized to carry out the operation requested in the message. These are usually met through encrypting messages. Security is critical to the adoption of Web services by enterprises, but the Web services framework does not meet basic security requirements. The fact that Web services involve exchange of messages means that securing the message exchange is an important issue to consider when building and using Web services. On the other hand, because Web services allow all systems, both internal and external, to communicate on HTTP ports, the application servers are inevitably opened up to “application level” attacks. A few standards have come out to alleviate the message security problem, including WS Security and various other initiatives towards enabling digital signatures on XML messages and transactions.
In general, there are four basic security requirements that the Web Services security layer must provide:

1. Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes, and guarantees that the contents of the message are not disclosed to unauthorized individuals.
2. Authorization is the granting of authority, which includes the granting of access based on access rights and guarantees that the sender is authorized to send a message.
3. Data integrity is the property that data has not been undetectably altered or destroyed in an unauthorized manner or by unauthorized users thereby insuring that the message was not modified accidentally or deliberately in transit.
4. Proof of origin is evidence identifying the originator of a message or data. It asserts that the message was transmitted by a properly identified sender and is not a replay of a previously transmitted message. This requirement implies data integrity.

5. SECURITY ALGORITHMS

Web Service security is big challenge for researchers as it requires a strong security algorithm for the encryption of data. The xml encryption scheme is being used presently for encrypting the messages between the different programming languages running on different platforms, but this xml encryption algorithm is symmetric key encryption algorithm and it creates communication overhead, hence there is need to use an asymmetric key encryption algorithm.

The more powerful version of DES is used for high security called Triple-DES. To start encrypting with Triple-DES, two 56-bit keys are selected. Data is encrypted via DES three times, the first time by the first key, the second time by the second key and the third time by the first key once more. This process creates an encrypted data stream that is unbreakable with today's code-breaking techniques and available computing power, while being compatible with DES. The National Institutes of Standards and Technology considers DES an absolute technology suitable only for legacy applications and today supports a new standard called Advanced Encryption Standard.

AES is a newer encryption standard and is now the preferred one to use for XML Encryption. AES is a substitution-linear transformation network with 10, 12, or 14 rounds, depending on the key sizes, which are currently set at 128, 192, or 256 bits. The block size used in AES is 16 bytes. The data block to be processed is partitioned into an array of bytes forming a matrix with rows and columns. Each cipher operation is byte-oriented.

Symmetric ciphers use the same key for encryption and decryption. That means both sides need to have it, and it needs to be kept secret, because anyone knowing the key can decrypt all messages encrypted with it. The standards DES and AES are examples of symmetric ciphers. Asymmetric ciphers use two keys, a public one for encryption and a private one for decryption. The advantage is that there's no harm in communicating the public key to anyone, because it can't be used to decrypt anything. The private key, on the other hand, doesn't need to be sent to anyone, and is thus easier to keep secret. RSA is an example of an asymmetric cipher. These ciphers are generally much more compute-intensive, so they are rarely used to encrypt large messages.

6. CONCLUSION

In this paper we have presented Web services, an emerging technology for the Web, The web service overview and the various security issues occurred in the implementation of the xml encryption of the messages. The security of web services is an important aspect and hence a security algorithm is required to implement in web services for key generation and encryption decryption of the messages.

The security algorithm described in this paper will be used together in combination for key generation and encryption decryption of the messages which will provide strong security in web services.

7. REFERENCES

[1] 1. Amazon web services: Overview of Security Processes, June 2013,http://aws.amazon.com/security

[2] 2. Joe M. Tekli, Ernesto Damiani, Richard Chbeir and Gabriele Gianini, "SOAP Processing Performance and Enhancement" IEEE Transactions On Services Computing, Vol. 5, No. 3, July-September 2012

[3] Nils Agne Nordbotten, "XML and Web Services Security Standards", IEEE Communications Surveys & Tutorials, Vol. 11, No. 3, Third Quarter 2009.

[4] Hongbing Wang, Joshua Zhexue Huang, Yuzhong Qu, Junyuan Xie, "Web services: Problems and Future Directions", 2005.

[5] "Programming Web Services with SOAP", Doug Tidwell, James Snell, Pavel Kulchenko, First edition, December 2001. .

[6] Web Services conceptual architecture, By Heather Kreger IBM Software Group, 2001Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[7] Locktyukhin, Max; Farrel, Kathy (2010-03-31), "Improving the Performance of the Secure Hash Algorithm (SHA-1)", *Intel Software Knowledge Base* (Intel), retrieved 2010-04-02

[8] http://aws.amazon.com/security.