Web Services Based Single Sign-on for Hospital Management System

Savitri Bevinakoppa Ravi Sankar Tripuramallu
School of Computer Science and Information Technology, RMIT University
Melbourne – 3001, AUSTRALIA
savitri@cs.rmit.edu.au rtripura@cs.rmit.edu.au

Abstract

The potential of Internet has been credited with the ability to communicate and collaborate among the partners and transact among multiple organizations. Thus companies are fast moving in the direction of collaborative Business-to-Business (B2B) e-commerce providing the customers and business partners more flexibility and efficiency. Web services are fast replacing the traditional Enterprise Application Integration (EAI) solutions that suffer from interoperability issues. Securing web services is important and difficult as the services and clients span disparate platforms and security domains.

The aim of this paper is to implement efficient and secured single sign-on service for a Hospital Management System (HMS). The HM system consists of various web services like pathology, radiology, clinics etc. These services are exposed to other internal and external entities through registries. The user or doctor once authorized to use one of the services is entitled to access any of the other services based on the type of authentication performed and the privileges acquired.

1 Introduction

The collaboration among businesses has led to necessity in minimizing the number of hurdles the user has to go through to avail the co-operating services. This gives the user the overall liberty to browse a federation of services once authorized at one of the services. Since Web services allow loose coupling of applications unlike traditional distributed computing models like COM [1] and CORBA [2], allow the incrementing of systems. Web services are simpler to design and develop when compared to other EAI solutions [3].

Web services are based on open standards unlike other EAI solutions that are based on proprietary protocols. Web services allow dynamic integration based on the service-oriented architecture. This allows the consumers to bind at runtime based on the business logic rather on technology platform of the service [5]. This section gives the software and protocols used for developing single sign-on service for Hospital Management System.

1) Web logic server 7.0 – This application server is used to host the various services that collaborate.
2) Windows 2000 Professional – Operating system for hosting the Application Server.
3) IBM XSS4J- security suite for Encrypting XML Documents

Section 2 gives description of the web services components that are used for the implementation. Design and analysis of the HMS single sign-on services are explained in section 3 and 4 respectively. Section 5 gives the implementation of the service and conclusion of the paper is given in section 6.

1.1 Web services Components

Software/Protocols used for HMS are

Apache XML Security Project – for generating XML Digital Signatures

2 Web services

Web services can be defined as network accessible interfaces to applications exposing functionality. The applications exposed by web services are accessible with protocols such as HTTP and open standards like XML that are driving the Internet. The following terms/protocols used for Hospital Management Systems (HMS).

1. Interface: Every UDDI registry is provided with interfaces that allow consumers or providers to retrieve the information or browse the registry.

2. Web Services Description Language (WSDL): A web service exposes functionality and the operations that can be invoked. These operations are invoked with certain parameters or specific information.

3. Security in Web Services: SAML is a standard used in this application. This project implements secured single sign on system for an existing federation of hospital web services taken as a case study.

4. Security Assertion Mark-up Language (SAML): SAML is used to implement single sign on for web services at application level. SAML works over the existing security technologies and is a solution for open, web-based interoperable single sign-on service. SAML is a combination of S2ML and AuthXML.

3 Design of Hospital Management System

A Hospital management system is taken as a case study for the project. This was chosen because of the underlying service oriented architecture. This single sign-on enabled service oriented architecture allows the authenticated end users to access the exposed services. This section details design of the hospital management system and the steps involved in design and various components that make up hospital management system. While the initial sections cover high-level deployment diagram and component diagrams, later sections cover in depth of the use cases and scenarios involved.

3.1 Hospital Management System Components

The system comprises various web services – pathology, radiology, radiotherapy, clinical details etc. The goal of the system is to expose all the available services to the end users such as doctors who have the access rights to one of the services. Various departments in the hospital play the role of service providers. The service providers register with the Universal Description Discovery and Integration registry. The web services architecture allows all these registered services on disparate systems to form a federation.

