The Hong Kong Polytechnic University

COMP5009 – Independent Study in e-Commerce

MSc. in e-Commerce for Executives

Department of Computing

An Independent Study

on

Key Industry Standards for Securing Web Services

Supervisor's Name: Dr. Vincent NG

Student’s Name: KWAN Shuk Han, Barbara (02727269g)

December 2003 (Sem 1, 2003/04)
# Table of contents

*Introduction* ........................................................................................................................................... 1

1  Purpose of report .................................................................................................................................. 1

**Part I – An Overview on Web Services** ................................................................................................. 2

2  What are Web Services? ....................................................................................................................... 2
   2.1  No Official Consensus on Definition ......................................................................................... 2
   2.2  Definition Adopted in this Report ............................................................................................. 3
   2.3  Uses of Web Services ................................................................................................................ 4

3  Architectural Layers of the Web Services Computing Paradigm ......................................................... 5
   3.1  Network Communications Protocols Layer .............................................................................. 6
   3.2  XML Technologies Layer .......................................................................................................... 6
   3.3  Web Services Technologies Layer ............................................................................................ 7
   3.4  Web Services Infrastructure Layer ........................................................................................... 7
   3.5  Services Layer ............................................................................................................................ 8
   3.6  Web Services-based Application Templates Layer .................................................................... 8

4  Security: A Critical Success Factor for Web Services ......................................................................... 8

**Part II – Key Industry Standards for Securing Web Services** ................................................................ 10

5  Principles of Information Security ...................................................................................................... 10

6  New Security Challenges & Threats in Web Services ....................................................................... 11

7  Key Industry Standards / Technologies for Secure Web Services ..................................................... 11
   7.1  Overview .................................................................................................................................... 11
   7.2  Standards for XML Content Confidentiality, Integrity and Authenticity .................................. 14
       7.2.1  XML Encryption ............................................................................................................... 14
       7.2.2  XML Signature .................................................................................................................. 14
   7.3  Standards for Trust Management .............................................................................................. 15
       7.3.1  XKMS (XML Key Management Specification) .................................................................. 15
       7.3.2  SAML (Security Assertion Markup Language) ............................................................... 16
       7.3.3  XACML (Extensible Access Control Markup Language) ............................................. 17
   7.4  Standard for Message Structure ............................................................................................... 18
       7.4.1  WS-Security (Web Services Security) ............................................................................. 18

8  High-level Architecture for Secure Enterprise Web Services ............................................................ 19
   8.1  Logical Architecture .................................................................................................................... 19
   8.2  Guidelines for deploying networks of Web Services in an enterprise .................................... 21

**Part III – Comparison of Two “Identity Federation” Specifications** ..................................................... 22

9  Introduction ......................................................................................................................................... 22
   9.1  Federated Identity Management ............................................................................................... 22

10 Overview of Identity Federation Framework (ID-FF) ........................................................................... 23
   10.1 Overview ................................................................................................................................... 24
   10.2 Features ..................................................................................................................................... 24
   10.3 Protocols .................................................................................................................................. 25

11 Overview of WS-Federation ................................................................................................................. 26
   11.1 Overview ................................................................................................................................... 26

12 Comparison: ID-FF vs. WS-Federation .............................................................................................. 27
   12.1 Scope of Comparison ................................................................................................................ 27
   12.2 Comparison on Non-functional factors ................................................................................... 27
       12.2.1 Goal .................................................................................................................................... 27
       12.2.2 Role assumed within the overall Specification Architecture ........................................... 28
12.2.3 Maturity ......................................................................................................................... 28
12.2.4 Standardization Approach and Current Status ............................................................... 28
12.2.5 Implementation Costs .................................................................................................... 29
12.2.6 Industry Adoption ........................................................................................................ 29
12.2.7 Scope of contents ........................................................................................................... 30
12.2.8 Underlying Technology ................................................................................................. 30
12.3 Comparison on Functional Factors .................................................................................. 31
12.4 Discussion ............................................................................................................................. 31

**Conclusion** ............................................................................................................................. 33

13 Conclusion ................................................................................................................................. 33
14 Future Work ............................................................................................................................... 33

**Appendix A: Listings of Sample XML Codes** ........................................................................ 34

15 Listing 1: Simple XML Encryption ........................................................................................... 34
16 Listing 2: Embedded XML Signature ........................................................................................ 35
17 Listing 3: X-KRSS Registration Request for a Key Pair ............................................................ 36
18 Listing 4: Simple SAML Authentication Assertion ................................................................. 38
19 Listing 5: A sample XACML Policy ......................................................................................... 39
20 Listing 6: WS-Security Header ............................................................................................... 40
21 Listing 7: SAML Assertion in a WS-Security Header ............................................................... 40
22 Listing 8: Referring SAML information in WS-Security .......................................................... 40

**Glossary** ..................................................................................................................................... 41

**References** ............................................................................................................................... 42

23 Cited References ..................................................................................................................... 42
24 Web resources referenced ........................................................................................................ 43
24.1 W3C standards ..................................................................................................................... 43
24.2 OASIS Standards .................................................................................................................. 43
24.3 General references ............................................................................................................... 44
25 Books referenced ..................................................................................................................... 45
Introduction

1 Purpose of report

The purpose of this report is to document the findings of an independent study carried out by the author in accordance to the requirements of the COMP5009 “Independent Study on E-Commerce” course.

The topic of interest selected for this independent study is “Key Industry Standards for Securing Web Services”.

A literature view on the topic has been carried out. The findings are documented in Part I (An Overview of Web Services) and Part II (Key Industry Standards for Securing Web Services) of this report.

In addition, a comparison has been done to compare and contrast two different sets of specifications, which are developed for addressing the identity federation requirements in the Web services world. The results of this comparison are documented in Part III (Comparison of Two “Identity Federation” Specifications) of this report.
Part I – An Overview on Web Services

The Web services computing paradigm has caught significant attention from the computer industry. Web services are seen as a promising solution to an age-old need of fast and flexible information sharing among people and businesses. However, along with the potential benefits of Web Services there come new sources of security requirements and challenges that need to be properly addressed and managed.

Standards bodies and industry players are serious in developing Web services related standards and specifications in the hope that this new computing paradigm will continue to evolve, and mature and reach its full potential.

However, there is no official consensus within the computer industry on the definition of Web services. For the purpose of this report, I have adopted the W3C’s definition of Web services as the base reference for common understanding.

In this part, an introductory overview on Web services is given in order to establish a common ground on some Web services fundamental concepts before we look into the key industry standards on Web services security in Part II.

The topics covered here are:

- What are Web Services?
- Architectural Layers of the Web Services Computing Paradigm
- Security: A Critical Success Factor for Web Services

2 What are Web Services?

2.1 No Official Consensus on Definition

Despite of the immerse interest generated from the computer industry in Web services, there exists no official consensus within the industry on its definition. Manes [1, p 27] has made the following observation:

“Some people use the term “Web Services” to describe applications that communicate with Simple Object Access Protocol (SOAP). Other folks use the term to describe only the SOAP interface. Still other people vehemently object to the idea of constraining the definition to a specific technology such as SOAP. Some people use the term to describe any application that communicates over the Internet. Other people use the term to describe any Web-based application. Some people view Web Services as anything that’s
accessible over the Web. And some people use the term to describe the **software-as-a-service** business model.”

Similar observation is also revealed in a Web services survey conducted by the Cutter Consortium in July 2003 [2]. Among the 240 organizations participated in that survey, there are different views on what Web services are. The distribution of these views are summarized as follows:

<table>
<thead>
<tr>
<th>View</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web services, in essence, XML and associated middleware</td>
<td>12%</td>
</tr>
<tr>
<td>Web services, in essence, components one accesses online</td>
<td>22%</td>
</tr>
<tr>
<td>Web services are components one accesses online as well as XML and associated middleware</td>
<td>59%</td>
</tr>
<tr>
<td>Other</td>
<td>7%</td>
</tr>
</tbody>
</table>

Evidently, Web services continue to elude concise definition [2].

