WEB SERVICE BASED RELIABLE - SHELTERED MEDI HELPER

L.Priya[1] , J. Jeyalakshmi[2], S. Usha[3]

Department of Information Technology, Rajalakshmi Engineering College, Thandalam.

pri_sun@yahoo.co.in [1]
balajeyalakshmi@gmail.com [2]
ushasarangapani@gmail.com [3]

Abstract: The veracity and secrecy of medical information which is transacted over the Internet is vulnerable to attack. But the transaction of such details is mandatory in order to avail the luxury of medical services anywhere, anytime. Especially in a web service enabled system for hospital management, it becomes necessary to address these security issues. It is mandatory that the services guarantee message delivery to software applications, with a chosen level of quality of service (QoS). This paper presents a VDM++ based specification for modelling a security framework for web services with non repudiation to ensure that a party in a dispute cannot repudiate, or refute the validity of a statement or contract and it is ensured that the transaction happens in a reliable manner. This model presents the procedure and technical options to have a secure communication over Internet with web services. Based on the model the Medi - Helper is developed to use the technologies of WS-Security, WS-Reliability and WS-Policy, WSRN in order to create encrypted messages so that the Patient's medical records are not tampered with when relayed over Internet, and are sent in a reliable manner. In addition to authentication, integrity, confidentiality, as proposed in this paper security framework for healthcare based web services is equipped with non repudiation which is not inclusive in many existing frameworks.

Keywords: Web Services, WS-Policy, WS-Reliability, METRO, WS-Security, Web Services Security, e – Healthcare, SOAP, VDM++, Security Framework, Formal

1. INTRODUCTION

Web services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Nowadays, an increasing amount of companies and organizations implement their core business and outsource other application services over Internet like healthcare applications which have a large customer base and wide application. Perimeter-based [20] network security technologies like firewalls are inadequate to protect SOAs for the following reasons: 1) SOAs are dynamic and can seldom be fully constrained to the physical boundaries of a single network. 2) SOAP is transmitted over HyperText Transfer Protocol (HTTP), which is allowed to flow without restriction through most firewalls. Moreover, Transport Layer Security (TLS), which is used to authenticate and encrypt Web-based messages, is inadequate for protecting SOAP messages because it is designed to operate between two endpoints. TLS cannot accommodate Web services' inherent ability to forward messages to multiple other Web services simultaneously.

The healthcare services [2,5] are widely used, when a user travels to different places it is necessary that the person's medical history is also available anywhere for easy access. But when it comes to the mode of transferring medical records [8] online, it is necessary to think about the security of the document as well, else it can be misused. The major security challenges are making the documents available to the right people and keeping the document from being viewed by anyone else and guaranteeing that the transfer is reliable. Keeping these challenges in

Nabendu Chaki et al. (Eds.): NeTCoM 2010,CSCP 01, pp. 26–40, 2011.
 DOI : 10.5121/csit.2011.1103

mind, a platform-neutral way for sharing medical records is proposed in this paper. It also becomes mandatory that the solution built is scalable and extensible making way for services like QOS and Security. To provide a fast secured medical services by making use of the fast growing web services [9], a medical assistant is developed using METRO STACK on jdk1.6.

Though Web service processing model can handle most attacks it needs to be further strengthened by means of enhancing and improving security. This paper presents a security framework in session II, to handle authentication, authorization, confidentiality, integrity and especially non-repudiation mechanisms along with reliable message transfer. It is presented as a specification in VDM++ so that it can be verified and proof analysis can also be done over the services. The Medi - Helper discussed in this paper is deployed to transfer the medical document across Internet in a secured manner and it is made available only to authorized people by providing good security

2. RELATED WORK

Web services expose the valuable XML-encoded healthcare information. Tampering the existing history or record will lead to heavily built problem even it may cause death without security. Web services might even make this situation worse. The reason is that Web services can be thought of as allowing in strange, new users who might take the existing hospital management system and may spoil the accessible database which is not likely to happen in case of the Medi – Helper due to the Single Sign On capability.

The [20]Medi – Helper may be are prone to following attacks and they have to be prevented against them.

