Hong Kong CyberU
Master of Science in E-Commerce

COMP5091 E-Commerce Dissertation

Securing Web Services

Supervisor : Dr. Vincent Ng
Student : Cham Pui Ying
          (02100020U)
Date : 2 April 2004
# Table of Contents

- Abstract ........................................................................................................... 1
- Introduction ....................................................................................................... 1
- Backgrounds ..................................................................................................... 1  
  - Barriers to Integration .................................................................................. 1  
  - What are Web Services .................................................................................. 1  
  - Advantages of Web Services ......................................................................... 2  
  - Uses of Web Services ................................................................................... 3  
- Web Services Technologies & Standards ............................................................ 3  
  - Web Services Architecture .......................................................................... 3  
  - Web Services Standards .............................................................................. 3  
  - Web Services Model ..................................................................................... 4  
  - Web Services Approach for a SOA Architecture ......................................... 4  
  - The Importance of the Security in Web Services ......................................... 5  
- Securing Web Services ...................................................................................... 6  
  - Security and Web Services .......................................................................... 6  
  - Security Considerations ............................................................................. 7  
  - Web Services Security Schemes ................................................................... 7  
- Identity Management .......................................................................................... 10  
  - Overview of Identity Management ............................................................... 10  
  - What is Federated Identity .......................................................................... 10  
  - Federated Identity Standards ....................................................................... 11  
- Problem Statement ............................................................................................ 11  
- Resources .......................................................................................................... 11
Abstract
This document is the proposal for the e-commerce dissertation with the topic on Securing Web Services. The objective of this document is to describe and give the brief information on the main ideas & concepts of the core Web services’ technologies & Web services’ security, so that the detail study can be progress in further.

Introduction
The IT industry has been talking about Web services for almost four years. Web services allows applications (e.g. automated business transactions, stock trading and order-tracking systems) to communicate with each other within organizations, across enterprises, and across the Internet in a loosely-coupled, platform- and programming language-independent manner. Several key standards have formed the foundation for Web services: XML (Extensible Markup Language), WSDL (Web Services Definition Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery, and Integration).

Since, the key benefit of Web services is to deliver integrated & interoperable solutions, ensuring the integrity, confidentiality & security is the most important key area that needs to be addressed for Web services.

Backgrounds

Barriers of Integration
Traditionally, the barriers of integration are due to the tight-coupling, where one application that calls another one is tied strongly by the function and the parameters. There is low flexibility or adaptability to changing environments or needs, due to:
1. different programming languages
2. different operating systems or hardware platforms
3. different software vendors & in-house coding
4. it’s difficult to integrate these systems internally
5. it’s even harder to integrate with external business partners

Therefore, we need a general standardized solution for integration.

What are Web Services
According to W3C, a Web service is defined as: “A Web service is 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.”

In substance, Web services are technology that allows applications to communicate with each other in a platform-, hardware- and programming language-independent manner. It uses XML based protocols to describe a collection of operations that can be accessed, executed or data exchanged over the network. A group of Web services interacting together in this manner defines a particular Web service application in a Service-Oriented Architecture (SOA).

Web services exhibit the following definitive characteristics:

- Web services communicate using platform-, hardware-independent and language-neutral Web protocols. These Web protocols ensure easy integration over the network & loosely coupling between applications.
- A Web service provides an interface that can be called from another program. This application-to-application programming interface can be invoked from any type of application client or service.
- A Web service is registered and can be located through a Web Service Registry. The registry enables service consumers to find services that match their needs.

**Advantages of Web Services**

- **Flexibility** – Web services allow loose-coupling, which means that interactions between service applications may not break even there is a change. These universal interfaces can cope with inevitable changes in software caused by changing business needs.
- **Agility and Productivity** – Rapid application assembly tools allow integration for new business opportunities or trying new business ideas.
- **Cost Savings** – It allows automatic transactions, replace manual methods, reduce staffing requirements, replace paper processing & reduce errors.
- **Leverage Existing Investments** – Web services provide existing or legacy software applications with service interfaces without changing the original applications, allowing them to fully operate in the service environment. Old software can be used in new ways by building Web services layer for universal access. This adapts existing applications to changing business conditions and customer needs.
- **Leverage Developer Skillsets** – The plumbing code is generated automatically and can be integrated & tested with traditional methods.
- **Same infrastructure for any M2M integration** – EAI, B2B, P2P, handhelds, browsers,
grid computing, technologies yet to be invented. Interact between services on any platform, written in any language.

