A Framework for Securing Web Services by Formulating an Collaborative Security Standard among Prevailing WS-* Security Standards

Conference Paper *in* Communications in Computer and Information Science · July 2011

DOI: 10.1007/978-3-642-22726-4_29 · Source: DBLP

CITATIONS
13

READS
265

4 authors:

Priyadharshini Muthukrishnan
VIT-AP University
15 PUBLICATIONS **193** CITATIONS
SEE PROFILE

Baskaran Ramachandran
Anna University, Chennai
155 PUBLICATIONS **2,234** CITATIONS
SEE PROFILE

Madhan Kumar Srinivasan
Wise Work
46 PUBLICATIONS **278** CITATIONS
SEE PROFILE

Paul Rodrigues
King Khalid University
108 PUBLICATIONS **879** CITATIONS
SEE PROFILE

A Framework for Securing Web Services by Formulating an Collaborative Security Standard among Prevailing WS-* Security Standards

M. Priyadharshini[1], R. Baskaran[2], Madhan Kumar Srinivasan[3], and Paul Rodrigues[4]

[1,2] Computer Science Department, Anna University, Chennai, India
mpriya1977@gmail.com, baaski@annauniv.edu
[3] Education & Research, Infosys Technologies, Mysore, India
madhan_srinivasan@infosys.com
[4] Department of IT, Hindustan University, Chennai, India
deanit@hindustanuniv.ac.in

Abstract. Web Services enables communication between applications with a less working out on the underlying mechanics of communication. This paper provides a brief introduction to security concepts and describes in detail various specifications related to security among WS-* family and association among those specifications. Web Service Standards available do not completely address security for web services. In this paper we have proposed a framework that consists of components which could secure web service interactions facilitating interoperability between various WS-* security standards, by devising a collaborative security standard based on the associability of WS-* security standards and can be furthermore customized by optimizing the selection and projection functions of standard list and parameter list. The parameter list is again formulated by clear understanding of association of the WS-* security standards.

Keywords: WS-* family, collaborative security standard, interoperability, web services.

1 Introduction

Today's enterprises take advantage of the benefits of loosely coupled web services and made it an integral part of their business process. Therefore, need for security in business process raises the level of security needs in web services as well. The loose coupling is possible in web services due to extensive usage of XML (Extensible Mark-up Language). XML is used in web services for describing, requesting, responding and so on, which drives us to secure XML messages if web services need to be secured. The chapter just following briefs about the Web Service Model, Chapter III about the various security issues need to be addressed in web services and Chapter IV describes about formulation of collaborative security standard and proposed framework which provides an interoperable and secure gateway for web service usage. Chapter V briefs about various WS-* security standards along with the

A. Abraham et al. (Eds.): ACC 2011, Part IV, CCIS 193, pp. 269–283, 2011.

issues addressed by those specifications and followed by Chapter VI about the associations that exist between the standards, which serves as the basis for formulating the collaborative security standard. In Chapter VII selection criteria based on scenarios is presented with few scenarios and finally Chapter VIII gives how evaluation process can be done for evaluating the security provision of the framework

2 Web Service Model

Web service model is one of the approaches for building SOA (Service Oriented Architecture). Service provider creates a web service and its service definition and publishes in the service registry. Service Requestor finds the service in the registry and obtains the WSDL description and URL to the service itself. The service requestor with the help of information obtained binds to the service and invoke it. Figure 1 shows the web services model as interaction between service requestor and service provider through UDDI registry which is same as that of the Service Oriented Architecture.