3.2 Roles or Actors in the system

The doctor: Doctor is the end user of the system who avails the services. Authenticated doctor is given the access to the available services for which he is authorized.

The service: Service is the functionality offered to the end user and in this case to a doctor.

Pathology service: Pathology department records the pathology details of the patients and makes them available to the doctor in the form of service.

Radiology service: Radiology department maintains the records of the patients referred to the radiology department.

Clinical Centres: keeps the record of patients details such as patients personal information, consultations, referral doctors, details of diagnosis and treatment outcome etc.

3.3 Functionality of Hospital Management Solution

Every doctor accessing a service belongs to one of the security domains in which each service is hosted. The doctor authenticated at one of the services

has the capability to add, alter, view or delete patient details. The authenticated doctor when crosses the security domain to access a remote service, the doctor have the capability to view patient details provided by the remote service.

The single sign-on capability integrated into the system gives the doctors authenticate once and only once to avail any of the exposed services that form a federation. The doctor interacts with the system while authenticating or while availing the services.

4 Analysis of Single Sign-on Service for Hospital Management System

Authorization is the act of granting privileges based on the credentials. This act involves exchanging information about the authentication act carried out.

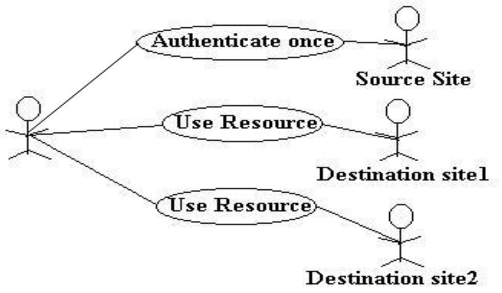

Fig. 1: Use case for single sign-on using SAML

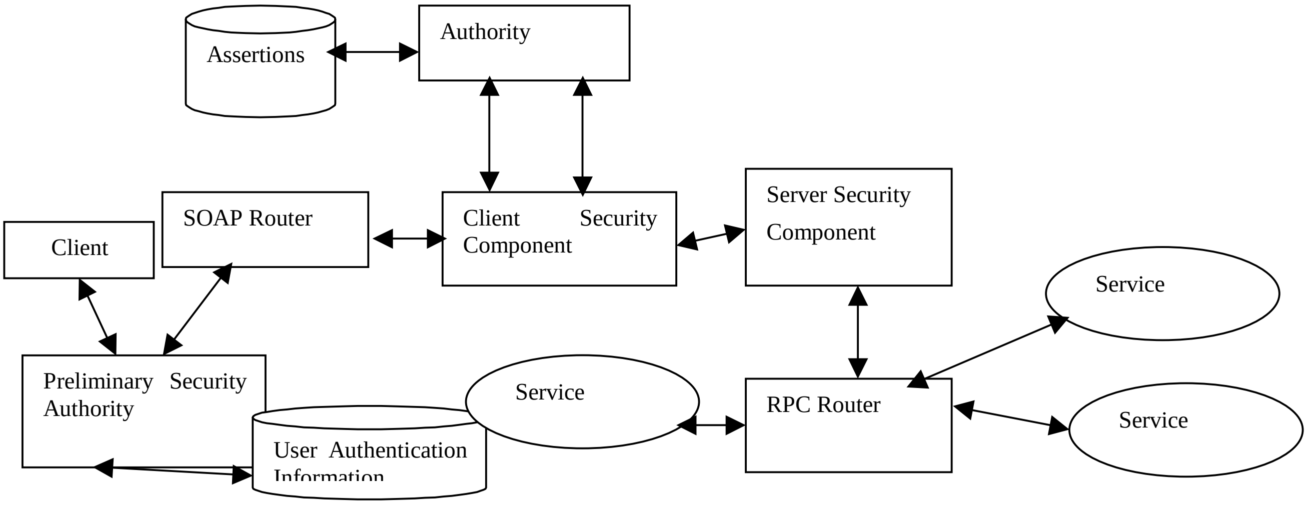
Assertions
Authority
Server Security Component
SOAP Router
Client Security Component
Client
Service
Preliminary Security Authority
Service
RPC Router
Service
User Authentication Information