Uses of Web Services
What can I do with Web services? While Web services provide all the advantages stated above, Web services allow us to implement as:

- A credit card service that processes credit card transactions for a given account number.
- A market data service that provides stock market data associated with a specified stock symbol
- An airline service that provides flight schedule, availability, and reservation functionalities.

Web Services Technologies and Standards
Web Services Architecture
Conceptually, Web services stack can be defined as the figures below.

There are nine layers:
1. Transport
2. Service Communication Protocol
3. Service Description
4. Service
5. Business Process
6. Service Registry
7. Policy - Security
8. Transaction
9. Management

Web Services Standards
Web services are widely adopted standards such as HTTP and eXtensible Markup Language (XML). Typically, these standards are maintained by independent, non-profit standards organizations.

A few of the major Web services standards groups are listed below:
- W3C (World Wide Web Consortium) - The driving force behind the largest number of
highly adopted standards in the Web services space including some Web building blocks such as HTML.

- OASIS – Source of the original specification from which XML evolved, as well as the home of the current XML and Universal Description, Discovery and Integration (UDDI) specification.
- WS-I (Web Services Interoperability Organization) – Acts as a watchdog group to ensure interoperability between implementations of Web services standards.

### Web Services Model

A typical Web services model consists of three entities:

- Service providers who create Web services and publish them to the outside world by registering the services with service brokers.
- Service brokers who maintain a registry of published services.
- Service requesters who find required services by searching the service broker’s registry. Requesters then bind their applications to the service provider to use particular services.

### Web Services Approach for a SOA Architecture

Web services allow applications interact with one another over the Web, so there is necessary for them to find one another, discover the information and patterns to interconnect. Therefore, Web services involve a family of related protocols to describe, deliver, and interact with services. And Web services require several related XML-based technologies to transport and to transform data into and out of programs and databases.

Web services are essentially founded upon four major technologies:

- XML (eXtensible Markup Language) is the markup language that underlies most of the specifications used for Web services. XML is a generic language that can be used to describe any kind of content in a structured way, separated from its presentation to a specific device.
- WSDL (Web Services Description Language) – WSDL is a series of XML statements that constitute the definition for the interfaces of each service.
- UDDI (Universal Description, Discovery and Integration) – UDDI lets Web services register their characteristics with a registry so that
other applications can look them up.

- **SOAP (Simple Object Access Protocol)** – SOAP provides the means for communication between Web services and client applications. It handles the issues of messaging, interface description, addressing and delivery.

### The Importance of the Security in Web Services

In February / March 2003 CBDA Forum carried out a survey of its subscribers who had practical experience in implementing Web services to understand how they were applying Web services, their motivation for adoption, and their experience to date as well as their detailed as well as their further plans for 2003.

The responses came from a number of industry sectors, Systems Integrators 27%, Government 10%, Telecoms 13%, Finance 26%, Manufacturing / Process 10%, Travel / Transport 7%, Retail / Logistics 7%.

From the survey result, we can understand the motivation for adopting Web services and the reasons for using Web services.

![Motivation for Adopting Web Services](image)

![Reasons for Using Web Services](image)
However, we also find that the security is the highest barrier to wider adopting Web services.

Therefore, Security is a key critical success factor for Web services.

**Securing Web Services**

Though there is nothing can ever be proven to 100% secure, we should make enough security to make Web services practical.

**Security and Web Services**

Security is important for any distributed computing environment. But, security is even more important for Web services due to the following reasons:

1. The boundary of interaction between communicating partners is expected to expand from intranets to the Internet. Obviously, security problem is much critical in Internet because Internet communication is much less protected than intranet communication.

2. There will have more anonymous to access the web services since communicating partners are more likely to interact with each other without establishing a business or human relationship first. This means that all security requirements such as authentication, access control, non-repudiation, data integrity, and privacy must be addressed by the underlying security technology.

3. More and more interactions are expected to occur from programs to programs rather than from humans to programs. Therefore, the interaction between communicating partners using Web services is anticipated to be more dynamic and instantaneous.

4. As more and more business functions are exposed as Web services, the number of participants in a Web services environment will be larger than what we have seen in
other environments.

