International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P), Volume-3, Issue-8, August 2015

Deriving secure extended WSDL for composition of web services

P.Prabhavathy, S.Bose

***Abstract*— Nowadays, web services are widely used because of its features such as loosely couple, interoperable, reusable and so on. The Web Service Definition Language (WSDL) file is used to provide various functional and non-functional descriptions of a web service. The Simple Object Access Protocol (SOAP) is used to transfer messages between various web services in a distributed manner. Service-Oriented Architecture (SOA) is a concept of building applications by assembling web services that are components of business functionalities. In SOA, this assembly of services is referred as a composite service. SOA is a convenient way for satisfying functional requirements, but it is more difficult to satisfy the non-functional requirements such as security. In composite web service, the workflow among component web services is framed using BPEL, which has to be defined in the extended WSDL document. The extended WSDL document has critical information such as definitions for service invocations, messages, bindings and workflow. Hence this extended WSDL document has to be secured from hackers for proper functioning of an enterprise application. Existing works focus on security for composite web service at the SOAP message level. In this paper, a novel way of securing composite web service is done by securely publishing the extended WSDL using Trusted Third Party (TTP). Two types of attacks are possible with respect to any WSDL document. They are scanning and parameter tampering which are threats to confidentiality and integrity respectively. Hence they can be overcome by using encryption and hash algorithms. The experiment results show that securely publishing an extended WSDL leads to an optimally secured composite web service in any enterprise application.**

***Index Terms*— Web services, Composition of services, extended WSDL protection, WSDL security**

I. INTRODUCTION

Service-Oriented Architecture (SOA) is a concept of building applications by assembling web services that are components of business functionalities. The web service provider publishes its WSDL in the UDDI registry. The service requester discovers apt WSDL based on its requirement by searching the UDDI registry [1]. Hence the UDDI registry is made publicly accessible. The published extended WSDL for composite web service contains critical information such as workflow among web services, message exchange patterns, service port bindings, input messages, output messages, fault messages, XML schema for message parts [2]. This scenario makes the WSDL document prone to more security attacks. Some security attacks on WSDL are scanning and parameter tampering. Scanning is a threat to confidentiality whereas tampering is to integrity [3].

P.Prabhavathy, DCSE, CEG campus, Anna University, India Research Scholar,
S.Bose, DCSE, CEG campus, Anna University, India, Associate Professor

Hence encryption and hashing algorithms can be used to overcome these security attacks. In symmetric, the key used for encrypt and decrypt processes are the same whereas it is different for asymmetric. In this paper, we have used an asymmetric encryption algorithm RSA. It uses public key cryptography. Public key cryptography is used because it uses a matched pair of keys, one for the encryption and the other for decryption. In encryption process, the sender encrypts using the receiver's public key that can be shared widely. The recipient decrypts using private key that known only to them. This helps to take over the difficulty of establishing confidential communication for key exchange. Secure Hashing is a common method for ensuring message integrity. It ensures that the received message has not been tampered and that message is the exact copy of original form.

In this paper, we used a novel technique by introducing Trusted Third Party (TTP) for providing key pair generation, encryption and hashing functionalities. This TTP securely publishes the extended WSDL in the public UDDI registry. The extended WSDL of composite web service is sent from the provider to the TTP. The TTP uses DOM parser for fetching the values of critical element tags and attributes in the extended WSDL document. These fetched values are encrypted using public key of the privileged service requester. The extended WSDL document is rebuilt by replacing the critical values with encrypted values. This critical value encrypted (CVE) extended WSDL is provided as input to the secure hash algorithm (SHA). SHA generates 160 bits length message digest value as output. The CVE extended WSDL is published along with message digest value in the UDDI registry.

The rest of this paper is organized as follows: Section 2 explains the existing works in the composition of web services and its security. Section 3 provides the hierarchical composition of web services for which the extended WSDL needs to be secured. Section 4 explains the novel work in securing the extended WSDL for composite web service. Finally, this paper concludes with Section 5.