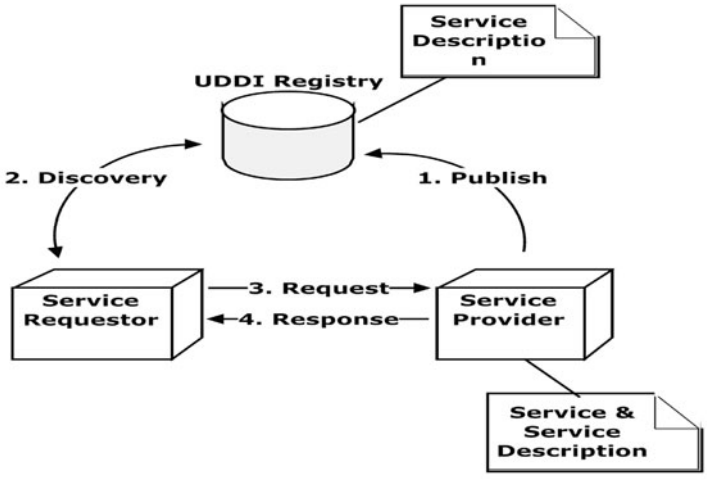

Fig. 1. Web Services Model

The core technologies which form the foundation of Web services are SOAP, WSDL and UDDI.

2.1 SOAP

Simple Object Access Protocol (SOAP) is used as a standard to exchange messages between client applications and services that run on server through Internet infrastructure. The method invocation is made as a SOAP request and result is passed as SOAP response. SOAP message are in form of XML and it encapsulates <Soap:Header> as optional element and <Soap:Body> as mandatory element inside a <Soap:Envelope>[1]. Soap Header holds the information needed by the SOAP node

to process the SOAP message such as authentication, routing etc. Soap body contains the information to be sent to the SOAP message receiver. The format of SOAP request and response will be as follows [7]:

Table 1. SOAP Request invokes OrdItem() method from http://www.Tanishq.com/Order and SOAP Response passes order number generated on processing the order to the client

SOAP Request to Process Order	SOAP Response on Processing Order
<Soap:Envelope xmlns:Soap-ENV= "http://schemas.xmlsoap.org/soap/envelope/" Soap:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/"> <Soap:Body> <Ord:OrdItem xmlns:Ord="urn:Order"> <CID>70010</CID> <ItNum>105057</ItNum> <ItNme>WGRWRD</ItNme> <ItDesc>WhiteGoldRing WithRoundDiamond</ItDesc> <ItPrice>8332</ItPrice> <OrdDateTime>2010-02-10 0:10:56</OrdDateTime> </Ord:OrdItem> </Soap:Body> </Soap:Envelope>	<Soap:Envelope xmlns:Soap= "http://schemas.xmlsoap.org/soap/envelope/" Soap:encodingStyle= “http://schemas.xmlsoap.org/soap/encoding/”> <Soap:Body> <Ord:OrdItemResponse xmlns:Ord ="urn:Order"> <OrdNum>20014</OrdNum> </Ord:OrdItemResponse> </Soap:Body> </Soap:Envelope>

Table 2. Sample WSDL for Placing Order is specified

OrderItem.WSDL
<?xml version="1.0" encoding="UTF-8"?> <definitions name="OrdService" targetNamespace="urn:Order" xmlns:tns="urn:Order" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"> <message name="OrdItem"> <part name="CID" type="xsd:int"/> <part name="ItNum" type="xsd:int"/> <part name="ItNme" type="xsd:string"/> <part name="ItDesc" type="xsd:string"/> <part name="ItPrice" type="xsd:double"/> <part name="OrdDateTime" type="xsd:string"/> </message> <message name="OrdItemResponse"> <part name="OrdNum" type="xsd:int"/> </message> <portType name="OrdItemPort"> <operation name="OrdItem" parameterOrder="CID ItNum ItmName ItmDesc ItPrice OrdDateTime"> <input message="tns:OrderItem"/> <output message="tns:OrdItemResponse"/> </operation> </portType> <binding name="OrdItemBinding" type="tns:OrdItemPort"> <soap:binding transport=”http://schemas.xmlsoap.org/soap/http” style="rpc"/> <operation name="OrdItem">