### 2.2 Definition Adopted in this Report

Without prejudice toward other definitions, the W3C’s definition for Web services is adopted in this report for the following reasons:

- a definition proposed / recommended by a standards body provides a stronger guarantee on authority, precision, currency and vendor-neutrality;
- among the standards bodies that are involved with Web services standards and specifications, only the W3C has got an official definition on Web service.

According to W3C [3, Section 2], a Web service is defined as follows:

“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.”.

W3C offers further elaboration on Web services in its “Web Services Architecture” document [4, Section 1.5.1]:

“A Web service is viewed as an abstract notion that must be implemented by a concrete agent. The **agent** is the concrete entity (a piece of software) that sends and receives messages, while the **service** is the abstract set of functionality that is provided. To illustrate this distinction, you might implement a particular Web service using one agent...
one day (perhaps written in one programming language), and a different agent the next day (perhaps written in a different programming language). Although the agent may have changed, the Web service remains the same.”

In essence, Web services are loosely-coupled, distributed environments that allow the service providers to expose business functions to their target service requesters (e.g. customers or partners) over the Internet. The beauty of this environment is that a provider and a requester do not have to worry about the operating system, language environment, or component model used to create or access the software programs that implement a particular service.

### 2.3 Uses of Web Services

Web services can be used for the following purposes:

1. **Alternative to Web applications**
   The user’s interaction with a Web application returning HTML documents is replaced by a Web service consumer (or client) interaction with a Web service provider returning XML documents.

2. **Front-end (or proxy) to legacy applications**
   Users access a Web service that interfaces with a back-end application or database instead of having to access the back-end application itself. In this way, the requester does not need to have a specific client to access the back-end application, making the environment more flexible and scalable.

3. **Linchpin of service-oriented architectures (SOAs)**
   Until recently, SOAs were based on object model based protocols such as Microsoft DCOM or CORBA/IIOP. Although these architectures rely on industry-standards, their implementations rely on binary protocols that don’t work well with firewalls. In addition, when you change an object, you need to change the client that accesses that object, making the environment tightly integrated, as opposed to the loosely-coupled approach of Web services.

4. **Standards-based alternative to proprietary business-to-business interactions and EAI solutions**
   Interestingly, most EAI vendors are embracing Web services and are providing more loosely-coupled integration solutions based on their proprietary platform.
3 Architectural Layers of the Web Services Computing Paradigm

From a high-level view, the Web services computing paradigm can be described in terms of six architectural layers, as illustrated in the figure below.

![Figure 1: Architectural layers of the Web services computing paradigm](image)

Legend:
- W3C specification or standard
- OASIS specification or standard
- Joint effort of IETF & W3C
The six architectural layers are (from the lowest to the top layer):

1. Network Communications Protocols;
2. XML technologies,
3. Web services technologies,
4. Web services infrastructure,
5. Services,
6. Web service-based application templates.

The constituents of each layer are briefly explained in Sections 3.1 to 3.6 below.

### 3.1 Network Communications Protocols Layer

Services are invoked and provide results via messages that must be exchanged over some communications medium. This layer denotes the communications component of the Web services paradigm. The W3C’s Web Services Architecture [4] encompasses a wide, almost infinite variety of communications mechanisms: HTTP (the dominant protocol of "the Web"), other Internet protocols such as SMTP and FTP, generic interface APIs such as JMS, earlier distributed object protocols such as IIOP, and so on.

Since this report focuses on key industry standards on Web services security, we will not go into the details of the communications protocols here.

### 3.2 XML Technologies Layer

This layer represents the XML technologies that provide the foundation for the development of many of the Web services technologies such as SOAP, WSDL, and many others.

These XML technologies include [4, Section 1.7, 3.15.1]:

- XML core syntax: XML 1.x, XML Base specification,
- the concepts of the XML Infoset,
- XML Schema Description Language,
- XML Namespaces

For the purpose of this report, two XML-based security technologies are also highlighted in Figure 1. They are “XML Encryption” and “XML Signature”. Descriptions on these technologies are found in Section 7.2 in Part II.

This layer is crucial to the overall picture of Web services.
3.3  Web Services Technologies Layer

The layer represents the core and advanced Web services technologies or specifications.

The core technologies are those that provide the most basic functionality. They include SOAP, WSDL, and UDDI. SOAP & WSDL are W3C standards while UDDI is an OASIS standard. Details on these technologies can be found from the W3C (www.w3.org) and OASIS (www.oasis-open.org) web sites respectively. A brief description on the history of SOAP, WSDL, and UDDI and their respective paths to becoming an industry standard can be found in [1, p. 83-97].

The advanced technologies are those that provide advanced middleware functionality such as security, management, transactions, choreography, and so on, in a Web services environment. For the purpose of this report, Figure 1 highlights only the major security-related specifications. They are:

- WS-Security (Web Services Security) Specification,
- SAML (Security Assertions Markup Language),
- XACML (Extensible Access Control Markup Language), and
- XKMS (XML Key Management Specification).

Descriptions of these standards / technologies are provided Section 7.3 and 7.4 in Part II.

3.4  Web Services Infrastructure Layer

This layer represents the software products that implement the Web services technologies for people to build, deploy, manage and use Web services. These products can be grouped into the categories of “core” products and “associated” products.

The core products provide the basic infrastructure that supports Web services. They can be sub-categorised into Web services platforms, Web services management extensions, and infrastructure-level Web services. A Web services platform consists of a SOAP runtime environment and the tools needed for the development, deployment and management of Web services. The Web services management extensions add advanced features, such as security, version control, and monitoring, to the Web services platform. The infrastructure-level services are Web services that implement parts of the infrastructure, such as a reliable network provider (RNP) service, a UDDI service, and a single sign-on (SSO) service.

The associated products are products that use or rely on Web services. Examples of this category of products include:
• XML tools
• Application adapters for Web services
• Message switching systems
• Web services testing, diagnostic, and optimization tools
• etc.

3.5 Services Layer

Logically, this layer represents the space where the “services” notion of the Web services paradigm exists. Physically, it represents those software programs that actually implement the “services”. The sample services listed in Figure 1 are just some of the oft-quoted examples (i.e. sales quote, order tracking, weather reports, stock trading, and map and directions) and are by no means exhaustive.

As a side remark, there are Web service broker sites, such as www.xmethods.net and www.salcentral.com, available for listing Web services to the public.

3.6 Web Services-based Application Templates Layer

This is the top layer of the Web services computing paradigm. The Web services-based application templates represent the kinds of applications and initiatives for which Web services technology offers the most significant benefits. Again, the examples listed in Figure 1 are by no means exhaustive. They just represent some of the more popular application templates, such as portals, enterprises resource planning, customer relationship management, enterprise application integration initiatives, and business-to-business integration.

4 Security: A Critical Success Factor for Web Services

Corporations are discovering the power of Web services-enabled e-business applications to increase customer loyalty, support sales efforts, and manage internal information. The common thread in these diverse efforts is the need to present end users with a unified view of information stored in multiple systems. To satisfy this need, there are situations where legacy systems are to be integrated with new Web services-based applications that provide broad connectivity across a multitude of back-end systems. These new Web services-based applications can bring direct bottom-line benefits. However, they can open a direct pipeline to the corporation’s most valuable information assets, presenting a tempting target for fraud, and malicious hackers. The pervasive reach and platform-agnostic nature of Web services demands a security framework that enables enterprises to secure and control access to applications and data, without impeding the exchange of data that is essential for successful Web services.
In essence, before Web services will be successful, security must be in place. Companies will never be willing to open up their internal corporate networks without the proper countermeasures. Security is a key critical success factor for Web services.
Part II – Key Industry Standards for Securing Web Services

In this part, the following aspects regarding web services security are covered:

- Principles of Information Security;
- New Security Challenges & Threats in Web Services;
- Key Industry Standards / Technologies for Secure Web Services;

5 Principles of Information Security

Information security focuses on protecting valuable and sensitive enterprise data. To secure information assets, organizations must provide availability to legitimate users, while barring unauthorized access.