II. RELATED WORKS

Many research works are done in the area of composition of web services. But securing composite web service is a leading demand from enterprise systems. In the paper [1] security is provided using aspects. Static approach of policy framing for security is given in [2]. Paper [3] implements SOAP level security mechanism for protecting WSDL document which is not suitable for composite web service. Security Match Maker [4] is provided by matching security requirements of service provider and requestor in SOA. In [5] constraint solver is used to resolve various constraints on

messages and parameters by preserving semantics for automatic generation of composition of services as message sequence chart.

In paper [6] Key building blocks for framework are semantic model for specifying security objectives and properties at the system and service levels, and the negotiation and re-negotiation techniques for service composition and evolution. In [7] three-phase composition protocol integrating information flow control is done with transformation factor to model the computation effect of intermediate services. Definition for process-independent policy composition rules along with provision of method for semi-automatically creating a security policy of the composite service is done in [8]. A method for modeling security constraints and a brokered architecture to build composite Web services according to the specified security constraint is proposed in [9]. A logic based approach used for specifying authorization policies and detecting conflicts resulting from the combination of various kinds of authorization and constraint policies used in Web services environments is provided in [10].

An access control model via a declarative policy specification language which uses pure-past linear temporal logic (PPLTL) is proposed in [11]. In [12] EXPTIME completeness of service composition, allows us to say that checking simulation from a single deterministic transition system to the asynchronous product of to n deterministic transition systems is an EXPTIME-complete problem.

III. Hierarchical Composition of Web Services

Service oriented architecture gives the provision of business logic as independent services. Consider an example health system with just one medical testing laboratory that serves two hospitals H1and H2, each with two specialized clinics, C1 and C2, C3 and C4 respectively.

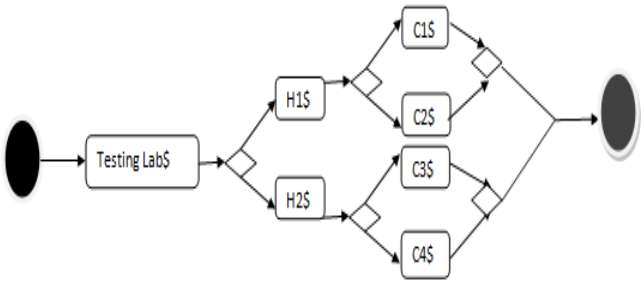

Figure 1: Workflow for an example health system

The medical testing laboratory reads the patient's UHID (Unique Health Identifier), identifies the corresponding hospital and sends the patient test results to it. Hospital checks the category of medical test performed and forwards the test result to corresponding specialized clinic. The doctor who prescribed the medical test is provided with the patient test result for further medical diagnosis. Composite web service is a web service that is composed from sub-services. For the above mentioned example, the workflow of the health system is shown in Figure 1, where $ denotes a parameter. This workflow is parameterized over all the operations of every web service involved.

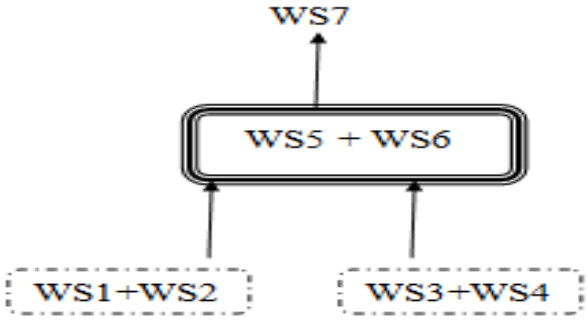

Figure 2: Hierarchical composition

Let WS1, WS2, WS3, WS4, WS5, WS6 and WS7 are web services. Here WS1, WS2, WS3 and WS4 are atomic web services with respect to WS5 and WS6 composite web services. WS5 and WS6 are atomic web services with respect to WS7 composite web service. We defined it hierarchically, composed web services to give composite web service, which in turn can be composed into even bigger composite services. This is illustrated by Figure 2.