Table 2. *(Continued)*

```
      <soap:operation soapAction=""/>
      <input>
        <soap:body  encodingStyle=”http://schemas.xmlsoap.org/soap/encoding/”
                    use="encoded" namespace="urn:Order"/>
      </input>
      <output>
        <soap:body  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                    use="encoded" namespace="urn:Order"/>
      </output>
     </operation>
   </binding>
   <service name="OrderService">
    <port name="Order" binding="tns:OrdItemBinding">
     <soap:address location="http://www.Tanishq.com/Order"/>
    </port>
    </service>
</definitions>
```

2.2 WSDL

WSDL is an XML document which is the web service interface published by the service providers. Service requestors who wish to access the service can read and interpret the WSDL file. The information in the WSDL file is as follows:

- Location of the service
- Operations performed by the service
- Communication Protocol supported by the service
- Message Format for sending and receiving request and responses.

2.3 UDDI

UDDI (Universal Description, Discovery and Integration) is the directory which has list of web service interfaces provided by various businesses. The interfaces are represented using WSDL which is rendered when businesses find the interfaces suitable for their search. UDDI are public or private platform-independent framework driven by service providers like Dell, IBM, Microsoft, Oracle, SAP, and Sun as well as few e-business leaders.

Web service is a powerful technology for distributed application development and integration. Today’s most e-commerce applications are based on the involvement of web services and hence make web service as an essential element in current scenario. Next chapter elaborates security issues in web services.

3 Security Issues

As stated earlier Web Services rely on Internet infrastructure and hence the security issues encountered in network is encountered in web services also.

3.1 Confidentiality

Confidentiality specifies that the content of the message should be accessed only by the sender and receiver. This is achieved by appropriate encryption and decryption

algorithms applied on entire message or parts of the messages. SSL using HTTPS can provide point-to-point data privacy i.e. security at transport level. At application level sensitive data fields can be applied with encryption mechanisms. Sniffing or eaves dropping is an attack with respect to confidentiality.

3.2 Authentication

Authentication is establishment of proof of identities among entities involved in the system. Username and password are used for authenticating the user at platform level. At message level to provide authenticity SOAP headers [5] is added with user name and password, assigned tickets and certificates such as Kerberos and X.509 certificate. In application level custom methods can be included for authentication. Single Sign on or Trust relationship need to be incorporated in routing to provide authentication between multiple services.

3.3 Authorization

One entity may be authorised to do certain operations, and access certain information whereas others may not be. In Web services access control mechanisms need to be provided in form of XML (XACML and SAML). Access control may be based on Role (RBAC), Context (CBAC), Policy (PBAC), Attribute (ABAC) and so on[2].

3.4 Non-Repudiation

Non-Repudiation is disclaiming the message sending or receiving, time of sending and receiving the message. On critical and secure service access non-repudiation is one of the major issues. A Central arbiter Trusted Third Party (TTP) [1] should be introduced along with XML Signature to provide security in these cases.

3.5 Availability

Authorized resources and services available at all times are meant by availability. Denial of Service (DOS) is the commonly encountered problem related to availability.

3.6 Integrity

The change of message content during transit leads to loss of Integrity. It is mainly concerned with the web service description (WDSL) file. On tampering and changing this file, intended service may not get bind to the requestor and even problems may arise in case of composition. Proper Hashing algorithm or XML Signature may overcome this issue.

4 Proposed Security Framework Including Formulation of Collaborative Standards

4.1 Security Framework

The Proposed Security Framework consists of components such as Security Manager, Static Analyser and Dynamic Analyser. Figure 2 depicts the Security Framework

which serves as a gateway to ensure the security of the web service access from various distributed client applications. Web Service model involves the process of publishing and invoking services. Proposed Security Framework includes security list formation and corresponding parameter list which is devised as a collaborative security standard.

Static Analyser is the component which is invoked during registering of service, which will guide the service provider or the publisher to customise and hence record the security standard values for the Standard List as well as corresponding Parameter List.