According to the recommendation of the National Institute of Standards and Technology (NIST) [5, p. 2-3], information technology security is characterized by the following security objectives:

1. Availability (of systems and data for intended use only)
   - To assure that systems work promptly and service is not denied to authorized users;
   - To protect against intentional or accidental attempts to perform unauthorized deletion of data or to cause a denial of service or data; or to use system or data for unauthorized purposes.

2. Integrity (of system and data)
   - System Integrity – to ensure that a system performs its intended function in an unimpaired manner, free from unauthorized manipulation;
   - Data Integrity – to ensure that data has not been altered in an unauthorized manner while in storage, during processing, or while in transit.

3. Confidentiality (of data and system information)
   - To ensure that private or confidential information is not disclosed to unauthorized individuals. Confidentiality protection applies to data in storage, during processing, and while in transit.

4. Accountability (to the individual level)
• It is the requirement that actions of an entity may be traced uniquely to that entity.
• Accountability is often an organization policy requirement and directly supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.

5. Assurance (that the other four objectives have been adequately met)
• Assurance is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes.

This same set of information security principles also applies to Web services.

6 New Security Challenges & Threats in Web Services

The breadth of information security in Web services applications is broad. It goes beyond the low-level topics of security for networks, firewalls, and operating systems; or of cryptography.

Web services change the risk levels associated with deploying software because of the increased ability to access data. They move transactions beyond firewalls and enable outside entities to invoke applications, potentially giving outsiders access to sensitive information. As a result, Web services present new security challenges. Although existing security standards protect data as it travels over the Internet, Web services require additional measures to secure data.

7 Key Industry Standards / Technologies for Secure Web Services

7.1 Overview

By and large, companies deploying Web services rely on traditional transport-level security for authentication, and application-specific security for authorization and accounting or audit. However, this setting is not secure enough. In the Web services paradigm, companies need to be able to validate the content of messages requesting a service before these messages reach the specific service, and need to be able to keep track of who is trying to access the service. This is elaborated in the paragraphs below.

Transport-level Security alone is not enough

Secure Socket Layer (SSL) is the most widely used transport-level data-communication protocol. It provides authentication (the communication is established between two trusted parties), confidentiality (the data exchanged is encrypted), and message integrity (the data is checked for
possible corruption). However, SSL supports transport-level security **between two SSL-enabled parties only**. This means that when the data is not “in transit” on the secure communication channel, it is not encrypted, and therefore it is not secure. When the documents involved in the transaction are between two SSL sessions, they are vulnerable to attacks. This is the reason why transport level security is not enough, particularly in multi-step Web service transactions.

**XML-based application-level security can complement Transport-level security**

Transport-level SSL can be complemented with XML-based, application-level security, including technologies that support message structure, XML content confidentiality, integrity, authenticity, and fine-grained XML content access control.

**Web Services Security Stack**

A more robust Web services security stack has emerged and is depicted below:

![Web Services Security Stack](image)

**Table 1 : XML-based security standards / technologies**

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>NAME OF THE STANDARD</th>
<th>BRIEF DESCRIPTION OF THE STANDARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Structure</td>
<td>1. WS-Security</td>
<td>Define security extensions to the SOAP protocol.</td>
</tr>
<tr>
<td>XML Content Confidentiality, Integrity, and Authenticity</td>
<td>XML Encryption</td>
<td>Represent the encrypted content of XML data, the information that enables a recipient to decrypt it, and a mechanism for conveying encryption-key information to the recipient.</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3. XML Signature</td>
<td>Define the representation of signatures on digital content, and procedures for processing those signatures. XML Signature provides detailed elements supporting data integrity, signature assurance, and non-repudiation for Web services data.</td>
<td></td>
</tr>
<tr>
<td>Status: a W3C recommendation (12 Feb 2002).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trust Management</td>
<td>4. XKMS (XML Key Management Specification)</td>
<td>Define the registration and distribution of encryption keys for Public Key Infrastructure (PKI) in Web services. XKMS improves PKI by establishing a platform-independent set of standards and by simplifying the steps necessary to implement PKI.</td>
</tr>
<tr>
<td>Status:</td>
<td>• XKMS V2.0 – a W3C Candidate Recommendation (27 Aug 2003).</td>
<td></td>
</tr>
<tr>
<td>5. SAML (Security Assertion Markup Language)</td>
<td>Describe authentication, attributes, and authorization decision objects (or assertions) that can be exchanged between trusted partners.</td>
<td></td>
</tr>
<tr>
<td>Status:</td>
<td>• SAML V1.1 – an OASIS Standard (2 Sep 2003).</td>
<td></td>
</tr>
<tr>
<td>• SAML V2.0 – still in OASIS standardization process.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. XACML (Extensible Access Control Markup Language)</td>
<td>Specify policies to access XML documents based on objects (elements to be accessed in the XML document), subject (a user or a service), and action (read, write, create, delete). XACML may use SAML for authentication. XACML can be used by policy management systems to exchange policies in a standard way.</td>
<td></td>
</tr>
<tr>
<td>Status:</td>
<td>• XACML V1.0 – an OASIS Standard (6 Feb 2003).</td>
<td></td>
</tr>
<tr>
<td>• XACML V1.1 – currently an OASIS Committee Draft.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• XACML V2.0 – currently an OASIS working draft.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These standards / technologies are explained in more detail in Section 7.2 to 7.4 below. Web resources from the W3C and the OASIS web sites were referenced for information on these standards (See Section 24.1 and 24.2 on page 43).
7.2 Standards for XML Content Confidentiality, Integrity and Authenticity

7.2.1 XML Encryption

The XML Encryption specification describes how to represent in XML a digitally encrypted Web resource. A Web resource can be an XML document or non-XML data, such as picture or audio files. Encryption of a document can be partial. For example, we can encrypt only a credit card number as shown in Listing 1: Simple XML Encryption on page 34.

The XML Encryption specification defines:
- How digital content is encrypted and decrypted.
- How the encryption key information is passed to a recipient.
- How encrypted XML data and non-XML data is identified in order to facilitate the decryption process by the recipient.

The XML encryption specification describes how to use XML signature with XML encryption so that trusted parties can selectively encrypt and sign portions of documents, whether those documents are already encrypted and signed or not.

The main element in the XML encryption schema is `<EncryptedData>`. Using the `<EncryptedData>` element, one can define which data in the XML document needs to be encrypted, as well as the encryption value. See Listing 1: Simple XML Encryption on page 34 for an example. In this example, the value of the `<CreditCardNumber>` element is encrypted.

7.2.2 XML Signature

The purpose of an XML signature is to associate a private key with referenced data. The key is used to ensure the origin of the signed data. In other words, an XML signature guarantees the sender’s authentication, thus assuring the recipient that the document is really coming from a trusted originator.

XML signatures apply to XML and non-XML documents. In the case of an XML document, XML signatures can apply to either a part of or the entire document. XML signatures can be used with encrypted and non-encrypted documents.

Enveloped XML signatures are embedded in the same document they apply to. Detached XML signatures apply to data that is external to the signature itself.
The XML Signature specification defines the signature syntax and data structures, as well as the signature processing rules (signature generation and validation). An XML signature is defined within the <Signature> element, which can be an XML document in its own right in the case of a detached signature.

Both XML signature and XML encryption may be applied to an XML document in random order. For example, a party may encrypt portions of a document after the document is signed. In this case, the signature is not verifiable. The portions of the document encrypted after signing have to be decrypted before verifying the signature. The Decryption Transform for XML Signature specification enables XML signature applications to distinguish between documents encrypted before signing and documents encrypted after signing.