From the figure 2, it is obvious that web services WS1 and WS2 are composed by a composition operator (Comp) to get a composite web service WS5. WS1 and WS2 have their interfaces described in the standard WSDL file, whereas WS5 has an interface that cannot be described in standard WSDL, because WS5 contains workflow embodied in the composition operator (Comp). So in order to define WS5 as a web service, we need to extend standard WSDL in order to incorporate workflow descriptions. Then we need to devise a method to generate its interface in the extended WSDL from the standard WSDL interfaces of WS1and WS2.

We define a new extensible element for WSDL documents, called workflow. It contains child elements which describe details of workflow structure. The extended WSDL document for a composite web service consists of standard elements such as types, messages, portType, binding and services, together with the additional workflow element, as shown in Figure 3.

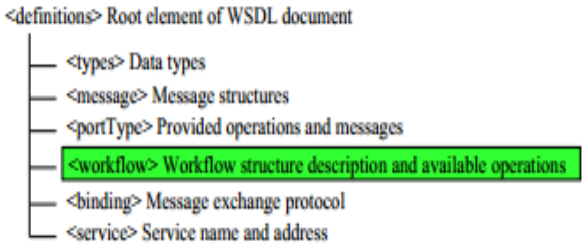

Figure 3: extended WSDL structure - composite web service

Extensible tags under workflow tag (highlighted in Figure 3) describe workflow structure. Connector behavior in composition is defined by their implementation on the concerned web server. Each connector tag has child tags specifying the services and their corresponding operations. Pipe and selector are two possible connections in workflow. If a connector provides sequential invocation, then the child tags

describe the sequence of services involved. If a connector provides a branching structure, then the child tags specify the branching condition and the corresponding services.

```
<workflow>
<pipe><set name="Testing Lab"><operation name="processUHID" > ... </operation>
</set>
<choice>
<case condition="1">
<pipe><set name="H1"><operation name="getHospitalName" > ... </operation></set>
<choice>
<case condition="1"><set name="C1">
<operation name="cash bill">...</operation>
<operation name="test report">...</operation> ...</set></case>
<case condition="2"><set name="C2">
<operation name="cash bill">...</operation>
<operation name="test report">...</operation> ...</set></case>
</choice>
</pipe>
</case>
<case condition="2">
<set name="H2"><operation name="getHospitalName" > ... </operation> .....</set>
```

Figure 4: Pipe and choice connections in extended WSDL

Schema for workflow consisting of pipe and selector connectors is depicted in Figure 4. Pipe and Choice may in turn contain one another. The pipe tag is used to represent the pipe connector which invokes every service, or passes requests to structures, in the sequence. The choice tag represents the selector connector, which has a number of cases specified by the case tag.

A case is a combination of a matching condition and an operation set (i.e. service) or another workflow structure. The choice workflow invokes a service or a structure if the corresponding matching condition is satisfied.

Extended WSDL document contains critical data such as definitions of types, messages and portTypes. This information can be used by the hacker for creating security breach into the composite web service access. Figure 5(i) shows types definition, message definition is depicted in Figure 5(ii) and portTypes definition in Figure 5(iii)

The definition of types and output messages in extended WSDL document for composite service having selector as top-level connector will be same as those for pipe connector. But invoke operation of selector has different signature (condition included) shown in Figure 6, affects input message and portType definitions.

A composite web service has all its operations available from the web services involved for the composition process. Figure 7 shows the composite service binding in the extended WSDL.

```
<wsdl:types>
<schema targetNamespace="urn:cbsd" .../>
<complexType name="ArrayOfString">
<sequence><element name="item" type="xsd:string"/>
</sequence></complexType></schema>...</wsdl:types>
```

```
<wsdl:part name="params" type="ArrayOfString"/>
</wsdl:message>
<wsdl:message name="invokeResponse">
<wsdl:part name="result" type="xsd:string"/>
</wsdl:message>
```

```
<wsdl:portType name="...">
<wsdl:operation name="invoke" parameterOrder="operations params">
<wsdl:input message="invokeRequest".../>
<wsdl:output message="invokeResponse".../>
</wsdl:operation>
</wsdl:portType>

<wsdl:message name="invokeRequest">
<wsdl:part name="condition" type="xsd:string"/>
<wsdl:part name="operations" type="impl:ArrayOfString"/>
<wsdl:part name="params" type="impl:ArrayOfString"/>
</wsdl:message>
...
<wsdl:portType name="...">
<wsdl:operation name="invoke" parameterOrder="condition operations params">
<wsdl:input message="impl:invokeRequest".../>
<wsdl:output message="impl:invokeResponse".../>
</wsdl:operation>
</wsdl:portType>
```