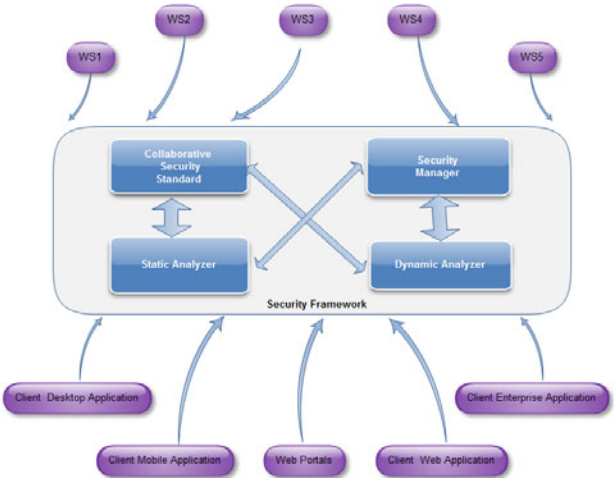

Fig. 2. Security Framework

Dynamic Analyser component is invoked during discovery and execution of the service, checks for the correctness of the security needs specified in the standard at various levels such as message and transport level.

Security Manager is the component in the framework which manages the process of proper execution of the framework maintaining the logs made during Static and Dynamic Analysis.

$$SM=\{<SL,PL(SL),S_{log}> \mid af(<SL,PL(SL),S_{log}>) \ \} \quad (1)$$

Where

SL – Standard List
PL (SL) – Parameter List of Standard List Item
S_{log} – Service Log
af – registering or access function

4.2 Collaborative Security Standard

Collaborative security standard consist of Standard List and Parameter List. A Standard list is selected based on the precise understanding of the security needs and the WS-* Security Standards and their associations, which could address the needs.

Standard List (SL) formulated with WS-*(all XML Security standards) pertaining to security as input:

$$SL = \{I \mid I \in WS\}$$
$$WS = \{t \mid t \in sf(WS\text{-}*)\} \quad (2)$$

Where

SL – Standard List

I - Standard List Item

sf – selection function selecting among WS standard Items with security as objective

Parameter List (PL) for each Standard List (SL) which are found to be suitable for inclusion:

$$PL = \{P \mid P \in pf(SL)\} \quad (3)$$

Where

pf – Projection function to list out only mutually exclusive parameters

5 WS-* Security Standards

Organization for the Advancement of Structured Information Standards (OASIS) and World Wide Web Consortium (W3C) devised many WS standards which were used for providing security, reliability, and transaction abilities in web services. These WS-* standard specifications help to enhance the interoperability between different industry platforms, most notably Microsoft's .NET platform and IBM's WebSphere software. This chapter will discuss on the standards which concentrate on Security.

5.1 WS-Policy

WS-Policy [9] is general-purpose framework model that describes web service related policies. A policy can specify properties, requirements and capabilities of the web services. While service request is been sent this policy is used to validate and accept the request. For example a Policy can mandate the web service for NEFT (Non-Electronic fund Transfer) to provide service between 10:00 AM and 5:00 PM from Monday to Friday and between 10:00 AM and 2:00 PM on Saturdays or that request should be signed using X.509.

The Policies defined in WS-Policy can be attached to service endpoints or XML data using WS-PolicyAttachment.

The Polices can be retrieved from SOAP node using WS-MetadataExchange.

Specific Policy assertion related to text encoding, SOAP Protocol version and Predicates that enforce the header combinations existing between SOAP messages are defined using WS-PolicyAssertions. The Proposed Security Framework consists of components such as Security Manager, Static Analyser and Dynamic Analyser. Figure 2 depicts the Security Framework which serves as a gateway to ensure the security of the web service access from various distributed client applications. Web Service model involves the process of publishing and invoking services. Proposed

Security Framework includes security list formation and corresponding parameter list which is devised as a collaborative security standard.