A sample XML code for an embedded XML Signature is provided in Listing 2: Embedded XML Signature on page 35. In this example, a request for a Web service is made over HTTP using a SOAP message including an XML signature.

### 7.3 Standards for Trust Management

#### 7.3.1 XKMS (XML Key Management Specification)

XKMS is a specification for registering and distributing encryption keys for Public Key Infrastructure (PKI) in Web services. XKMS revolutionizes PKI by establishing a platform-independent set of standards that places portions of the PKI workload on the server side, thus freeing application resources for other processes. XKMS works with proprietary PKI solutions to integrate encryption, digital signatures (including revocation and processing of certificates) and authentication. It simplifies the steps necessary to implement key management in PKI, providing an easy and user-friendly method for secure transaction.

XKMS was designed for use with XML Signature and XML Encryption. XML Signature and XML Encryption provide a high level of security for XML documents, but do not address trust management (the handling of public and private keys), which is essential to successful PKI. XKMS provides the necessary trust management.

XKMS is comprised of two specifications: X-KISS (XML Key Information Service Specification) and X-KRSS (XML Key Registration Service Specification). K-KISS is the set of protocols that processes key information (located in an XML signature’s Key-Info element) associated with XML encrypted data, digital signatures and other aspects of public-key
cryptography. X-KRSS is the set of certificate-management protocols that addresses the life of a
digital certificate – from registration to revocation and recovery.

Listing 3: X-KRSS Registration Request for a Key Pair on page 36 shows a register request sent to
an X-KRSS server.

7.3.2 SAML (Security Assertion Markup Language)

SAML is an XML framework for sharing security information on the Internet through XML
documents. It is designed to:

- provide a standard way to describe existing security models;
- enable universal sharing of authentication and authorization information across
  enterprises;
- enable a platform-neutral solution with support for industry-standard transport
  protocols and messaging formats; and
- keep security frameworks independent of vendor implementation and architecture.

The SAML framework includes:

- Assertions: How people define authentication, attributes, and authorization decision
  information.
- A protocol: How people ask and get the assertions they need.
- Bindings and Profiles: How SAML documents (assertions) ride “on” (bindings) and
  “in” (profiles) industry-standards transport and messaging frameworks.

SAML Assertions

A SAML assertion makes statements about a subject (an individual or a service). There are three
kinds of statements: authentication, attribute, and authorization decision. SAML assertions can be
digitally signed (see 7.2.2 XML Signature above). They are issued by “authorities,” such as
security applications or Permission Management Infrastructures (PMIs). In practice, a single
authority produces and issues all three types of assertions.

All SAML assertions include the following common information:

- Issuer ID and issuance timestamp;
- Assertion ID;
- Subject (name and optional subject confirmation, for example a public key);
- Conditions under which an assertion is valid; and
- Advice on how an assertion was made.
SAML Authentication statements are typically designed for single sign-on (SSO) use cases.

Sample code for a simple SAML authentication assertion is provided at *Listing 4: Simple SAML Authentication Assertion* on page 38.

**SAML Protocol**
The SAML protocol defines interaction between a SAML requester and a SAML responder (Requester and responder must have a trusted relationship: they’re talking about the same subject). A request is made by a SAML-aware client, a response is returned by a security service. A SAML request may include queries for authentication, attribute, and authorization decision. All types of SAML requests are met with a common SAML response that may contain one or several assertions.

**SAML Bindings**
SAML bindings specify how SAML request-response message exchanges are mapped to standard messaging protocols. SAML defines a SOAP-over-HTTP binding whereby SOAP is used to query a SAML authority and get back a response.

**SAML Profiles**
SAML profiles specify how SAML assertions are inserted in, and extracted from, a message framework or protocol. SAML defines a Web browser profile and a WS-Security profile (part of the WS-Security specification).

**7.3.3 XACML (Extensible Access Control Markup Language)**

XACML is a markup language that allows organizations to communicate their policies for accessing online information. It defines which client can access information, what information is available to clients, when clients can access the information and how clients can gain access to the information.

XACML security policies can regulate information access using factors such as a client’s identity, the client’s method of authentication and the port through which the client is communicating.

Working with SAML, XACML policies can provide the basis for authoritative decision-making at a Policy Decision Point (PDP). The Policy Enforcement Point (PEP) sends an authorization decision query to the PDP, which consults established security policies. The request must contain the target resource to access and the operation to perform (read, write, etc.). The assertion may also contain optional authentication credentials. Files requiring different levels of security will
have different authentication standards. The PDP will return access decisions on the basis of the security policies in place.

See Listing 5: A sample XACML Policy on page 39 for an illustration of the XML codes.

7.4 Standard for Message Structure

7.4.1 WS-Security (Web Services Security)

WS-Security specifies SOAP security extensions providing data integrity (i.e. ensuring that the information included in a SOAP message has not been modified or destroyed), and data confidentiality (i.e., ensuring that the information included in a SOAP message is not accessed by an unauthorized party).

WS-Security defines how to attach signature and encryption headers to SOAP messages. It also provides profiles that specify how to insert the following types of security tokens in WS-Security headers:

- Username / Password digest (defines how a Web service consumer can supply a username as a credential for authentication. The username can be accompanied by an encrypted password).
- X.509 certificate (an X.509 certificate defines a binding between a public key and a set of attributes used for authentication, e.g., username, certificate issuer, etc.).
- Kerberos ticket (Kerberos tickets are security tokens used for authentication).
- SAML assertion (SAML assertions are security objects that define authentication, attributes, and authorization-decision information).
- XrML document (The Right Expression Language (REL) license tokens inserted in WS-Security headers are used for authorization and are based on the Extensible Rights Markup Language (XrML)).
- XCBF document (defines how to use the XML Common Biometric Format (XCBF) with the WS-Security specification).

Listing 6: WS-Security Header on page 40 shows a SOAP message, including a WS-Security header enclosing a Password Digest security token.

WS-Security and SAML

The WS-Security profile of SAML is based on a single interaction between a sender and a receiver:
• The sender (a Web service consumer) obtains one or more SAML assertions and / or assertion identifiers.
• The sender adds the assertions and / or assertion identifiers to a SOAP message using WS-Security headers.
• The sender sends the SOAP message to the receiver (a Web service provider).
• The receiver processes the assertions and / or assertion identifiers present in the SOAP message.

SAML assertions and references to assertion identifiers are contained in the <wsse:Security> element, which, in turn, is included in the <SOAP-ENV:Header> element as shown in Listing 7: SAML Assertion in a WS-Security Header on page 40.

Assertion identifier references and information about assertion retrieval services are included in the <wsse:SecurityTokenReference> element.

One or more <saml:AssertionIDReference> elements holding the assertion identifier references may be included within the <wsse:SecurityTokenReference> element. The URI attribute of the <wsse:Reference> element specifies the location of a SAML responder implementing the SAML SOAP binding. See Listing 8: Referring SAML information in WS-Security on page 40 for an illustration of the XML codes.

8 High-level Architecture for Secure Enterprise Web Services

A high-level logical architecture for securing enterprise-level Web services has been suggested by Netegrity Inc. [6]. It is summarized in this section.

8.1 Logical Architecture

The logical architecture is depicted in the figure below:
The model consists of the following layers of functionality:

- Protection and threat prevention (Network security)
- Access enablement (Identity and Access Management -- IAM)
- Business policy enforcement (Web services management -- WSM)

Figure 3 shows the SOAP message flow going in the enterprise to the Web services provider, and the SOAP message flow coming out of the enterprise from the Web services consumer to outside partners.

Traditional network firewalls and security appliances make up the protection and threat prevention layer (network security). Identity and access management (IAM) is the access enablement layer, Web services management is the business policy enforcement layer, and Web services are deployed to and hosted by a container or a group of containers.