Figure 5: (i) Types definition
(ii) Message definition
(iii) PortTypes definition

```
<wsdl:message name="invokeRequest">
<wsdl:part name="operations" type="ArrayOfString"/>
...
<wsdl:message name="invokeRequest">
<wsdl:part name="condition" type="xsd:string"/>
<wsdl:part name="operations" type="impl:ArrayOfString"/>
<wsdl:part name="params" type="impl:ArrayOfString"/>
</wsdl:message>
...
<wsdl:portType name="...">
<wsdl:operation name="invoke" parameterOrder="condition
operations params">
<wsdl:input message="impl:invokeRequest".../>
<wsdl:output message="impl:invokeResponse".../>
</wsdl:operation>
</wsdl:portType>
```

Figure 6: Selector in extended WSDL

```
<wsdl:binding name="..." type="...">
<wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="invoke">
 <wsdlsoap:operation soapAction=""/>
<wsdl:input name="invokeRequest">
<wsdlsoap:body encodingStyle="..." namespace="urn:cbsd"
use="encoded"/></wsdl:input>
<wsdl:output name="invokeResponse">
...
<wsdl:service name="..."> <wsdl:port binding="..." name="...">
<wsdlsoap:address location="http://server/composite_service"/>
</wsdl:port>
</wsdl:service>
```

Figure 7: Composite service binding in extended WSDL

IV. Securing the Extended WSDL

The extended WSDL is securely published in the public UDDI registry by the Trusted Third Party (TTP). TTP provides key pair generation, encryption and hashing functionalities. The extended WSDL of composite web service is sent from the web service provider to the TTP. The TTP uses DOM parser for fetching the values of critical element tags and attributes in the extended WSDL document such as PortTypes, messages, bindings, workflow, pipe, case, etc.

Existing works consider encryption at element tag but not attribute values. Attribute value also contains critical value that has to be encrypted for secured composite web service. Hence the values fetched from the DOM parser are encrypted using public key of the privileged service requester. The extended WSDL document is rebuilt by replacing the critical values with encrypted values. This critical value encrypted (CVE) extended WSDL is provided as input to the secure hash algorithm (SHA). SHA generates 160 bits length message digest value as output. The CVE extended WSDL is published along with message digest value in the UDDI registry.

The privileged service requester has got the key pair from TTP during key generation process. The public key is used by TTP for encryption. Hence the privileged service requester can alone decrypt the CVE extended WSDL published in the UDDI registry. Thus it overcomes the scanning attack.

The CVE extended WSDL can be digested using SHA provided by TTP. This helps the privileged user to check integrity of CVE extended WSDL document provided in the registry. Thus it ensures integrity by overcoming parameter tampering attack.

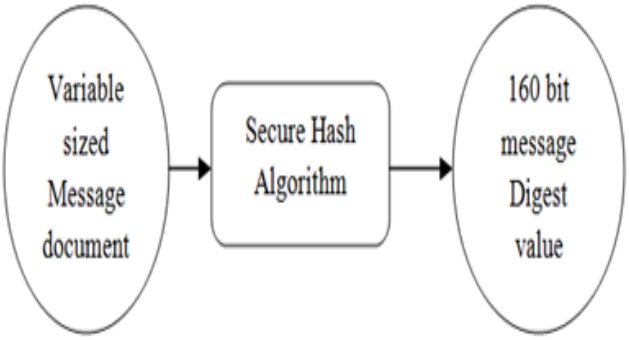

Figure 8: Message digesting process

Hashing follows a mathematical scheme for providing integrity of the document. Any size of input message to SHA yields only fixed 160 bits length message digest as output. A matched digest value ensures recipient that the document has not been altered during transit. Message digest is a string of binary digits computed using a set of rules to ensure the integrity of document. The process of digesting variable sized message into fixed length digest value is shown in figure 8. The overall functionality provided by the Trusted Third Party (TTP) is shown in the figure 9.