5.2 WS-SecurityPolicy

WS-SecurityPolicy[8] consists of the security related assertions such as 'Security Token' which tells the requestor which security token need to be used while calling a given web service. The other assertions include assertions specifying about Integrity, Confidentiality, and Visibility which are used to specify the message part that need to be protected and that parts need to remain unencrypted. Message expiry can be prompted using 'Message Age exception'. For Instance, XPath based SignedElements assertion is used to arbitrary message element that need Integrity protection. The RequiredParts and RequiredElements using QNames and XPath are used to specify the header element the message should contain.

WS-SecurityPolicy also consists of assertions related to cryptographic algorithms, transportation binding and the order of applying cryptographic algorithms.

5.3 WS-Security

WS-Security Standard addresses Confidentiality and Integrity of XML messages transferred as request and responses. The header <wsse:Security>[12] is used to attach security related information. WS-Security standard defines cryptographic processing rules and methods to associate security tokens. Since SOAP messages are processed and modified by SOAP intermediaries, mechanisms such as SSL/TLS are insufficient to provide end-to-end security of SOAP messages and hence WS-Security gain importance.

WS-Security specifies that signature confirmation attribute included to digital signature of request and again back included in the response message, as signed receipt , in order to ensure that the request or response are tied to corresponding response or request.

WS-Security defines a mechanism to associate a security token by including them in the <wsse:Security> header and a reference mechanism to refer the tokens in binary and XML formats. 'Username Token Profile' adds literal plaintext password, hashed password, nonce (time variant parameters), and creation timestamp to already available Username Token. 'Kerberos token profile' defines the way in which Kerberos tickets are embedded into SOAP messages. The Other profiles include 'WS-Security X.509 Certificate token profile', 'SAML token profile' and 'Rights Expression Language Token profile'.

5.4 WS-SecureConversation

WS-Secure Conversation [10] defines way to establish 'security contexts' identified by an URI, which will permit existing SSL/TLS connection to be shared by subsequent requests to a web server in the transport level. When overheads related to key management raises due to introduction of message level security and as a result of which scalability becomes a problem this standard proves to be a better solution.

There are three different ways to establish Security contexts. First, SCT (Security Context Token) retrieval using WS-Trust i.e. SCT is retrieved from a security token

service trusted by the web service. Second, SCT created by the requestor which has a threat of getting rejected by the web service. Third, using security context mutually agreed by requestor as well as provider using challenge-response process. This is SCT is then used to derive the session key, which is used for subsequent encryption and authentication codes. When Security context time exceeds the communication session then it will be cancelled but if it gets expired then it has to be renewed.

5.5 WS-Trust

WS-Trust [11] standard introduces 'Security Token Service' which is a web service that issue, renew and validate security tokens. While multiple trust domains are involved, one security token can be converted into other by brokering trust. When a requestor wants to access a web service and he doesn't hold the right security token specified in the policy. The requestor may state the available token and ask for the needed token to STS else requestor may delegate the responsibility of finding the 'right' token to STS itself and state only available token and just ask for the 'right' token.

When the requestor includes time variant parameters as entropy while requesting for token, STS will return a secret key material which is called proof-of-possession. In this case token may be a certificate whereas the proof-of-possession is the associated private key. Requestor who needs an authorisation token for a colleague which need to be valid only till a particular time period can get a token from WS-Trust.

5.6 WS-Federation

'Federation' means two or more security domains interacting with each other, letting users to access the services from other security domain. Each domain has its own security token service and each of them has their own security policies.

There are few XML standards used along with the WS-* security standards discussed above, which could help those standards in addressing the security issues. They include XMLSignature, XMLEncryption, SAML (Security Assertion Mark-up Language), XACML (Extensible Access Control Mark-up Language) and XKMS (XML Key Management Specification) and so on.