- Message alteration - An attacker inserts, removes or modifies information within a message to deceive the receiver
- Loss of confidentiality - Information within a message is disclosed to an unauthorized individual
- Falsified messages - Fictitious messages that an attacker intends the receiver to believe are sent from a valid sender
- Man in the middle - A third party sits between the sender and provider and forwards messages such that the two participants are unaware, allowing the attacker to view and modify all messages
- Principal spoofing - An attacker constructs and sends a message with credentials such that it appears to be from a different, authorized principal
- Forged claims - An attacker constructs a message with false credentials that appear valid to the receiver
- Replay of message - An attacker resends a previously sent message
- Replay of message parts - An attacker includes portions of one or more previously sent messages in a new message
- Denial of service - An attacker causes the system to expend resources disproportionately such that valid requests cannot be met.

Prevention needs focus on Integrity, Confidentiality, Authentication, Authorization, Non-repudiation as suited for multi -tiered security. The following is a course of action proposed in order to secure the Medi – Helper which is in general applicable to any web service based application. The data flow of the same is presented in Figure 1.

The course of action is explained as below.

1. Message Level Security is ensured by keeping the SOAP messages from being viewed or modified by attackers as the messages traverse the Internet. The credentials are acquired from the user by the service which is left to the designer.

There are several options available for securing Web service messages[20]

- HTTP over SSL/TLS (HTTPS) Because SOAP messages are transmitted using HTTP, it is trivial to modify a Web service to support HTTPS.
- XML Encryption and XML Signature These XML security standards developed by W3C allow XML content to be signed and encrypted. Because all SOAP messages are written in XML, Web service developers can sign or encrypt any portion of the SOAP message using these standards, but there is no standard mechanism for informing recipients how these standards were applied to the message.
- WS-Security WS-Security was developed to provide SOAP extensions that define mechanisms for using XML Encryption and XML Signature to secure SOAP messages.
- SAML Authentication of SOAP Headers

2. Identity Management may follow any of the following architectures[20].

- Isolated identity management is the architecture used by most Web applications on the Internet. In isolated identity management, service providers act both as a credential provider and identity provider.

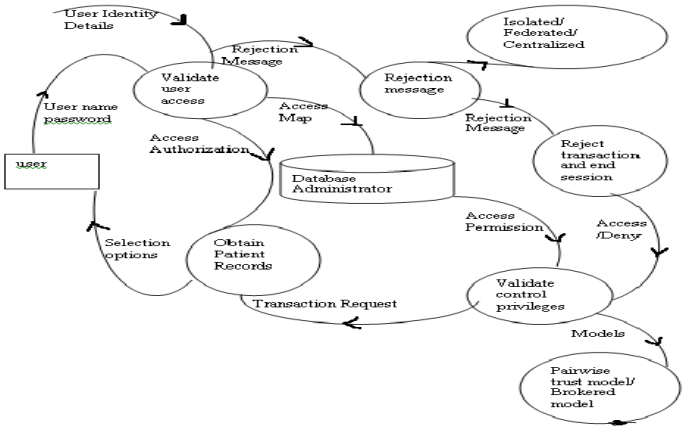

Figure 1 Context Layout Showing Medi – Helper

In identity federation, a group of providers agrees to recognize user identifiers from one another. Each service provider acts as a credential and identity provider for a subset of requesters.

In centralized identity management, providers rely on a single TTP to provide credentials and identifiers to requesters. Centralized identity management is similar to federated identity management in that the identity and credential providers supply assertions directly to service providers, allowing requester access without authenticating a second time.

3**. Session Management** is proposed to use the credentials of the user which are already secured but along with de identification. [22]Deidentification of medical records involves 2 steps:
(1) the identification of personally identifying references within medical text
(2) the masking, coding, and/or replacing of these references with values irreversible to unauthorized personnel.4 Some computation methods have been described previously to achieve this goal in medical text documents.

4. Resource Management [20] is done by ensuring that they are adequately protected. Usually, Web services are intended to be accessible only to authorized requesters, requiring mechanisms for access control. Several different methods are available, including transport layer authentication, token authentication via the WS-Security specification using SAML assertions or other tokens, and the SOAP authentication header.

5. Trust Management [20] Each trust model provides different benefits and drawbacks, allowing trust to be supported in a wide variety of environments.