Fig. 2 Components of SAML enabled Single Sign-On Web services Framework

In a federation, the user achieves the capability to roam around the federation server with out re-authenticating. The use case for single sign-on is depicted in figure 1. The user only needs to authenticate to one of the services of the federation and is granted permission to any other services of the federation. The source site is the site at which the user visits first and authenticates. A destination site is a site that the user accesses further on his authentication at the source site. Security Assertion Markup language (SAML) is about writing assertions and exchanging them. Assertions are nothing but the facts about the authentication events that have taken place. These assertions are exchanged to enable single sign-on in a web services infrastructure. The assertions are passed along with service requests.

4.1 Architectural Overview of single sign-on

Architectural overview is shown in figure 2. Preliminary Security Authority is responsible for initial authentication process for the end users. This authority is supported by a database, which holds the authentication information. This authority permits the users with proper id and password.

User Authentication Information Database assists preliminary security authority in basic authentication implementation. This holds the information about the users intend to login to the hospital management system.

SOAP Router is responsible for processing the SOAP requests and dispatching them to the services. SOAP Routers are also responsible for delivering the responses from the services back to the clients or request makers. SOAP Router takes the responsibility for browsing UDDI registry, WSDL descriptions of the services and association of requests to services.

The client security component is responsible for SAML requests. The client security component creates SAML requests depending on the type of assertion needed. These requests are then passed on to the authority. The client security component is also responsible for maintenance of artifacts for each login. The client security component also dispatches the artifacts to the server security component for further processing.

Server security component is the component that requests the processes the assertions. It is the responsibility of server security component to fetch the assertions from the authority. Server security component acts as the Policy Decision Point. It takes decision on permitting the requests to the services based on the assertions.

Authority is typically a third party entity that is trusted by the services that are forming a federation. The authority is the entity that issues the assertions based on the users method of authentications scheme and information. Authority is supported by a database for assertions.

Assertion database holds assertions for each login. This database provides the backend for authority. RPC Router dispatches the SOAP requests to the services. A service is the information provider for the authenticated users of the hospital management system. Services register themselves with UDDI registry to participate in the federation and also trust the authority and the assertions that are vouched by authority.

4.2 Steps Involved in Service Access

The following sequence diagram figure 3 depicts the time line sequence of steps involved in accessing a remote service. The steps involved initial authentication to a local service that is carried out of band and eventually accesses the desired target service.

As we can see in figure 3, the client security component initiates the SAML message flow: The client security component gets the requests only from the users that are authenticated by basic password authentication technique. The client security component provides the authority with the details of the authenticated user who is making request. The authority checks if the authenticated user is eligible to access the service as requested. Authority builds an assertion based on the authorization information and a reference to the assertion or so-called *artifact*. The artifact is passed back to the client security component as a response to the details submitted. The client security component holds the artifact for future access to other services. The artifact is passed to the server security component along with the service request. Server security component acts the gateway to the services hosted by the server. The authority security component builds the assertion request along with the artifact and dispatches it to the authority. Authority on attaining the artifact and the assertion request fetches the assertion that was built corresponding to the artifact. This assertion is dispatched as SAML Response to the server security component. The server security component processes the assertion received from the authority, analyzes the authorization and conditions. The server security component then decides whether to allow the access to the services.

Security of SAML messages is out of scope of SAML in this paper. Messages can be secured using XML digital signatures or SOAP Security Extensions.

5. Key Elements for Implementation of Single Sign-on for HMS

This section gives the protocols used and testing approach.