XMLSignature. XMLSignature is the protocol which describes the signing of digital contents as whole or in parts. This provides data integrity and also important for authentication and non-repudiation of web services. This may also be used to maintain integrity and non-repudiation of WSDL files to enable definition of web service to be published and later trusted.

XMLEncryption. XMLEncryption ensures confidentiality and hence provide secure exchange of structured data [3]. XMLEncryption can be applied to parts and even for documents in persistent storage, in contrast to SSL or VPN. Algorithms such as RSA, Triples DES are used for encryption, combination of these algorithms also prove to increase security during message exchange.

SAML. SAML [4] is an XML standard for asserting authentication and authorisation information. Single sign-on (SSO) between different systems and platforms are realised using SAML. SAML does not establish or guarantee the trust between participants instead assumes and requires trust between them. Also SAML does not guarantee confidentiality, integrity or non-reputability of the assertions in transit. This could only be provided by XMLEncryption and XMLSignature or any other mechanisms supported by underlying communication protocol and platform.

XACML. Extensible Access Control Mark-up Language express access control rules and policies used to derive access decision for set of subjects and attributes. In case of multiple rules and policies encoding rules, bundling rules into policies and defining selection and combination algorithms are done by XACML.

Access control list in XACML consists of four tuples:

- Subject – UserIds, groups or role names
- Target Object – single document element
- Permitted action – read, write, execute or delete (not domain specific)
- Provision – execute on rules activation – initiating log-in requesting additional credential etc,

XMLKeyManagementSpecification. XMLKMS is the web service interface which provides public key management environment for usage in XMLSignature and XMLEncryption. It consists of two sub protocols XML key information service specification and XML key registration service specification. The former is used for locating and retrieving public keys from key server. The later defines service interfaces to register to revoke and recover escrowed keys from key server.

So far we had discussed about various WS-* standards that are related to security. Other than these standards we also have standards which ensures factors such as reliability, transaction, routing in web services such as WS-ReliableMessaging, WS-Transaction, WS-Routing, WS-Discovery etc.,

6 Collaboration of WS-* Security Standards

All the above standards discussed do not provide an entire solution on their own but need to be used along with other specification to finally arrive at an end to end security standard. For example WS-Security does not provide session management and that is done by WS-SecureConversation. The security solution can be tailored by the solution providers according to the specific need. In order to tailor the security solution it becomes necessary for the service providers and researchers involved in providing such solution to have a clear insight about the association of these standards in detail, which is as follows.

WS-SecurityPolicy provides assertions specific for security. WS-Policy extends WS-SecurityPolicy provides all generic assertions. Hence WS-Security Policy fits into WS-Policy. The security assertions specified in the WS-Security Policy are utilized by WS-Trust, WS-Security and WS-SecureConversation. Security Assertions are represented using SAML.

WS-Trust utilizes WS-Security for signing and encrypting SOAP Messages with the help of XMLSignature and XMLEncryption[6]. WS-Trust utilizes WS-Policy/WS-SecurityPolicy for expressing Security Token and to determine which particular security token may be consumed by a given web service.

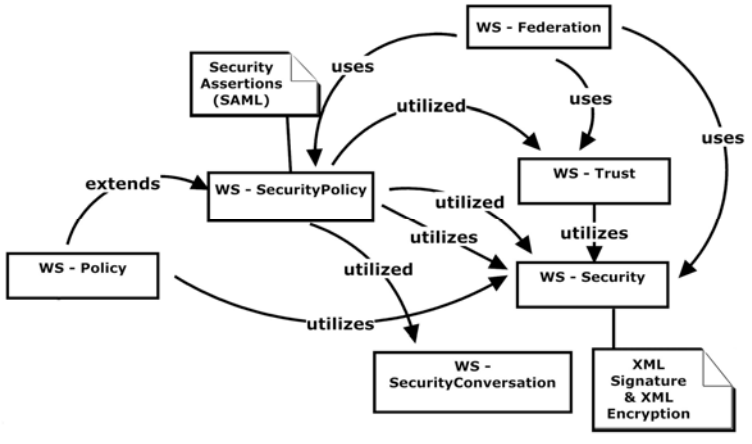

Fig. 3. Collaboration of WS-* Security Standards

Table 3. Summarises the WS-* Security Standards, their purpose and how they collaborate