Security Considerations
Security is about protecting assets. In the Web services context data and computational services are assets under consideration. The following security considerations must be considered as part of a comprehensive security framework:

- Identification – The party accessing the resource is able to identify itself to the system.
- Authentication – the proven identification of users in a computer system.
- Authorization – There exists a set of transactions the authenticated party is allowed to perform.
- Integrity – the prevention of unauthorized modification of data.
- Confidentiality – the prevention of unauthorized disclosure of data.
- Accountability – the provision of activity logs recording all user activity.
- Non-repudiation – Both parties are able to provide legal proof to a third party that the sender did send the information, and the receiver received the identical information.

Web Services Security Schemes
Web services security language can be defined into two types: computer security and communications security.

- Computer security is a node-oriented security focus and it is essentially access control within a computer system. A permission rule expresses restrictions on the usage at the server side and a client can execute the operations only if the permission rule is allowed.
- Communications security is a connection-oriented security focus and it is about providing a secure logical connection between two agents. A requirements rule expresses the necessary security-relevant preparations for the use of a service, or security measures needed after the service execution. The activities authenticates and encrypts are associated with authentication and confidentiality, respectively.

And currently, the most common security scheme available for today’s Web service is SSL (Secure Socket Layer), which is typically used with HTTP. It provides authentication, confidentiality, and message integrity. However, SSL is designed to provide point-to-point security, which falls short for Web services because:

1. We need end-to-end security, where multiple intermediary nodes could exist between the two endpoints.
2. SSL secures communication at transport level rather than at message level. As a result, messages are protected only while in transit on the wire.
3. HTTPS in its current form does not support non-repudiation well. Non-repudiation is critical for business Web services.
4. SSL does not provide element-wise signing and encryption.

In order to complement SSL, the technology industry has been working on various XML-based security schemes to provide comprehensive and unified security schemes for Web services. These schemes include:

- **XML digital signature** – XML digital signature provides authentication, data integrity and non-repudiation. It is to develop XML syntax for representing digital signatures over any data type. The XML digital signature specification also defines procedures for computing and verifying such signatures. Another important area that XML digital signature addresses is the canonicalization of XML documents. Canonicalization enables the generation of the identical message digest and thus identical digital signatures for XML documents that are syntactically equivalent but different in appearance. XML digital signature provides a flexible means of signing and supports diverse sets of Internet transaction models.

- **XML encryption** – Its goal is to develop XML syntax for representing encrypted data and to establish procedures for encrypting and decrypting such data. (Unlike SSL, with XML encryption, you can encrypt only the data that needs to be encrypted.)

- **XKMS (XML Key Management Specification)** – XKMS consists of two parts: XKISS (XML Key Information Service Specification) and XKRSS (XML Key Registration Service Specification). XKISS defines a protocol for resolving or validating public keys contained in signed and encrypted XML documents, while XKRSS defines a protocol for public key registration, revocation, and recovery. The key aspect of XKMS is that it serves as a protocol specification between and XKMS client and an XKMS server in which the XKMS server provides trust services to its clients by performing various PKI operations.

- **XACML (Extensible Access Control markup Language)** – Its goal is to standardize access control language in XML syntax.

- **SAML (Secure Assertion Markup Language)** – It’s to outline a standard XML framework for exchanging authentication and authorization information. As a framework, it deals with three things. First, it defines syntax and semantics of XML-encoded assertion messages. Second, it defines request and response protocols between requesting and asserting parties for exchanging security information. Third, it defines rules for using assertions with standard transport and message frameworks.

- **WS-Security (Web Services Security)** – It defines a set of SOAP header extensions for end-to-end SOAP messaging security. It supports message integrity and confidentiality by allowing communicating partners to exchange signed and encrypted messages in a
Web services environment.

- **ebXML Message Service** – The ebXML initiative is a set of next-generation XML-based standards enabling electronic business transactions via the Internet. One of the ebXML standards is ebXML Message Service, which defines how to securely and reliably send and receive SOAP messages.

The SAML assertions can be digitally signed using XML digital signature. The same assertions can be encrypted using XML Encryption to ensure privacy. The public key used for digital signing and encryption can be validated and registered via XKMS. As for XACML, an SAML asserting party could use it to define an access control policy as a basis for handling SAML-based assertion requests.