- The **pairwise trust model** is the simplest of all trust architectures, but the least scalable. In the pair wise architecture, each Web service is provided—at configuration—the security information of all other Web services that will be interacted with so that those transactions and Web services can be trusted.
- In the **brokered trust model**, an independent third party acts as a trusted third party (TTP) for the Web service. The requester and provider interface with the third party for a variety of security services. Unlike the pair wise trust model, Web services using the brokered trust model need to be designed with the broker's interface in mind, so that identity information can be properly retrieved by the Web service.

6. Policy Framework - WS-Policy represents a set of specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries and end points (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points. Application and domain specific policies need to be designed.

7. The documents representing patient's medical history need to be encrypted or signed appropriately.

8. Establishing a secure communication channel is necessary. [24]Secure Web communication protocols provide a way to authenticate clients and servers on the Web and to protect the confidentiality of communication between clients and servers. A variety of secure communication standards that use public key technology have been developed, including Secure Hypertext Transfer Protocol (SHTTP), IP Security (IPSec), PPTP, and L2TP. The leading general-purpose, secure Web communication protocols are SSL 3.0 and the open TLS protocol that is based on SSL. The SSL and TLS protocols are widely used to provide secure channels for confidential TCP/IP communication on the Web.

9. Web Services Security: Non–Repudiation This specification extends the use of XML Digital Signature in the context of WSS: SOAP Message Security to allow senders of SOAP messages to request message disposition notifications that may optionally be signed to prove that the receiver received the SOAP message without modification. The specification also defines a method for embedding SOAP message dispositions in a SOAP message header. This specification constitutes a protocol for voluntary non-repudiation of receipt that when used systematically provides cryptographic proof of both parties participation in a transaction. This

specification does not define any mechanism to prove receipt of a message by a non-conformant implementation.

The formal specification of the procedure is presented as a VDM++ snippet herein generated using Overture.

```
service.vdmpp

class service
types
id_management =  <Isolated> | <Federated> |   <Centralized> | <nil>;
securing_credentials = <HTTPS> | <xml_enc_or_dsig> | <WS_Security> | <other> | <nil>;
session_management =<authenticated_with_deidentification> | <nil> | <othercredentials> ;
repository_acl = <access> | <deny> | <nil> ;
trust_management = <pairwise> | <Brokered> | <nil>;
Security_Policy = <rolebasedaccess> | <encryption> | <authentication> | <nil>;
encrypt_document = <yes> | <no> ;
secure_comm_channel = <yes> | <no> ;
ws_non_repudiation = <wsrn> | <others> | <nil>;
endstatus = <true> | <false> ;

instance variables
public pid : map service to set of PatientInfo;
identity_scheme : id_management := <Isolated>;
securing_scheme : securing_credentials :=  <WS_Security>;
repoacl_scheme : repository_acl := <deny>;
trust_scheme : trust_management := <pairwise>;
session_scheme : session_management := <authenticated_with_deidentification> ;
policy_choice : Security_Policy := <authentication>;
encryption : encrypt_document := <yes>;
comm_channel : secure_comm_channel := <yes>;
repudiation_scheme : ws_non_repudiation := <wsrn>;
status : endstatus := <false>;
operations
functions
request(service: service , PatientInfo: PatientInfo) status:endstatus
pre service.pid = PatientInfo.pid
post

if securecredentials(service) = nil then
(
if manageid(service) = nil  then
(
if managesession(service) = nil then
(
if aclauthorization(service) = nil then
(
if managetrust(service) = nil then
(
if policyapplication(service) = nil then
(
if encryptdoc(service) = nil then
(
if securecommchannel(service) = nil then
(
if addnonrepudiation(service) = nil then status = <true>  else status = <false>
)  else status = <false>
```

```
) else status = <false>
) else status = <false>
) else status = <false>
) else status = <false>
) else status = <false>
) else status = <false>
) else status = <false> ;

manageid(service: service) status:endstatus
pre nil
post if service.identity_scheme = <nil> then status = <false> else status = <true> ;
securecredentials(service: service) status:endstatus
pre nil
post if service.securing_scheme = <nil> then status = <false> else status = <true> ;
aclauthorization(service: service) status:endstatus
pre nil
post if service.repoacl_scheme = <nil> then status = <false> else status = <true> ;
managetrust(service: service) status:endstatus
pre nil
post if service.trust_scheme = <nil> then status = <false> else status = <true> ;
policyapplication(service: service) status:endstatus
pre nil
post if service.policy_choice = <nil> then status = <false> else status = <true> ;
encryptdoc(service: service) status:endstatus
pre nil
post if service.encryption = <no> then status = <false> else status = <true> ;
securecommchannel(service: service) status:endstatus
pre nil
post if service.comm_channel = <no> then status = <false> else status = <true> ;
addnonrepudiation(service: service) status:endstatus
pre nil
post if service.repudiation_scheme = <nil> then status = <false> else status = <true> ;
managesession(service: service) status:endstatus
pre nil
post if service.session_scheme = <nil> then status = <false> else status = <true> ;
end service
```