Several logical architecture layers may be implemented in a single product. For example, security appliances combine network security with some identity services (e.g., authentication). Often, containers include some components of the WSM and IAM layers. This logical architecture model is consistent with these and other possible implementation realizations.
8.2 Guidelines for deploying networks of Web Services in an enterprise

This section briefly highlights the general security properties that enterprises should pay attention to when they deploy networks of Web services in their organization. They are:

- Abstracting security from Web services development;
- Complying with Web services security standards;
- Relying on centralized, policy-based security management;
- Delivering the IAM (Identity & Access Management) layer as a set of shared services;
- Integrating with existing enterprise security systems.

Further elaboration on these guidelines can be found in [6, p. 8-10].
Part III – Comparison of Two “Identity Federation” Specifications

9 Introduction

In this Part III, the following sets of specifications are compared and contrasted:

(1) **Identity Federation Framework** (referred as **ID-FF** hereafter) developed by the Liberty Alliance Project (or Liberty); and

(2) **Web Services Federation** (referred as **WS-Federation** hereafter) jointly developed by IBM, Microsoft, BEA Systems, Inc., RSA Security, Inc., Verisign, Inc.

ID-FF and WS-Federation are specifications developed to address the *identity federation* need in the Web services world. The computing industry and the corporate users hold a general opinion that these two specification sets share similar objectives and address similar problems.

Before we proceed to the comparison in Section 12, let’s have a look into the following information, which serves as background reference:

- introduction to Federated Identity Management in Section 9.1,
- overview on ID-FF in Section 10, and
- overview on WS-Federation in Section 11.

9.1 Federated Identity Management

The appeal of *federations* is that they are intended to allow a user to seamlessly traverse different sites within a given federation. Federations provides 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 (e.g. another sign-on with a different pair of ID and password). In addition, Federation standards also deal with the matter of providing trusted attributes about users (e.g. including roles and group information) allowing for privacy and business-specific rules.

**Federated Identity Management (FIM)** defines the process by which a relying party (e.g. a Web site) is able to determine a locally valid identifier for the user based on the trusted information received from the issuing party.

Federated **Single Sign-On (SSO)** for users is made possible as a result of the participation of the Web services providers in FIM within a trusted domain and/or across trusted domains.
The flexibility and appeal of the Web Services model can only be successful if it is appropriately supported by FIM and federated SSO, which are key components in the Web services security landscape.

More information and reference on Identity Federation can be found from [7, 8, 9].

10 Overview of Identity Federation Framework (ID-FF)

According to the Liberty Alliance Specification Architecture [7, p. 6], Identity Federation Framework (ID-FF) is the first set of specifications developed by Liberty as their Phase 1 deliverables.

The ID-FF (Version 1.0) specification set was first released in July 2002. That version was updated in January 2003 to Version 1.1 and turned over to OASIS. Version 1.2 has just been released in November 2003 as part of the Liberty Phase 2 final specification set.
10.1 Overview

ID-FF enables identity federation and management. It can be used on its own or in conjunction with existing identity management systems. It is designed to work with heterogeneous platforms and with all types of network devices, including personal computers, mobile phones, PDAs and other emerging devices.

This specification defines a set of protocols that collectively provide a solution for identity federation management, cross-domain authentication, and session management. The Liberty architecture contains three actors: Principal, identity provider, and service provider. A Principal is an entity (for example, an end user) that has an identity provided by an identity provider. A service provider provides services to the Principal.

Once the Principal is authenticated to the identity provider, the identity provider can provide an authentication assertion to the Principal, who can present the assertion to the service provider. The Principal is then also authenticated to the service provider if the service provider trusts the assertion. An identity federation is said to exist between an identity provider and a service provider when an identity provider issues assertions with a persistent name identifier regarding a particular Principal to the service provider. This specification defines a protocol where the identity of the Principal can be federated between the identity provider and the service provider. Service providers can also request a non-persistent, one-time only, anonymous name identifier for the Principal.

ID-FF relies on the SAML specification. In SAML terminology, an identity provider acts as an Asserting Party and an Authentication Authority, while a service provider acts as a Relying Party. Interestingly, ID-FF Version 1.1 is in turn used as a foundation document to help create Version 2 of OASIS’s SAML.

10.2 Features

- **Opt-in Account Linking**
  Allows a user with multiple accounts at different Liberty enabled sites to link these accounts for future authentication and sign-in at these sites (i.e. federation).

- **Simplified Sign-On**
  Allows a user to sign-on once at a Liberty ID-FF enabled site and to be seamlessly signed-on when navigating to another Liberty-enabled site without the need to authenticate again. Simplified sign-on is supported both within a circle of trust and across circles of trust.
• **Fundamental Session Management**
  Enables companies or organizations that link accounts to communicate the type of authentication that should be used when a user signs-on. It also enables “global sign-out”, i.e., once users sign-out of a Liberty-enabled site, they can be automatically signed-out on all the sites they have linked to in that session.

• **Affiliations**
  Enables a user to choose to federate with a group of affiliated sites, which is critical in addressing the needs of portals and in the business-to-employee environment.

• **Anonymity**
  Enables a service to request certain attributes without needing to know the user’s identity.

• **Protocol for the Real-time Discovery and Exchange of Meta Data**
  For providers to communicate with each other, they must have previously obtained metadata regarding each other such as X.509 certificates and service endpoints. This feature of the ID-FF facilitates the real-time exchange of this information between Liberty-compliant entities.

### 10.3 Protocols

The ID-FF (Version 1.2) protocol suite consists of the following protocols:

1. **Single Sign-On and Federation**
   - The protocol by which identities are federated and by which single sign-on occurs.

2. **Name Registration**
   - The protocol by which a provider can register an alternative opaque handle (or name identifier) for a Principal.

3. **Federation Termination Notification**
   - The protocol by which a provider can notify another provider that a particular identity federation has been terminated (also known as de-federation).

4. **Single Logout**
   - The protocol by which providers notify each other of logout events.

5. **Name Identifier Mapping**
   - The protocol by which service providers can obtain (often encrypted) name identifiers corresponding to an identity federation in which they do not participate.
11 Overview of WS-Federation

The WS-Federation specification set is jointly developed by IBM, Microsoft, BEA Systems, RSA Security, and Verisign. Its current version is Version 1.0, which was released for public review on July 8, 2003.

WS-Federation is one of the seven specification sets proposed in the “Security in a Web Services World: A Proposed Architecture and Roadmap” whitepaper [10, p. 8-11]. See figure below.

![Diagram of IBM / Microsoft Specification Architecture](image)

**Figure 5: High Level Overview of IBM / Microsoft Specification Architecture [10, p.8]**

11.1 Overview

**WS-Federation** allows a set of organizations to establish a single, virtual security domain. It defines mechanisms used to enable identity, account, attribute, authentication, and authorization federation across different trust realms. The specification extends the WS-Trust model to allow attributes and pseudonyms to be integrated into the token issuance mechanism to provide federated identity mapping mechanisms. The models defined in **WS-Security**, **WS-Trust**, and **WS-Policy** provide the basis for federation; WS-Federation extends this foundation by describing how these models are combined to enable richer trust realm mechanisms across and within federations.

IBM/Microsoft said that they would likely submit this specification to a standards-body, but did not say when.

**Supporting profiles:**

1. **Passive Requestor Profile**

   which describes how the cross trust realm identity, authentication and authorization federation mechanisms defined in WS-Federation can be utilized by passive
requestors such as Web browsers to provide Identity Services (under the HTTP protocol); and

(2) **Active Requestor Profile**

which defines how federation mechanisms are used by active requestors such as SOAP-enabled applications.

Participation in a federation requires knowledge of metadata such as policies and potentially even WSDLs and schemas for the services within the federation. It is part of the WS-Federation design roadmap to include both WS-Policy and WS-MetadataExchange for mechanisms for metadata exchange. The WS-Policy and WS-MetadataExchange are specifications that outline a variety of specified mechanisms for services to acquire metadata. WS-MetadataExchange specification has still not yet been released for public review at the time of this writing.