Standard	Purpose	Related Standards
WS-Policy	Define assertions for web services	WS-SecurityPolicy
WS-SecurityPolicy	Define security Assertions	WS-Trust WS-Federation WS-Security
WS-Security	Provide Message Security	WS-SecureConversation WS-Federation
WS-SecureConversation	Establish security context	WS-Security
WS-Trust	Security Token Management	WS-Security WS-SecuirtyPolicy
WS-Federation	Enable cross domain access	WS-Security WS-SecurityPolicy WS-Trust

WS-Security uses session keys generated by WS-SecureConversation for subsequent encryption and decryption of messages. WS-Federation uses the WS-Security, WS-SecurityPolicy and WS-Trust to specify the scenarios in which the requestors from one domain can get access to services in the other domain.

7 Scenarios of Security Challenges and Technological Solutions

The proposed system provides us with an environment which could handle different scenarios of security challenges in different ways, but in an integrated and exhaustive manner ensuring the whole security challenges as per the requirements.

Table 4. Selection criteria for choosing the WS-* Standards based on security challenges

Security Challenge	Standard List	Parameter List
Confidentiality	WS-Security	XMLEncryption
Authorisation	WS-Trust	SAML Assertion XACML Assertion
Authentication	WS-Security	Username Token Profile Kerberos token profile Certificate token profile SAML token profile Rights Expression Language token profile
Non-repudiation	WS-SecureConversation	STS,X.509,Kerberos
Availability	WS-Security	XMLSignature
Integrity	WS-Security	XMLSignature
	WS-SecurityPolicy	Username Token Profile Kerberos token profile Certificate token profile SAML token profile Rights Expression Language token profile

SCENARIO #1. A Project Proposal is formulated which is aiming at accessing lab results from various laboratories, who are specialist in performing diagnosis for various diseases. This project is intended to be used by all hospital management applications. This could give doctors a clear picture of the status of a patient. In the system requirements, identified significant non-functional requirement is security.

SCENARIO #2. A renowned Bank plans to provide a facility of Tax Payment, which access a Tax calculator service, followed by payment through their payment gateway. The Service implementation needs to be incorporated so as to secure the profile of user details and tax amount as well.

Table 5. We provide a listing for the above said scenarios, which gives the list of Security Objectives, possible Standard List and corresponding Parameter List, which could be the inputs from Static Analyzer to our system and used by Dynamic Analyzer during discovering and binding process

Scenario	Security Challenge	Standard List	Parameter List
# 1: **Provider:** Diagnostic Laboratories **Requestor:** Doctors	Confidentiality Authorisation Authentication Non-repudiation	**SL**= {WS-Security, WS-Trust, WS-SecureConversation } **WS**= {WS-Security, WS-Trust, WS-SecureConversation | sf (WS-Security, WS-Trust, WS-SecureConversation, WS-SecurityPolicy)}	**PL=** { ***pf*** (XMLEncryption, SAML Assertion, XACML Assertion, Username Token Profile ,Kerberos token profile, Certificate token profile ,SAML token profile, Rights Expression Language token profile, STS Token, X.509 Token, Kerberos Token)} **PL**= {XMLEncryption, SAML Assertion, XACML Assertion, (Username Token Profile || Kerberos token profile || Certificate token profile|| SAML token profile || Rights Expression Language token profile), (STS Token || X.509 Token || Kerberos Token)}
#2: **Provider:** Accounting Offices **Requestor:** Bank	Confidentiality Integrity	**SL**= {WS-Security, WS-SecurityPolicy } **WS**= {WS-Security, WS-SecurityPolicy | sf (WS-Security, WS-Trust, WS-SecureConversation, WS-SecurityPolicy)}	**PL=** { ***pf*** (XMLEncryption, XMLSignature, Username Token Profile ,Kerberos token profile, Certificate token profile ,SAML token profile, Rights Expression Language token profile)} **PL**= {XMLEncryption, XMLSignature, (Username Token Profile || Kerberos token profile || Certificate token profile|| SAML token profile || Rights Expression Language token profile)}