Figure 2 Specification of service class which has the pid mapped with pid of PatientInfo class

```
PatientInfo.vdmpp

class PatientInfo
types
string = seq of char;
details = seq of char;
values
instance variables
username: string := [ ];
password: string := [ ];
public pid : map service to set of PatientInfo;
detail: details := [ ];
operations
functions
sync
--thread
Traces
end PatientInfo
```

Figure 3 . Specification of PatientInfo class

The specification stated in VDM++ in Figure 2 and Figure 3 makes sure all the attributes of security like authentication, confidentiality, integrity and non repudiation are met by the service and returns true if not. There are two classes namely service and PatientInfo which are mapped to each other on a one – to – one basis with the help of pid attribute.

Using the WS-Security Specification presented here, service end-points have a standard means for securing SOAP messages using XML Signature and XML Encryption. In this paper, in addition to usage of WS – Security for securing messages, a technique for negotiating a mutually-acceptable security policy based on WSDL is proposed. The Medi – Helper discussed in session III shows a secure architecture for transacting healthcare information over the Internet.

3. INTEGRATED SECURE WEB SERVICE RELIABLE MEDIHELPER ARCHITECTURE

In this section we propose how to implement the security mechanisms and integrate the security framework into Web services in order to make Web services robust against the attacks. This framework shown in Figure 4 consists of three layer architecture. They are legacy layer, Integrated service layer and application layer. Legacy Layer consists of Server management system and server(s) for data storage and manipulation. These are updated to the Log Server in a standard format. It plays a vital role in making the medical history of a Patient available anywhere anytime. The role of infrastructure services renders the services for Patients like X-Ray, ECG, and ICU etc. The data obtained is transferred to the integrated services layer for creating WS Policy. Wherein, the data is synchronized by the data source. MIS component manipulates, filters the data over the data source and provides a view of the medical history of user to the Doctor. It hides the underlying complexity attributed by the Legacy Layer and provides an integrated view of the data.

The WS-Security and integrated security services that come along with METRO stack are made available to the application. The Application Adapter accesses and updates the data source for user oriented information, with configuration details on WSDL and generates the SOAP messages for the application. This Layer depicts the practical aspect of web service deployment where the messages are based on SOAP technologies, which is definitely not capable of replacing HTTP, because of its wide acceptability and usage. The Integrated Services Layer provides secure and reliable transactions with the help of WS-IT Stack. Metadata specifications describe the structure of messages that can be sent. So it is good to extend the existing framework with SOAP messages over HTTP and reliable messaging using METRO STACK.

The METRO is the Middleware that offers the underlying technologies like WS-POLICY and WS-SECURITY. Medi – Helper uses the non–profit HL7[6] effort for healthcare systems, to manipulate to the full extent the capability of XML for a standard globally accepted messaging syntax and document structure. The security can be provided by selective policy assertion and WS – Security.The policy assertion which is used in the sheltered medi-helper is exposed in Figure 3. It identifies a behavior that is a requirement of a policy subject. Satisfying assertions in the policy usually results in behavior that reflects these conditions. A policy assertion is supported by a requester if and only if the requester satisfies the requirement. In the Figure 3, the policy is previously agreed upon by the participants. It is proclaimed by the Provider. The Consumer should supply the parameters demanded by the operating security policy, crafted by the Provider. If the parameters satisfy the conditions then the Consumer is allowed to access the resources or services. If the parameters don't satisfy the conditions, the Consumer is denied access.

Web services are being successfully used for interoperable solutions across various industries. One of the key reasons for interest and investment in Web services is that they are well-suited to enable service-oriented systems.

