Securing Agent and Web Services on the Semantic Web using Cryptographic Algorithm and PKI

Rashmi Pandey[#1], Suresh Kumar[*2] , Dr. Manjeet Singh[#3]

[##], Department of C S E,IP University
Ambedkar Institute of Technology, Geeta Colony, New Delhi India
[*] Department of CSE
Ambedkar Institute of Technology,Geeta Colony, New Delhi, India
[#] Department of CE, YMCA
University of Science & Technology Faridabad, India

Abstract
Data is one of the most important, confidential and private data on the World Wide World (WWW). Many of the organization have secure source of the data. But these data is not so confidential on the WWW because these data is extracting at the regular basis. These data source can be managed by the database management system or by the database warehouse. The privacy of the data is crack by the attacker on the WWW. Semantic web is auxiliary extracting these data from the WWW. Semantic web is a collection of different web services, these web services are used by the Agent/client on the web. Semantic web manage all type of request send by the Agent/client. Data and web services are extracted by the Agent is more porn to attack and require high level of security contemplation. Paper proposing a model for protecting the web services request by the Agent.

KEY WORDS
Cryptographic Algorithm, PKI, XKMS, SAML, Authentication, Authorization

I. INTRODUCTION

Semantic web is a group of methods and technologies to allow machines to understand the meaning or "semantics" of information on the WWW and understand the web pages. The term was coined by WWW consortium (W3C "specified standards for the semantic web") director Tim Berners Lee. He defines the semantic web as "a web of data that can be processed directly and indirectly by machines" [1].

Semantic web is an extension of current web in which information is given well defined meaning, better enabling computers and people to work in cooperation. The main aim of this version is to make the web more intelligent so that there will be less human intervention. The main purpose of the semantic web is driving the evolution of the current web by allowing users to use it to its full potential thus allowing users to find, share, and combine information more easily. Humans are capable of using the web to carry out tasks such as finding the Iris, word for "folder", reserving a library book, and searching for a low price for a DVD. While the term semantic web is not fully defined, it is mainly used for describing the technologies such as XML (eXtensible Markup Language), RDF (Resources Description Framework), and OWL (Web Ontology Language), methods and model proposed by the W3C.

Much progress has been done toward technologies and standards for the semantic web, still lots of work to be done in terms of security, privacy, confidentiality and integrity of agent and web services. In the case of privacy, a user enters his or her personal identification information after reading the privacy policy enforced by the web site server. In the case of confidentiality, the goal is to secure the request of the user, the user only get the information that he or she is authorized to see. In the case of integrity, the goal is to secure the message that is transmitted through one place to another; the message should not be tempered [2].

II. BASIC CONCEPT

A. *Web Services*

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. XML-based technologies such as SOAP, XML Schema and WSDL provide a broadly-adopted foundation on which to build interoperable Web services. The WS-Policy and WS-Policy Attachment specifications extend this foundation and offer mechanisms to represent the capabilities and requirements of Web services as Policies.

The Web service technology comprises three basic standards [3][4][5]:

- Web Services Description Language (WSDL): a layout for describing the functionality of services on the web.
  1. It also defines services as collections of network endpoints or ports.
  2. WSDL is an XML document.
  3. it is also used to locate Web services.
- Universal Description Discovery & Integration (UDDI): a registry that supports service periodical and innovation.
  1. UDDI stands for Universal Description, Discovery and Integration

2. UDDI serves as a “Business and services” registry and directory and are essential for dynamic usage of Web services.
3. Is a platform-independent framework for describing services, discovering businesses, and integrating business, and integrating business services by using the internet.

- SOAP (formerly Simple Object Access Protocol): a protocol for message exchange among services. SOAP is a message layout specification that defines a uniform way of passing XML-encoded data. It also defines a way to bind to HTTP as the underlying communication protocol. SOAP is basically a technology to allow for “RPC over the web” [4][5][6].

B. Architecture of Semantic Web Process:-

A model is a detachment of the architecture that typically revolves around a particular facet of the overall architecture. Message Oriented Model focuses and explains Web services strictly from a message transitory perception. in general, it is not important to correlate message to the service provider. The Service Oriented Model is lies on the top of the architecture and extract the Message Oriented Model to explain the properties and fundamental concepts involve in service- infect it extract to explain the purpose of the message in the Message Oriented Model [7].