5.1 Protocols

SAML Artifact:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/env
elope/">
<SOAP-ENV:Body>
```

```
<samlp:AssertionArtifact Type-
Code="0x0001"
xmlns="urn:oasis:names:tc:SAML:1.0:cm:ar
tifact-01"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0
:protocol">
```

```
<SourceID>AU_AUTHORITY</SourceID>
<AssertionHan-
dle>1291c9e5</AssertionHandle>
</samlp:AssertionArtifact>
</SOAP-ENV:Body>
```

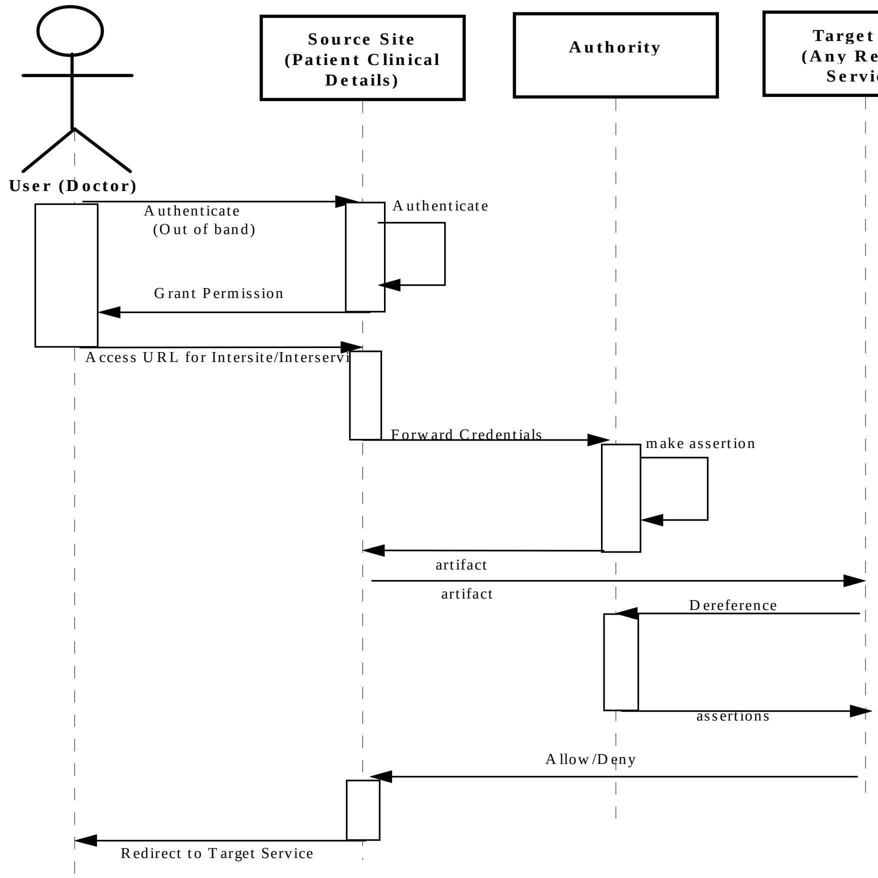
User (Doctor)
Source Site (Patient Clinical Details)
Authority
Target Site (Any Remote Service)
Authenticate (Out of band)
Authenticate
Grant Permission
Access URL for Intersite/Interservi
Forward Credentials
make assertion
artifact
artifact
Dereference
assertions
Allow/Deny
Redirect to Target Service

Fig. 3: Sequence Diagram to Access Remote Service – SAML message flow

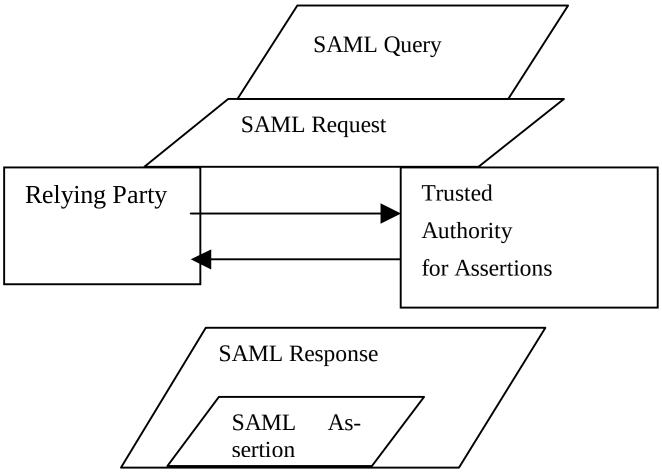

Fig. 4: Protocol underlying SAML