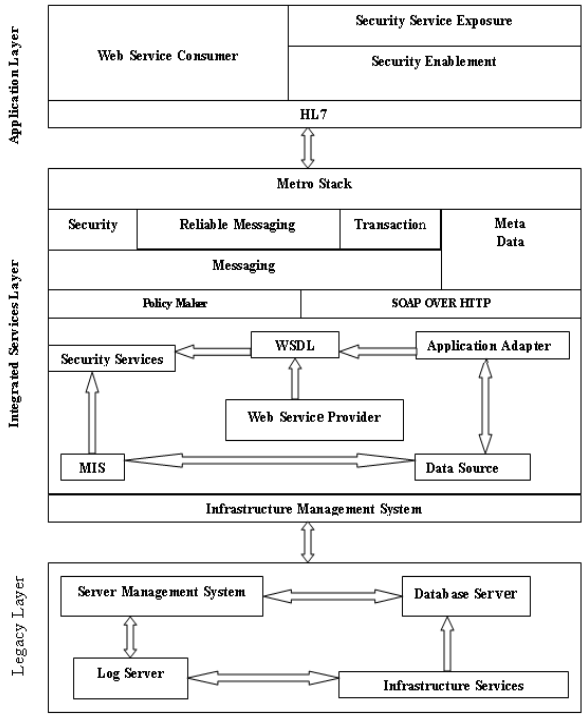

Figure 4 Integrated Secure Web Service Architecture

3.1Username Authentication with Symmetric Key mechanism

The Medi - Helper uses “Username Authentication with Symmetric Key mechanism”. The Username Authentication with Symmetric Key mechanism protects your application for integrity and confidentiality. Symmetric key cryptography relies on a single, shared secret key that is used to both sign and encrypt a message. Symmetric keys are usually faster than public key cryptography. For this mechanism, the client does not possess any certificate/key of his own, but instead sends its username/password for authentication. The client shares a secret key with the server. The shared, symmetric key is generated at runtime and encrypted using the service's certificate. The client must specify the alias in the trust store by identifying the server's certificate alias. Using the existing AES encryption algorithm, mixing of data re-encryption is done. The following a code snippet from the security parameter configuration files used in the application.

<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />

3.2 Digital Signatures

XML signatures are digital signatures designed for use in XML transactions. The standard defines a schema for capturing the result of a digital signature operation applied to arbitrary XML data. XML signatures add authentication, data integrity, and support for non-repudiation to the data that they sign. However, unlike non-XML digital signature standards [3], XML signature has been designed to both account for and take advantage of the Internet and XML. The Figure 4 shows the structure of elements in Xml Digital Signatures.

3.3 WS-Security

IBM and Microsoft have begun a joint initiative to define an architecture and roadmap to address gaps between existing security standards and Web Services and SOAP. The Medi – Helper uses Binary Security Token with X.509 Certificates. A security token[18] asserts claims and can be used to assert the binding between authentication secrets or keys and security identities.WS-Security handles credential management in two ways. It defines a special element, UsernameToken, to pass the username and password if the Web service is using custom authentication. WS-Security also provides a place to provide binary authentication tokens such as Kerberos Tickets and X.509[18,15] Certifications: BinarySecurityToken. The Security Token service might be Kerberos, PKI, or a username/password validation service. When using X.509 certificates, the message can be signed using the private key. The message should contain the certificate in a BinarySecurityToken. When using X.509, anyone who knows the X.509 public key can verify the signature.

3.4 WS-Reliability

WS-Reliability is a SOAP-based specification that fulfills reliable messaging requirements critical to some applications of Web Services. SOAP over HTTP is not sufficient when an application-level messaging protocol must also guarantee some level of reliability and security. This specification defines reliability in the context of current Web Services standards. This specification has been designed for use in combination with other complementary protocols and builds on previous experiences e.g., ebXML. Reliable messaging requires the definition and enforcement of contracts between: 1)The Sending and Receiving message processors (contracts about the wire protocol) 2)The messaging service provider and the users of the messaging service (contracts about quality of service).