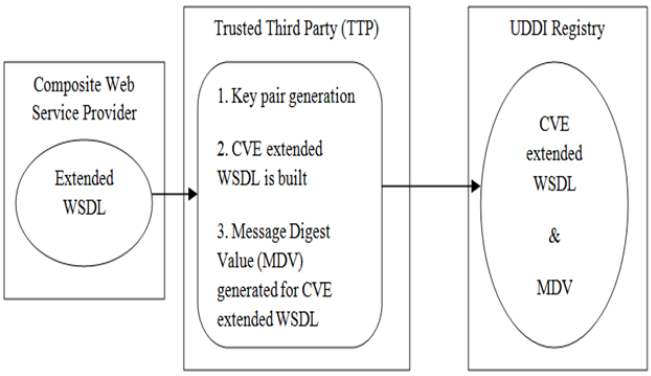

Figure 9: Overview of securing Extended WSDL system

The service requester is considered as privileged requester if and only if he has registered with the Trusted Third Party (TTP). During registration process, TTP provides key pair along with security algorithms. The CVE extended WSDL along with MDV is fetched from the registry.

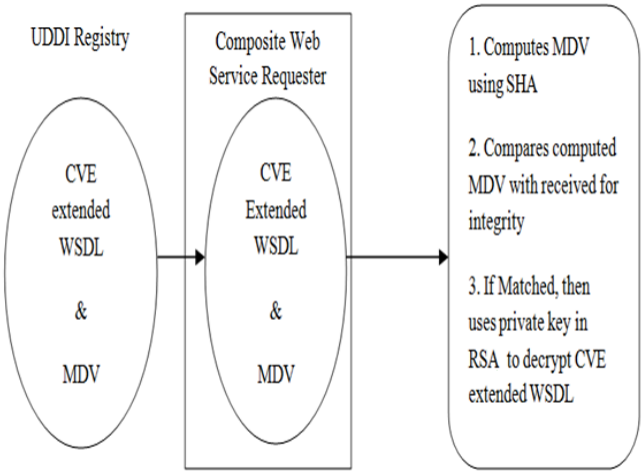

Figure 10: Secured access of extended WSDL by privileged service requester

Using SHA, the MDV is computed for fetched document and compared with the received MDV. If a match occurs, then the service requester uses his private key for decrypting the CVE extended WSDL to read the critical value information provided for the successful operation of composite web service.

V. Conclusions

Security of SOAP messages is dealt for Composite web service security in the previous works. The WS-Security standard provides the protection for SOAP messages. In this paper, a novel way of securing composite web service is done by protecting the extended WSDL with the aid of trusted third party.

The extended WSDL document contains explicit instructions on how to communicate with component web services, which can cause serious security breach if those web services are compromised in the composite web service. This document has critical information such as PortTypes,

messages, bindings, workflow, pipe, case, etc. Hence securing the extended WSDL is very important.

A composition of web services is done by composing atomic web services in hierarchical manner. An extended WSDL is generated for the composite web service. The extended WSDL is securely published in the public UDDI registry by the Trusted Third Party (TTP). TTP provides key pair generation, encryption and hashing functionalities. The TTP uses DOM parser for fetching the critical values from the extended WSDL document. It uses public key of the privileged service requester for encrypting the fetched critical values (elements/attributes).