12 **Comparison: ID-FF vs. WS-Federation**

12.1 **Scope of Comparison**

The comparison is conducted from a non-technical perspective. This is because the author is taking the executive stream of her Master degree course in eCommerce, which does not emphasize on the technical perspective. Another reason is due to time constraint. It is expected that extensive time would be required for going through and understanding the technical aspects of these specifications.

In this comparison, the two specification sets are compared and contrasted in terms of the following two types of factors:

(1) **Non-functional factors** (related to the individual characteristics of the specs);

(2) **Functional factors** (related to the features and functionality provided in the specs).

12.2 **Comparison on Non-functional factors**

12.2.1 **Goal**

Both ID-FF and WS-Federation are specifications for handling identity federation in Web services. They share similar goal but differ in the way they fit into their respective Liberty and IBM/Microsoft specification architecture for FIM (see 12.2.2 below).
12.2.2 Role assumed within the overall Specification Architecture

Referring to Figure 4 on page 23, we can see that, within the overall Liberty Alliance Specification Architecture, ID-FF’s role is to enable the “identity federation and management” through features such as identity/account linkage, simplified sign-on, and simple session management. It does not cover the next-level requirements such as the creating, discovering, and consuming identity services, permission-based attribute sharing, and security profiles. These requirements are addressed by ID-WSF.

ID-FF version 1.0, as the first set of Liberty specifications released in July 2002, served to provide the plumbing for a federated network identity management approach [7, p. 12]. When ID-FF version 1.2 was released in November 2003, its scope had been extended to cover cross-domain authentication, with new protocols for name registration, federation termination, and single logout.

In contrast, WS-Federation represents the key place within the IBM/Microsoft Web services security specification architecture where the FIM and federated SSO framework is to be defined (see Figure 5 on page 26). WS-Federation’s role, within this specification architecture, is to describe how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, sharing of attributes, and management of pseudonyms [8, p. 8]. WS-Federation is intended to be “the” specification for FIM and SSO within the WS-Security family of specifications. Therefore, its scope is broader than that of ID-FF. WS-Federation overlaps with the ID-FF functionality as well as some of the ID-WSF functionality (i.e. in the areas of identity services discovery, security profiles, and SOAP bindings).

12.2.3 Maturity

The Liberty Alliance Project was established in September 2001. After ten months, Liberty released its ID-FF Version 1.0 in July 2002, which was followed by a minor enhancement release (Version 1.1) in January 2003. Version 1.2 has just been released in November 2003.

In contrast, WS-Federation V1.0 was first released in July 2003 for public review. A WS-Federation feedback workshop has just been conducted at the Verisign offices on November 20, 2003, where it was attended by 10 other companies [11].

If we look at the functionality common in ID-FF V1.2 and WS-Federation V1.0, ID-FF specification is more mature than WS-Federation.

12.2.4 Standardization Approach and Current Status

ID-FF is developed by the Liberty Alliance Project, a consortium consists of more than 160 companies, non-profit and government organizations from around the globe. All the
specifications and standards developed by Liberty are non-proprietary, open standards. Any companies or organizations can adopt and implement these standards without the need of paying royalty. Liberty also commits to turning its specifications to open industry standards, as the case with ID-FF Version 1.1, which was passed to OASIS soon after publication.

In contrast, WS-Federation is jointly developed by a group of five software vendors, led by IBM and Microsoft. All the WS-Federation documents produced by this group are copyrighted. The copyright owners have granted the public the permission to copy and display the WS-Federation specifications free of fee and royalty, provided that the copyright notice is attached to any copy or display of the specifications. However, there is no explicit statement on whether companies implementing these specifications are required to pay any royalty. There is also no information from the IBM or Microsoft web sites regarding any plan to turn WS-Federation over to a standards body.

ID-FF is an open-standard whereas WS-Federation is currently a published draft produced by a group of leading software vendors.

12.2.5 Implementation Costs
As mentioned in Section 12.2.4, Liberty specifications including ID-FF are free to implement in products and services. For WS-Federation, the specifications are free to review; but IBM/Microsoft do not reveal a clear position on whether they will charge implementers any implementation and distribution costs.

12.2.6 Industry Adoption
ID-FF is supported in more than 25 products as of November 2003 [12]. Liberty has recently announced that 9 companies (constisting of vendors, manufacturers and services providers) had passed the first conformance test on ID-FF V1.1 and 5 other companies have plans to support the Liberty Phase 2 Specifications (which include ID-FF V1.2) in their existing and future products and services. See www.projectliberty.org.

In contrast, with reference to the “WS Protocol Workshop Process” [13], WS-Federation is still at the “published draft” stage today. In September 2003, SourceID (see www.sourceid.org), an organization for open source FIM, released a proof-of-concept for WS-Federation for the .NET platform.

In terms of industry adoption, ID-FF has achieved a much stronger position than WS-Federation as of today.
12.2.7 Scope of contents

ID-FF Version 1.2 specification set consists of 3 non-normative documents and 2 normative documents as listed below:

**Non-normative:**
- ID-FF Architecture Overview
- ID-FF Implementation Guidelines
- ID-FF Static Conformance Requirements

**Normative:**
- ID-FF Protocols and Schema Specification
- ID-FF Bindings & Profiles Specification

WS-Federation V1.0 specification set consists of the following documents:
- Web Services Federation Language (WS-Federation)
- WS-Federation: Active Requestor Profile
- WS-Federation: Passive Requestor Profile

We can see that the ID-FF specification set is richer in content than the WS-Federation specification set. Besides technical specifications, the ID-FF set also includes documentation on architecture, implementation guidelines and conformance requirements. WS-Federation only concentrates on technical specifications.

Dave Kearns, a writer and consultant in Silicon Valley, wrote [14]:

“one difference between WS-Federation and the Liberty Alliance specification for federated identity is that WS-Fed is the product solely of technology vendors while the Liberty Alliance, which has tech vendors as members, is primarily composed of consumer products vendors, government and nonprofit organizations. Reflecting that makeup, Liberty is quick to point out that technology is only part of the federation problem, and in many ways it’s the easiest part to solve.”.

12.2.8 Underlying Technology

ID-FF’s underlying technology extends and builds on SAML and relies on SSL and WS-Security for transport and message security.

WS-Federation’s underlying technology builds on WS-Trust, WS-Policy, and WS-Metadata foundation and relies on SSL and WS-Security for transport and message security.
### 12.3 Comparison on Functional Factors

The comparison on functional factors is presented in table form as follows:

<table>
<thead>
<tr>
<th>Feature / Functionality</th>
<th>ID-FF</th>
<th>WS-Federation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Federation</td>
<td>Account federation is done via Identity Mapping enabled by opaque identifiers (a key privacy feature).</td>
<td>Account federation is done via Identity Mapping enabled by the Pseudonym Service.</td>
</tr>
<tr>
<td>Client Profiles</td>
<td>Specifies client profiles for both browser and smart clients.</td>
<td>- same -</td>
</tr>
<tr>
<td>SSO control flow</td>
<td>SSO control flows specify both front-and-back channel mechanisms.</td>
<td>SSO control flows specify and strongly recommend the front-channel mechanism and mentions, but discourages use of “pointer-based” back-channel mechanisms.</td>
</tr>
<tr>
<td>Single Logout</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Federation Termination</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Trust Brokering</td>
<td>Yes</td>
<td>Built on WS-Trust</td>
</tr>
<tr>
<td>Cross-domain authentication</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Privacy Management</td>
<td>Through NameRegistration protocol</td>
<td>Through Set messages of Pseudonym service</td>
</tr>
<tr>
<td>Attribute Service</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Identity Service Discovery</td>
<td>No (handled by ID-WSF)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 12.4 Discussion

Overall, the Liberty Alliance specifications, including the ID-FF specifications, have got a clear focus on business, something lacking in much of the IBM/Microsoft Web Services Initiatives, including the WS-Federation.