```
</SOAP-ENV:Envelope>
```

Every SourceId element refers to an URI. AU_AUTHORITY represents the URI of authority. The service maintains a list of SourceID and the corresponding URIs. The AssertionHandle is a randomly generated 64-bit string by the authority. The authority holds an assertion for each corresponding AssertionHandle. The assertion is returned when assertion handle is de referenced.

SAML Assertion

```
  <saml:AuthenticationStatement Authen-
ticationInstant="Sun Apr 14 15:09:02EST
2002" AuthenticationMethod="password" />
   <saml:AttributeStatement>
   <saml:Subject>
```

```
      <saml:NameIdentifier          For-
mat="urn:oasis:names:tc:SAML:1.0:asserti
on
    #emailAddress">doctor@hope.com
   </saml:NameIdentifier>
  </saml:Subject>
  <saml:Attribute                Attribu-
teName="Designation">
 <saml:AttributeValue>DOCTOR</saml:Attri
buteValue>
  </saml:Attribute>
<saml:Attribute                  Attribu-
teName="AllowedServices">
  <saml:AttributeValue>Pathology
</saml:AttributeValue>
  </saml:Attribute>
  </saml:AttributeStatement>
  </saml:Assertion>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

The value of the element, AttributeName="Designation" asserts that the user for which the assertion belongs is a doctor. The value of the element AttributeName="AllowedServices", asserts that the user has right to access Pathology service.

5.2 Securing of SAML messages: XML Encryption:

XML Encryption provides end-to-end security for XML data that is exchanged. The XML Encryption has certain advantages over Transport Layer Security (TLS) which is based on Secure Socket Layer (SSL). XML Encryption can encrypt the parts of XML documents that are exchanged.
An XML Encrypted document can carry both the secured and unsecured XML/binary data. In the case of securing SOAP/SAML messages through XML Encryption, only parts of SAML message are encrypted [13].
Symmetric key encryption assumes a key known as ‘secret key’ known only to both the sender and receiver. The sender uses this key for encryption and eventually for decryption by the receiver. SAML messages are exchanged in this fashion among the security components.

SOAP message header and body that envelop SAML message pass through intermediaries unencrypted. The <urn> element in SAML Artifact is encrypted using triple DES

Security needs for web services are similar to that of typical web based applications [12]. Securing of SAML messages is out side the scope of SAML. XML Encryption and XML Digital Signatures ensure confidentiality and non-repudiation aspects of securing SAML messages in an inter-operable fashion.

key and the key itself if encrypted with sender' s RSA key. It is assumed that keys are exchanged out of band and are available to the sender and receiver

Xml digital signatures:
An XML Digital signature ensures the integrity and authenticity of origin for XML messages. Unlike traditional signature mechanisms, with XML Digital signature, a sender has the capacity to sign only the portion of the document for which the sender is liable. This granularity is important as XML based SAML messages tend to pass several intermediaries on the way to destination. Each SAML message that is exchanged in implementing single sign-on service needs to be digitally signed. This ensures that each security component in the application has the confidence on any incoming SAML message.

5.3 Testing

Testing environment:
Application server: BEA Web logic version 7.0 hosts the services
The server runs on Pentium III processor with 128 MB RAM.
The client is a web browser, Internet Explorer 5.0
Operating System for Application Server is Windows 2000Professional

Testing Approach
The aim of the testing is to establish that the application developed enables single sign-on for the services in a secure fashion.