The extended WSDL document is rebuilt by replacing the critical values with encrypted values. This critical value encrypted (CVE) extended WSDL is provided as input to the secure hash algorithm (SHA). SHA generates 160 bits length message digest value as output. The CVE extended WSDL is published along with message digest value in the UDDI registry.

Thus our approach is novel since attribute as well as element tag values are considered for encryption in extended WSDL for composite web service security. Functional requirements are fully satisfied by hierarchical composition of atomic web services. Non-functional requirement such as security can be satisfied by protecting extended WSDL against various attacks such as scanning and parameter tampering.

REFERENCES

[1] A.Charf, M. Mezini,"Using Aspects for Security Engineering of Web Service Compositions," Proceeding IEEE International Conference Web Services (ICWS '05), pp. 59-66, 2005.
[2] Mohsen Rouached,Claude Godart "Reasoning about Events to specify Authorization Policies for Web Services Composition" 2007 IEEE International Conference on Web Services (ICWS 2007).
[3] M. Srivatsa, A. Iyengar, T. Mikalsen, I. Rouvellou, and J. Yin, "An Access Control System for Web Service Compositions," Proceeding IEEE International Conference Web Services (ICWS '07), pp. 1-8, 2007.
[4] D. Berardi, F. Cheikh, G.D. Giacomo, F. Patrizi Automatic service composition via simulation Int. J. Found. Comput. Sci., 19 (2) (2008), pp. 429–451
[5] Bartoletti, M., Degano, P., Ferrari, G.L.: Enforcing secure service composition. In: Proc. 18th Computer Security Foundations Workshop (CSFW) (2005)
[6] B. Carminati, E. Ferrari, P.C. K. Hung, "Web Service Composition: A Security Perspective", WIRI, 2005, Proceedings. International Workshop on Challenges in Web Information Retrieval and Integration 2005, pp. 248-253,
[7] Yannick Chevalier, Mohamed Anis Mekki , Michael Rusinowitch "Automatic Composition of Services with Security Policies" 2008 IEEE Congress on Services
[8] Jun Han, Ryszard Kowalczyk ,Khaled M. Khan, Security-Oriented Service Composition and Evolution, XIII ASIA PACIFIC SOFTWARE ENGINEERING CONFERENCE (APSEC'06)
[9] Security-Aware Service Composition with Fine-Grained Information Flow Control, Services Computing, IEEE Transactions on (Volume:6 , Issue: 3), Page(s): 330 – 343 Date of Publication : 17 January 2012
[10] Fumiko Satoh, Takehiro Tokuda, "Security Policy Composition for Composite Web Services", IEEE Transactions on Services Computing, vol.4, no. 4, pp. 314-327, October-December 2011,
[11] Barbara Carminati , Elena Ferrari, Patrick C. K. Hung, Security Conscious Web Service Composition, IEEE International Conference on Web Services (ICWS'06)
[12] Brucker, A.D. ; SAP AG, Karlsruhe, Germany ; Malmignati, F. ; Merabti, M. ; Qi Shi ,A Framework for Secure Service Composition, Social Computing (SocialCom), 2013 International Conference on, 8-14 Sept. 2013, 647 – 652, Alexandria, VA

P.Prabhavathy received her B.E. and M.E. degrees in Computer Science & Engineering in the year 2005 and 2010, respectively. During 2005-2006, she worked as software engineer in Benchmark Electronic Systems for developing network security demonstration kit for engineers study purpose. Currently, she is a research scholar in anna university and her area of interest includes mobile computing, web services, wireless XML broadcasting information systems.

S.Bose is one of the leading cryptographer scientists in India and has written a book titled "Cryptography and Network Security". He had obtained PhD in the area of Information Security from Anna University in 2007. He is the Dean-in-charge in University College of Engineering, Arni. He is a renowned expert and mass stunning speaker who have inspired a lot of students on ethical hacking through his hands on experience sessions. His publications in highly indexed journals talks about the vast knowledge and skill set in the area of cryptography. His research interest is also extended in the areas of network analysis using social networking, 4G communication, group key management in multicasting, information security.