The Liberty Alliance originally articulated a strong business case with a lot of blanks to be filled about the technology. For example, Liberty only specifies the SAML as its transport mechanism
whereas WS-Federation supports SAML, Kerberos, X.509 certificates, XrML, and can be extended to other methods.

WS-Federation, on the other hand, is a specification from two high-profile technology leaders. It is heavy on the technology of federation but lacks the necessary emphasis on business requirements and issues. The WS-Federation suite says nothing about the agreements, trust and liability that federation brings to the various parties in a transaction, while the Liberty Alliance has a series of well-thought out scenarios and recommendations for the business agreements that are a necessary part of identity federation.

Though some industry experts and corporate executives are pressing for a convergence of these two specifications for the sake of interoperability and standardization, so far Liberty and the IBM/Microsoft tandem are continuing to jockey for a leadership position.

Since the Liberty specifications are strong in addressing the business issues of FIM and the IBM/Microsoft Web Services Initiatives are heavy on technology, a combination of the two specifications may be what will succeed in the real world.
13 Conclusion

The biggest concern regarding Web services security is the immaturity of the underlying standards. As with any new technology, certain vulnerabilities are not discovered until attacks occur in a real-world setting. Individuals and organizations must take extra steps to protect their computing resources and assets when they allow external access to their applications and data through Web services. Web services security is an ongoing process, not a one-time solution. Web services security standards and technologies are evolving and emerging at a rapid pace. Anyone who have an interest in keeping up their awareness and knowledge in Web services security need to stay appraised of all ongoing developments in this area.

14 Future Work

Web service security is a big topic. The scope of this independent study has been set to focus on the key industry standards for securing Web services and on the ID-FF and the WS-Federation specifications for federated identity management. Through this independent study, the author has developed foundational knowledge on the topic and an interest in further study on the federated identity management arena. The groundwork established in this independent study is an asset for pursuing future research on federated identity management. It is suggested that future work to be carried out to research on federated identity management from the business perspective.
Appendix A: Listings of Sample XML Codes

15 Listing 1: Simple XML Encryption

```xml
<?xml version="1.0"?>
<PaymentInfo xmlns="http://www.example.com/payment">
  <CreditCard>
    <Name>Marc Chanliau</Name>
    <CreditCardNumber>
      <EncryptedData xmlns="http://www..." Type="http://www...">
        <CipherData>
          <CipherValue>A23B45C56</CipherValue>
        </CipherData>
      </EncryptedData>
    </CreditCardNumber>
    <ExpireDate>06/03</ExpireDate>
  </CreditCard>
</PaymentInfo>
```
16 Listing 2: Embedded XML Signature

POST/RatingService HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "getCreditRating"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAPENV="http://…">
  <SOAP-ENV:Header>
    <!-- The signedInfo element allows us to sign any portion of
    a document, in this case, we sign the body -->
    <CanonicalizationMethod Algorithm="http://www…"/>
    <SignatureMethod Algorithm="http://www…"/>
    <Reference URI="#Body"/>
    <DigestMethod Algorithm="http://www…"/>
    <DigestValue>
o+jtqlieRtF6DrUbX8O9M/CmySg=…
    </DigestValue>
  </Reference>
</SignedInfo>

  <!-- Following is the result of running the algorithm over
  the document. If changes are made to the document,
  the SignatureValue is changed. The security application
  verifies the SignatureValue, extracts the X509
  certificate and uses it to authenticate the user -->
  <SignatureValue>oa+tthsvsFIETRD2oNC5iRu2eIoqWpD6PVYIKqc…</SignatureValue>
</KeyInfo>

  <!-- Following is the public key that matches the
private key that actually signs the document -->
  <RSAKeyValue>
    <Modulus>5TT/o0lzTiP++LS6GLQM8x0FFrA12Q…
    </Modulus>
    <Exponent>EQ==</Exponent>
  </RSAKeyVal...
</KeyInfo>

  <!-- Following is the certificate -->
  <X509Data>
    <X509Certificate>
      MIIBwDCCAXqgAwIBAg…
    </X509Certificate>
  </X509Data>
</KeyInfo>
</Signature>
</SOAP-ENV:Header>
</SOAP-ENV:Body>

<!-- Here we can have any business payload such as a purchase
order. This is the part we signed in this example -->
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Listing 3: X-KRSS Registration Request for a Key Pair

<?xml version="1.0" encoding="UTF-8"?>
<!-- X-KRSS Registration Request for a key pair -->

<Register>
  <Prototype Id="keybinding">
    <Status>Valid</Status>
    <KeyID>mailto:joesmith@example.com</KeyID>
    <KeyInfo>
      <KeyName>mailto:joesmith@example.com</KeyName>
      <KeyValue>
        <X509Data>
          <X509SubjectName>
            CN=Joe Smith, STREET=123 Example Street, L=Maynard, ST=MA, C=US
          </X509SubjectName>
          <X509Certificate>
            CWDCCAgICAQAwDQYJKoZIhvcNAQEEBQ . . . . . . . .
          </X509Certificate>
        </X509Data>
        <PassPhrase>
          Wio923482hUSHuda389OHIA04u3jMNBZhduiWIUYTIUWYoiiH3748i=
        </PassPhrase>
      </KeyInfo>
      <AuthInfo>
        <AuthUserInfo>
          <ProofOfPossession>
            <Signature xmlns="http://www.w3.org/TR/xmldsig-core">
              <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
                <SignatureMethod Algorithm="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212#sha1" />
                <Reference URI="#keybinding">
                  <DigestMethod Algorithm="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212#sha1" />
                  <DigestValue>
                    8JiS0H32zISULqWerualYR1298Lsaeu24WHE79=
                  </DigestValue>
                </Reference>
              </SignedInfo>
            </Signature>
          </ProofOfPossession>
        </AuthUserInfo>
      </AuthInfo>
    </KeyInfo>
  </Prototype>
<SignedInfo>

<SignatureValue>
P98395u04HUSIJS586HDWHa8y98OPHEu3oUIUSu324eh
</SignatureValue>

</Signature>
</ProofOfPossession>

<KeyBindingAuth>
  <Signature xmlns="http://www.w3.org/TR/xmldsig-core">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <SignatureMethod Algorithm="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212#sha1" />
      <Reference URI="#keybinding">
        <DigestMethod Algorithm="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212#sha1" />
        <DigestValue>
fHdnkj394jnKJXOSDJ451RUGP932A94IHFIdlohr
984HiuZXBNMBDLiOWEIUyfhas8943uihSdho=
</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
Bo39udIOSijeAZjlMXioisd9u23rKLJSDPASe3fU1232u
Sk0291jfjzSIQ82uiouderiyq90UAES95yyhrus398d=
</SignatureValue>
  </Signature>
</KeyBindingAuth>
</AuthUserInfo>
</AuthInfo>

<Respond>
  <string>KeyName</string>
  <string>X509Cert</string>
</Respond>

</Register>
Listing 4: Simple SAML Authentication Assertion

<saml:Assertion
    AssertionID="10.255.1.3.1034108172377"
    IssueInstant="2002-10-08T20:16:12.377Z"
    Issuer="TransactionMinderSAMLIssuer"
    MajorVersion="1" MinorVersion="0"
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
    <saml:Conditions
        NotBefore="2002-10-08T20:16:12.307Z"
        NotOnOrAfter="2002-10-08T22:16:12.307Z"/>
    <saml:AuthenticationStatement
        AuthenticationInstant="2002-10-08T20:16:12.307Z"
        AuthenticationMethod="urn:oasis:names:tc:SAML...">
        <saml:Subject>
            <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.0:..."
                NameQualifier="Domain Name">
                Marc Chanliau
            </saml:NameIdentifier>
            <saml:SubjectConfirmation
                <saml:ConfirmationMethod>http://www...</saml:ConfirmationMethod>
                <saml:SubjectConfirmationData>R1VD8fkvlrhp...</saml:SubjectConfirmationData>
            </saml:SubjectConfirmation>
        </saml:Subject>
    </saml:AuthenticationStatement>
</saml:Assertion>
19 Listing 5: A sample XACML Policy