3.5 WS-Policy

WS-Policy provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of entities in an XML Web services-based system[8,15]. WS-Policy defines a framework and a model for the expression of these properties as policies. The Medi - Helper uses the authentication oriented policies. The policies used by client and server are shown in Figure 5 and Figure 6.
SecureWebServiceService.xml

```
<wsp:Policy wsu:Id="SecureWebServicePortBindingPolicy">
<wsp:ExactlyOne>
<wsp:All>
<sc:KeyStore wspp:visibility="private" storepass="changeit" type="JKS" location="client-keystore.jks"
alias="xws-security-client"/>
<sc:TrustStore wspp:visibility="private" storepass="changeit" type="JKS" location="client-truststore.jks"
peeralias="xws-security-server"/>
<sc:CallbackHandlerConfiguration wspp:visibility="private">
<sc:CallbackHandler default="12345" name="usernameHandler"/>
<sc:CallbackHandler default="54321" name="passwordHandler"/>
</sc:CallbackHandlerConfiguration>
</wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
<wsp:Policy  xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200702"
wsu:Id="tryWSPortBindingPolicy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" >
<wsrmp:RMAssertion>
<wsp:Policy>
<wsrmp:DeliveryAssurance>
<wsp:Policy>
<wsrmp:InOrder />
</wsp:Policy>
</wsrmp:DeliveryAssurance>
</wsp:Policy>
</wsrmp:RMAssertion>
<wsam:Addressing />
</wsp:Policy>
```

Fig 5. Client File Showing Ws-Policy Code For Carrying Username And Password

```
<wsp:Policy>
<sp:ProtectionToken>
<wsp:Policy>
<sp:X509Token
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/
IncludeToken/Never">
<wsp:Policy>
<sp:WssX509V3Token10/>
</wsp:Policy>
</sp:X509Token>
</wsp:Policy>
</sp:ProtectionToken>
<sp:Layout>
<wsp:Policy>
<sp:Strict/>
</wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp/>
<sp:OnlySignEntireHeadersAndBody/>
<sp:AlgorithmSuite>
<wsp:Policy>
<sp:Basic128/>
</wsp:Policy>
</sp:AlgorithmSuite>
</wsp:Policy>
```

```
……
<wsp:Policy wsu:Id="MediHelperPortBindingPolicy">
<wsrmp:RMAssertion>
<wsp:Policy/>
</wsrmp:RMAssertion>
<wsam:Addressing/>
</wsp:Policy>
……
<sp:EncryptedParts>
<sp:Body/>
</sp:EncryptedParts>
```

Figure.6 Server Side File Showing Policy To Accept Binary Security Tokens And Encrypting Body Segment

3.6 VDM++

The [21] VDM++ specification is written using OvertureIde. Models in VDM are formal in the sense that they have a very precisely described semantics, making it possible to analyze models in order to confirm or refute claims about them. Such an analysis often reveals gaps in the developer's and the client understanding of the system, allowing these to be resolved before an expensive commitment is made to program code. The Vienna Development Method (VDM) is one of the longest-established Formal Methods for the development of computer-based systems. Originating in work done at IBM's Vienna Laboratory in the 1970s, it has grown to include a group of techniques and tools based on a formal specification language - the VDM Specification Language (VDM-SL). It has an extended form, VDM++ which supports the modeling of object-oriented and concurrent systems. Support for VDM includes commercial and academic tools for analyzing models, including support for testing and proving properties of models and generating program code from validated VDM models.

4 RESULTS AND DISCUSSIONS

In Medi - Helper the web service messages are subjected to the policy check and then are allowed to access the actual web services. The messages are encrypted for Security purpose. The web service client enters the username and password and passes the details of personal identification, disease indication and remedial treatments undergone. The details are not visible to the onlooker of the SOAP messages[18] since they are encrypted as shown in Figure 7 and Figure 8 They are only visible to the server as shown in Figure 8, as printed on the server console. The WS-POLICY code for the server and client are shown in Figure 5 and Figure 6. The client embeds the username and password information on the SecureWebServiceService.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Header>
<To xmlns="http://www.w3.org/2005/08/addressing">http://localhost:8080/RWS/RWSService</To>
<Action xmlns="http://www.w3.org/2005/08/addressing">http://RWS/RWS/addRequest</Action>
<ReplyTo xmlns="http://www.w3.org/2005/08/addressing">
<Address>http://www.w3.org/2005/08/addressing/anonymous</Address>
</ReplyTo>
<MessageID xmlns="http://www.w3.org/2005/08/addressing">uuid:5a6b1c6a-d055-4157-9598-
7beb67953183</MessageID>
<ns2:Sequence
xmlns="http://www.w3.org/2005/08/addressing"
xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrm/200702"
xmlns:ns3="http://docs.oasis-open.org/ws-rx/wsmc/200702"
xmlns:ns4="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns5="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:ns6="http://schemas.microsoft.com/ws/2006/05/rm"
xmlns:ns7="http://schemas.xmlsoap.org/soap/envelope/" ns7:mustUnderstand="true">
<ns2:Identifier>uuid:774a96eb-9c14-4f02-b50c-9543c81fabf7</ns2:Identifier>
<ns2:MessageNumber>1</ns2:MessageNumber>
</ns2:Sequence>
<ns2:AckRequested
xmlns="http://www.w3.org/2005/08/addressing"
xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrm/200702"
xmlns:ns3="http://docs.oasis-open.org/ws-rx/wsmc/200702"
xmlns:ns4="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns5="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:ns6="http://schemas.microsoft.com/ws/2006/05/rm"
xmlns:ns7="http://schemas.xmlsoap.org/soap/envelope/" ns7:mustUnderstand="true">
<ns2:Identifier>uuid:774a96eb-9c14-4f02-b50c-9543c81fabf7</ns2:Identifier>
</ns2:AckRequested>
</S:Header>
```

Figure 7 Request - SOAP Body