8 Evaluation Process

The formulation of collaborative security standard done by the framework can be justified by performing combinations of testing appropriate to the security objectives. Inputs for this testing are taken from S_{log} managed by Security Manager.

$$\text{Security Metric} \quad sm = \sum_{i=1}^{n} \frac{(N_{ai} - N_{fi})}{N_{ai}} \tag{4}$$

Where n is the number of security objectives,
N_{ai} is total number of times client request for the service with security objective i
N_{fi} is number of times program fails to access the service with security objective i

The security metric(sm) when maximum denotes a better security objective achievement. The individual values of N_{ai} and N_{fi} as well as security metric(sm) gets updated for each discovery and binding in the S_{log}, which can be used for further optimisations.

9 Conclusion

To provide better interoperability it is not enough to have a good level of understanding on these WS-* Security Standards but collaboration of these standards need to be clearly known without any discrepancies. Web Services Interoperability Organization (WS-I) provides security profiles which specifies the best combinations of these standards, yet it is difficult to devise a customized collaborative security standard and a framework to implement the standard which is proposed in this paper. The Optimization of the customization process can be performed by the logs maintained by the Security Manager component which will be taken care during the implementation of the above proposed framework.

References

1. Sinha, S., Sinha, S.K., Purkayastha, B.S.: Security Issues in Web Services: A Review and Development Approach of Research Agenda. AUJST: Physical Sciences and Technology 5(II) (2010)
2. Zhang, Y., Sun, C., Yang, J., Wang,Y.: Web Services Security Policy. In: International Conf. on Multimedia Information Networking and Security (2010)
3. Liu, W.-j., Li,Y.: Research and Implementation Based on Web Services Security Model. In: International Conference on Innovative Communication and Asia-Pacific Conference on Information Technology and Ocean Engineering (2010)
4. Nortbotten, N.A.: XML and Web Service Security Standards, IEEE Communications Surver & Tutorials, 3 (Third Quarter 2009)
5. Kadry, S., Smaili, K.: A Solutions for Authentication of Web Services Users. Information Technology Journal 6(7), 987–995 (2007)
6. Geuer-Pollman, C., Calessens, J.: Web Services & Web Services Security Standards. Information Security Technical Report, 10, 15–24, Published by Elsevier (2005)
7. WSDL Binding for SOAP 1.2, `http://schemas.xmlsoap.org/wsdl/soap12/soap12WSDL.htm`

8. WS-SecurityPolicy 1.2 (July 1, 2007), http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html
9. Web Service Policy 1.5 Framework, W3 Recommendations, 4 (September 2007), http://www.w3.org/TR/2007/REC-ws-policy-20070904
10. WS-SecureConversation 1.3, OASIS Standard (March 1,2007), http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html
11. WS-Trust 1.3, OASIS Standard (March 19, 2007), http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html
12. Web Services Security: SOAP Message Security1.0 (WS-Security 2004), OASIS Standard (March 01, 2004), http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0
13. Chakhar, S., Haddad, S., Mokdad, L., Mousseau, V., Youcef, S.: Multicriteria Evaluation-Based Conceptual Framework for Composite Web Service selection, http://basepub.dauphine.fr/bitstream/handle/123456789/5283/multicriteria_mokdad.PDF?sequence=2