<?xml version="1.0" encoding="UTF-8"?>
<!-- XACML policy that restricts access to purchase histories -->

<rule effect="permit"
ruleID="http://www.BobsAppliances.com/customers/policy1"
xmlns="http://www.oasis-open.org/committees/xacml/docs/draft-xacml-schema-policy-12.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oasis-open.org/committees/xacml/docs/draft-xacml-schema-policy-12.xsd">
<description>
Allow customers to access and read their purchase history at BobsAppliances.com
</description>

<target>

<subjects>
  <saml:Attribute AttributeName="RFC822Name"
  AttributeNamespace="http://www.BobsAppliances.com">
    <saml:AttributeValue>*</saml:AttributeValue>
  </saml:Attribute>
</subjects>

<resources>
  <saml:Attribute AttributeName="documentURI"
  AttributeNamespace="http://www.BobsAppliances.com">
    <saml:AttributeValue>
    </saml:AttributeValue>
  </saml:Attribute>
</resources>

<actions>
  <saml:Action>read</saml:Action>
</actions>

</target>

<condition>
  <equal>
    <saml:AttributeDesignator AttributeName="requestor"
    AttributeNamespace="http://www.oasis-open.org/committees/xacml/docs/identifiers" />
    <saml:AttributeDesignator AttributeName="customerID"
    AttributeNamespace="http://www.BobsAppliances.com/customers/record/custID" />
  </equal>
</condition>
</rule>
20 Listing 6: WS-Security Header

```xml
  <SOAP-ENV:Header>
      <wsu:Timestamp wsu:Id=“W3dz2rZ0W6f-2232-66c1b9-0605-18500160295”>
        <wsu:Created wsu:Id=“WS76720766-f43a-5a3e-2a05-a8bae3a762d”>2003-01-30T17:08:20Z</wsu:Created>
      </wsu:Timestamp>
      <wsse:UsernameToken wsu:Id=“WS3d7bf6c8-0d0f-f3c4-000d-2f3133be85c8”>
        <wsse:Username>echaniloe</wsse:Username>
        <wsse:Password Type="wsse:PasswordDigest”>MgyW8WzXnrc8w/YpWkAm3CYCR</wsse:Password>
      </wsse:UsernameToken>
      <wsse:Security>
        <wsse:BillingInformation>
          <CustomerCredentials>
            <Username>echaniloe</Username>
          </CustomerCredentials>
        </wsse:BillingInformation>
      </wsse:Security>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <purchaseOrder orderDate=“2002-03-06” xmlns=“urn:proc:order”>
      Shipping Information inserted here</purchaseOrder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

21 Listing 7: SAML Assertion in a WS-Security Header

```xml
<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    <wsse:Security>
      <saml:Assertion/>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <purchaseOrder orderDate=“2002-03-06” xmlns=“urn:proc:order”>
      Shipping Information inserted here</purchaseOrder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

22 Listing 8: Referring SAML information in WS-Security

```xml
<wsse:SecurityTokenReference>
  <saml:AssertionIDReference>
    XVB12921abc...
  </saml:AssertionIDReference>
  <wsse:Reference URI=“http://www.example.com/SAMLservice”/>
</wsse:SecurityTokenReference>
```
# Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed Component Object Model</td>
</tr>
<tr>
<td>EAI</td>
<td>Enterprise Application Integration</td>
</tr>
<tr>
<td>FIM</td>
<td>Federated Identity Management</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization of the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>PMI</td>
<td>Permission Management Infrastructure</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery, and Integration</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>X-KISS</td>
<td>XML Key Information Service Specification</td>
</tr>
<tr>
<td>X-KRSS</td>
<td>XML Key Registration Service Specification</td>
</tr>
<tr>
<td>XACML</td>
<td>eXtensible Access Control Markup Language</td>
</tr>
<tr>
<td>XCBF</td>
<td>XML Common Biometric Format</td>
</tr>
<tr>
<td>XKMS</td>
<td>XML Key Management Specification</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>XrML</td>
<td>eXtensible rights Markup Language</td>
</tr>
</tbody>
</table>
References

23 Cited References


### 24 Web resources referenced

#### 24.1 W3C standards

**XML Encryption:**
- XML Encryption Working Group Homepage  
  [http://www.w3.org/Encryption/2001/](http://www.w3.org/Encryption/2001/)
- Charter of XML Encryption Working Group  
  [http://www.w3.org/Encryption/2001/01/xmlenc-charter.html](http://www.w3.org/Encryption/2001/01/xmlenc-charter.html)
- XML Encryption Activity Statement  
  [http://www.w3.org/Encryption/2001/Activity.html](http://www.w3.org/Encryption/2001/Activity.html)
- XML Encryption Core Specification  

**XML Signature:**
- XML Signature Working Group Homepage  
  [http://www.w3.org/Signature/](http://www.w3.org/Signature/)
- Charter of XML Signature Working Group  
  [http://www.w3.org/Signature/charter-20020607](http://www.w3.org/Signature/charter-20020607)
- XML Signature Activity Statement  
  [http://www.w3.org/Signature/Activity.html](http://www.w3.org/Signature/Activity.html)
- XML Signature Core Specification  

**XML Key Management Specification (XKMS):**
- XKMS Working Group Homepage  
  [http://www.w3.org/2001/XKMS/](http://www.w3.org/2001/XKMS/)
- Charter of XKMS Working Group  
  [http://www.w3.org/2001/XKMS/2003/06/charter.html](http://www.w3.org/2001/XKMS/2003/06/charter.html)
- XKMS Activity Statement  
  [http://www.w3.org/2001/XKMS/Activity](http://www.w3.org/2001/XKMS/Activity)
- XKMS Core Specification  

#### 24.2 OASIS Standards

**Web Services Security (WS-Security):**
- Web Services Security Technical Committee (TC) Homepage
http://www.oasis-open.org/committees/wss/charters.php

- Charter of Web Services Security TC
  http://www.oasis-open.org/committees/wss/charters.php


**Extensible Access Control Markup Language (XACML):**

- OASIS Extensible Access Control Markup Language Technical Committee (TC) Homepage
  http://www.oasis-open.org/committees/xacml/charters.php

- Charter of XACML TC
  http://www.oasis-open.org/committees/xacml/charters.php

- XACML Specification V1.1

**Security Assertion Markup Language (SAML):**

- OASIS Security Services Technical Committee (TC) Homepage

- Charter of Security Services TC

- SAML Specification V1.1

- Security and Privacy Considerations for SAML V1.1

### 24.3 General references

1. Web Services Security
   http://www.xwss.org/index.jsp

2. OASIS
   http://www.oasis-open.org/home/index.php

3. WWW consortium
   www.w3.org

4. Liberty Alliance Project
   http://www.projectliberty.org/

5. Microsoft Developers’ Network
   http://msdn.microsoft.com/webservices/
6. IBM
   http://www.ibm.com

7. Sun Microsoft’s Developer’s Network

8. O’Reilly’s xml.com & webservices.xml.com sites
   http://www.xml.com/
   http://webservices.xml.com

9. Web services Architect
   http://www.webservicesarchitect.com/

10. Netegrity Inc.
    http://www.netegrity.com

11. Federation ID management
    www.sourceid.org

12. XML Security reference page
    http://www.nue.et-inf.uni-siegen.de/~geuer-pollmann/xml_security.html

13. Network World Fusion online magazine
    www.nwfusion.com

25 Books referenced