Take an example: When a client placing an order, she uses XML digital signature and encryption to digitally sign and encrypt the purchase order XML document. She then sends the document to her supplier using SOAP, whose header structure is defined either in the WS-Security or ebXML Message Service standard. The document's receiver then could use XKMS to look up and validate the public key. Once the key is determined trustworthy, the receiver then validates and decrypts the purchase order. Finally, the receiver checks a policy server for authorization by sending and receiving SAML requests and responses. The policy server might maintain the access control policy information in XACML.

**Identity Management**

**Overview of Identity Management**

Identity is a set of attributes that describes a profile of an individual, business organization, or software entity. E-business initiatives – such as enterprise, B2B and B2C applications – typically reach throughout and beyond and enterprise, requiring users to move across networks, applications, and security domains. If there is lack of a well-integrated and interoperable identity management architecture, this makes managing Web properties, applications, identities, and policies non-scalable, and effectively prohibits the interaction of identities across applications or Web services. To be effective, this movement must be transparent to the user. Consider what’s involved in this: a single identity with one registration process and one login procedure.

There are two identity management architectures, centralized model and federated model.

- Centralized model – In the centralized model, a single operator performs authentication and authorization by owning and controlling all the identity information. It makes the constructing and managing the identity network much easier. However, there is the
dangerous potential for the single operator becoming a tollgate for all transactions over the Internet, the single operator could represent a single point of security failure or hacker attack and a single operator can take away the most important business asset (customer identity and profile information).

- **Federated model** – In the federated model, both authentication and authorization tasks are distributed among federated communities. It’s to create an open standard for identity, authentication and authorization, which will lower e-commerce cost and accelerate organizations’ commercial opportunities, while at the same time increasing customer satisfaction. Furthermore, organizations can maintain their own customer data while sharing identity data with partners based on their business objectives and customer preferences.

As centralized model cannot effectively manage or control an e-business initiative from beginning to end, especially when multiple partners are involved. That’s why organizations are turning to federated identity management. The appeal of federation is that they are intended to allow user to seamlessly traverse different sites within a given federation. Federations provide a simple and flexible mechanism to identify and validate users from partner organizations and provide them with seamless access to Web sites within that trusted federation without requiring re-authentication. In addition, Federation standards also deal with the matter of providing trusted attributes about users allowing for privacy and business-specific rules.

Perhaps nothing is more important to the future of Web services than federated identity, the ability to securely establish a person's or a service's identity and to share that identity across domains and enterprises. Establishing a unique identity is the key component in being able to take advantage of services and applications beyond a domain or firewall, which is the ultimate promise of Web services.

**What is federated identity**

Federated identity is a way to establish someone's identity across companies, domains and applications. The idea is that once that identity is established in one place, it can be carried across to other Web services. So complex transactions and applications can be used, without the person having to log into separate applications or services and information about that person can be carried across as well.
Federated Identity Standards
A number of different standards apply to federated identity, but there are three primary ones:

- **SAML** - This standard concerns itself with authentication and authorization. The current version is 1.1, but a major new version, 2.0, is due out this summer, and integrates more closely with the Liberty Alliance federated identity standards.
- **WS-Federation (Web Services Federation Language)** - This is an attempt to build an overriding federated identity standard, to work in concert with SAML and other security standards. Prime movers behind it are BEA, IBM, Microsoft, RSA Security and VeriSign.
- **Liberty Alliance** - This is a set of standards for federated identity overseen by a group of companies called the Liberty Alliance, of which Sun was a prime mover and founder.

Problem Statement
As more and more business organizations adopt Web services, ensuring secure communication between communicating partners is becoming even more important. For Web services environments, security is becoming even more important due to the Web services’ unique characteristics. In the dissertation, I will discuss the Web services’ characteristics, technologies & standards, however the focus will be on the Web services’ security.

In the dissertation, there will be two main parts, technologies study and implementation. Under the technologies study, I will introduce the main ideas & concepts of the core Web services’ technologies, explain why SSL falls short when it comes to Web services and XML-based Web services security schemes. And I will also discuss the importance of the federated identity. Then I will implement the Web services security under the technologies of WebSphere and J2EE by deploying a Web service on a case of e-marketplace so to show how these technologies might be used together.

Resources
Reference Papers
University.

Reference Books

Web Resources
iv. Web Services Architecture - http://www.w3.org/TR/ws-arch/#whatis

Workshops