```
<S:Body wsu:Id="_5006">
<xenc:EncryptedData
xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns
:ns16="http://www.w3.org/2003/05/soap-envelope"
Type="http://www.w3.org/2001/04/
xmlenc#Content" Id="_5007">
<xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
/>
<ds:KeyInfo xmlns:xsi="http://www.w3.org/2001/XMLSch
ema-instance"
xsi:type="keyInfo"><wsse:SecurityTokenReference>
<wsse:KeyIdentifier message-security-1.1#EncryptedKeySHA1" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200 401-wss-soap-message-security-1.0#Base64Binary">
pwUAg2MQMEmIsXrUHUQDAs1YC04=
</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
```

```
</ds:KeyInfo>
<xenc:CipherData>
<xenc:CipherValue>
0qz5nhr4RQ5ITUWzwxLK1QPF7YzqLTeH4O6eGyHCDGgl
4wXiuNpee93DucAu35uroIlXGSov+Xu6HfBb3LcRZ02e85e8
Gzj+XHX98muaudqGKFkVGCVikez0sVqdE3kEsQlAhsjYxF
Bken5g2O6qpo7jfhhv7abiWw5zaCK+ZUs=
</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</S:Body>
```

Figure 8 Response - SOAP Body

5 PERFORMANCE ANALYSIS

The graph shown in Figure 10 shows the benefits of proposed security framework in terms of securing the web services against several attacks. It clearly states that, the proposed approach is providing good security by making use of METRO – STACK and the HL7[6] document structures meant for critical services like healthcare. The table 1 shows that the Medi – Helper uses the technologies of XML Encryption, Signatures and WS-Security Tokens and HTTP Authentication and its comparative study. Together they prevent against almost all attacks except Denial of Service

	Message Alteration	Loss of Confidentiality	Falsified Message	Man in Middle	Principal Spoofing	Forged Claims	Replay of Message Parts	Replay of Message	Denial Of Service
XML Encryption		x		x	x	x	x		
XML Signature	x		x		x	x	x	x	
WS-Security Tokens			x		x	x			
WS-Addressing								x	
SSL/TLS									
SSL/TLS with client certificates									
HTTP Authentication			x		x	x			

TABLE 1- COMPARATIVE STUDY

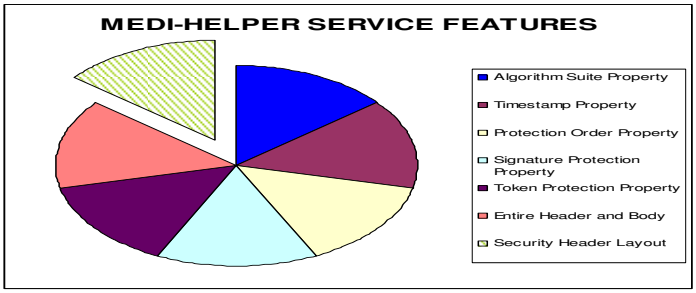

Figure 9. Performance Analysis

6 CONCLUSION AND FUTURE WORK

The Web Service Based Secure Medical Assistant serves as a platform for the transfer of medical documents and also ensuring confidentiality and integrity of the same data in conformance to the Security Framework. The framework is a proposal for optimal security and reliability, which may be suitable generally for any domain. It is also scalable to use any web service related technology for acquiring features like QOS and Security. The Secure Medical Assistant can be used with standards like HL7 which are created to depict medical information in terms of XML so that the documents can be interchanged in a standard manner. The service can be implemented with Third Party Authentication mechanisms like Kerberos so that we can manage large number of patient's details and in a secure manner. By incorporating the proposed technology with UDDI and encapsulating security layer the Medi-Helper can become a universally available security solution.

REFERENCES

[1] Institute of Medicine, Crossing the Quality Chasm: A New Health System for the 21st Century. Washington, DC: National Academy Press, 2001.