Opportune scenario
A Doctor logs in into Patient Clinical Details Service. Once he is logged in, the doctor should be able to browse the other services available.

Access to multiple services (Single sign-on)
Doctor logs into Patient Clinical Details service at the URL,
http://distributed28:7001/examplesWebApp/doctorlogin.jsp
.
The doctor is challenged to produce user id and password. The system allows only the doctor with correct user id and password with respect to the security domain in which Patient Clinical Details services is hosted.
The doctor is provided with Patient Clinical Details and also the links to other services. The doctor is not challenged for user id and password when he accesses the links to the other services. The doctor can browse through other services at will. This proves that the services are single sign-on enabled.

Accessing Individual Services.

An end user can access any individual service. The access is restricted to the users who have credentials with respect to that particular security domain. The authenticated users have the access only to the individual service.

In this case, when doctor logs into Patient Clinical Details Service, he has the access to the Patient Clinical Details Service. The doctor is restricted to that particular security domain or in other words, to that service alone. The doctor when tries to access the other services individually, access is denied. This is because the doctor does not possess the access rights to any other service other than Patient Clinical Details. Any user with access permission to any of the service can access that particular service alone.

The above test cases prove that access to the services is obtained only when the services participate in the federation and participate in single sign-on.

6. Conclusions and Future Work

This research demonstrated an implementation of a single sign-on feature for web services. It includes technologies that are used to accomplish the task. Traditional security technologies like Transport Layer Security, Encryption techniques have to go hand in hand with up-coming XML security standards like S2ML, SAML and XKMS to achieve total security for single sign-on enabled web services. Implementing the support security services using traditional security technologies is out side the scope of this project. SAML does not alone guarantee the security of single sign-on enabled web services.

The future work includes LDAP directories can be used for maintenance of assertions, Public Key infrastructure can be implemented to achieve over all security for a web services infrastructure. Currently signing and verification of SAML messages is painfully slow process. Work can be done in reducing the delays in message processing.

Acknowledgement

Authors would like to acknowledge Dr Kailash Narayan, Head of the Radiation, Peter MacCallum Cancer Institute for providing the information for Hospital Management Systems.

References

1. Nathan, A., *NET and COM: The Complete Interoperability Guide*, Sams Publisher; 1st edition (January 31, 2002), ISBN: 067232170X
2. Cerami, E., *Web Services Essentials (O'Reilly XML)*, O'Reilly & Associates; 1st edition (February 2002), ISBN: 0596002246
3. Graham, S., Simeonov, S., Boubez, T., et. Al.: *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI*, Sams Publisher; 1st edition (December 12, 2001) ISBN: 0672321815
4. Oellermann, W.: *Architecting Web Services*, APress; 1st edition (October 15, 2001) ISBN: 1893115585
5. http://www.w3.org/TR/wsa-reqs
6. Ray, E., Maden, C.: *Learning XML*, O'Reilly & Associates; 1st edition (February 2001) ISBN: 0596000464
7. Zuffoletto, J., Wells, G., Gill, B., et. Al.: *BEA Weblogic(R) Server Bible*, John Wiley & Sons; 1st edition (February 21, 2002) ISBN: 0764548549
8. http://www.ptsdirect.co.uk/saml.cfm
9. Steelman, A., Murach, J.: *Murach's Java Servlets and JSP*, Mike Murach & Associates; Book and CD edition (January 2003) ISBN: 1890774189
10. http://www.javaworld.com/javaworld/jw-12-1998/jw-12-servletapi.html
11. *Simple Object Access Protocol (SOAP) 1.1* W3C Note 08 May 2000 http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
12. Krishnamurthy, B., Mogul, J., Kristol, D.: *Key Differences Between HTTP/1.0 and HTTP/1.1*, WWW8 Conference Refereed Papers, Compiled from Authors' Original Papers by E. Tang, Foretec Seminars 1999