[2] M. J. Field and K. N. Lohr, Guidelines for Clinical Practice: From Development to Use. Washington, DC: Institute of Medicine, National Academy Press, 1992.

[3] J. Boyer et al., Exclusive Canonicalization Version 1.0, 18 January 2002, World Wide Web Consortium, http://www.w3.org/TR/xml-exc-c14n/.

[4] R. N. Shiffman, Y. Liaw, C. A. Brandt, and G. J. Corb., "Computer-based guideline implementation systems: A systematic review of functionality and effectiveness," J. Amer. Med. Informat. Assoc., vol. 6, no. 2, pp. 104– 114, Mar./Apr. 1999.

[5] M. Entwistle and R. N. Shiffman, "Turning guidelines into practice: Making it happen with standards—Part," in Healthcare and Informatics Review Online. Auckland, New Zealand: Enigma, Mar. 2005.

[6] www.hl7.org/implement/standards/index.cfm

[7] A. Seyfang, S. Miksch, and M. Marcos, "Combining diagnosis and treatment using Asbru," Int. J. Med. Informat., vol. 68, no. 1–3, pp. 49–57, 2002.

[8] M. Peleg, O. Ogunyemi, and S. Tu, "Using features of Arden syntax with object-oriented medical data models for guideline modeling," in Proc. AMIA Symp., 2001, pp. 523–527.

[9] P. Ciccarese, E. Caffi, L. Boiocchi, S. Quaglini, and M. Stefanelli, "A guideline management system," in Proc. MedInfo 2004, pp. 28–32.

[10] Joch, A., "Heads Above the Crowd." Healthcare Informatics, Volume 18, Number 1, 2001, 27-32.

[11] Stein, M., "Medical Education and the Internet: This Changes Everything." JAMA, Volume 285, Number 6, 2001, 809.

[12] E. Ferrari and B. Thuraisingham, "Security and Privacy for Web Databases and Services," E. 2004 and L. 2992, Eds. Berlin Heidelberg 2004: Springer-Verlag, 2004, pp. 17-28.

[13] Evenhaim, A., "Taking e-Health Relationship Management into the next Millennium." Medical Marketing and Media, Volume 36, Number 2, 2001, 104-110.

[14] Bachar Alrouh and Gheorghita Ghinea A Performance Evaluation of Security Mechanisms for Web services Fifth International Conference on Information Assurance and Security,2009

[15] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris and D. Orchard. (2004, Feb.). Web services architecture. W3C, http://www.w3.org/TR/ws-arch/.

[16] Improving Web Application Security: Threats and Countermeasures on MSDN: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html /ThreatCounter.asp.

[17] http://www.ibm.com/developerworks/library/ws-secure/

[18] http://msdn.microsoft.com/en-us/library/ms788756.aspx

[19] http://msdn.microsoft.com/en-us/library/ms977327.aspx

[20] Anoop Singhal, Theodore Winograd, Karen Scarfone "Guide to Secure Web Services" NIST

[21] Peter Gorm Larsen "Tutorial for Overture/VDM-SL - Overture – Open-source Tools for Formal Modelling TR-2010-01March 2010"

Authors

Ms. Priya Loganathan is a M.Tech graduate from Madras Institute of Technology, India. The author Specializes in Data Structures, Image Processing. The author is pursuing projects in Machine Vision.

Ms.Jeyalakshmi Jeyabalan is a M.Tech graduate from Sathyabama University, India. The author specializes in Web Services, Operating Systems.

Ms.Usha Sarangapani is a M.Tech graduate from Sathyabama University, India. The author specializes in Web Services, Object Oriented Programming.