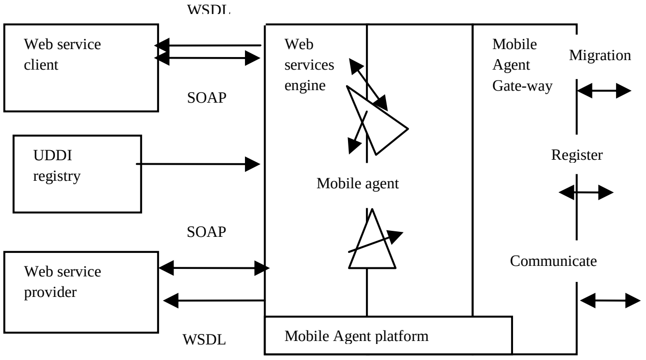

Figure 1. Architecture of Semantic Web Process

Figure1 shows, UDDI is universal directory to publish web services to all the users for accessing the web services and web service engine enables agent contribution summarize functionality as web services (service provider) as well as agent utilizing web services(service client). WSDL is used to define the information and the configuration setting of web service. SOAP is simple object access protocol that defines the standard structure of the information or data to send and receive data. Mobile agent platform provides resources to define and to execute process for the agents, one agent might provide functionality provided by the web services which in turn can be used by the another agent, in this case, the another agent is acting as web service client[8].

C. Web Services Innovation

Web service provider must assure that the confidentiality, integrity, privacy and availability of the information or the data that is collected, maintained, use, or transmit by the users is protected. The confidentiality of the information is not only vulnerable by the risk of improper access to stored information, but also by the risk of interception during transmission.

Network identity is extremely important for maintaining privacy and identity control because it provide and offers single sign-on (SSO) functionality to users in one domain or in another domain for the alleviate of use and swift access to resources [9].

1) Web Services Security:

- *Authentication*

If your Web service is sensitive, restricted data or if it provides restricted services, it needs to authenticate callers of the web services. A number of authentication schemes are available and these can be broadly divided into three categories:

Platform Level Authentication

If you are in control of both the points and both points are in the same domain or trusting domains, you can use Windows authentication to authenticate the callers.

Basic Authentication

The end user must configure the proxy and provide credentials in the form of a user name and password. The proxy then transmits user name and password with each Web service request through that proxy. The credentials are transmitted in plaintext and therefore you should only use Basic authentication with SSL (Secure socket Layer).

The following code fragment shows how a web application can extract the basic information of the end user to authenticate them and use this information to provide the web services to the end user and invoke a downstream Web service configured for Basic authentication.

```
// Retrieve client's credentials (available with Basic
authentication)
String pwd = Request.ServerVariables
["AUTH_PASSWORD"];
String uid = Request.ServerVariables ["AUTH_USER"];
// Set the credentials
CredentialCache cache = new CredentialCache ();
Cache. Add (new Uri (proxy.Url), // Web service URL
      "Basic",
       New NetworkCredential (uid, pwd, domain));
proxy.Credentials = cache;
```

Integrated Windows Authentication

The advantage of this approach is that, in comparison to Basic authentication information and credentials are not sent over the network, which eliminates the network eavesdropping threat.

To call a web services which are configured for the windows authentication, the end user must clearly configured or construct the credentials property on the proxy server. To flow the security perspective of the client's Windows security perspective to a Web service you can set the **Credentials** property of the Web service proxy to **CredentialCache**.**DefaultCredentials** as follows.

```
proxy.Credentials =
System.Net.CredentialCache.DefaultCredentials;
```

You can also use an explicit set of credentials as follows:

```
CredentialCache cache = new CredentialCache ();
Cache. Add( new Uri(proxy.Url), // Web service URL
        "Negotiate",        // Kerberos or NTLM
         new NetworkCredential(username, password, domain));
proxy.Credentials = cache;
```

If you need to stipulate explicit or plain credentials, do not store the credentials in a plain text form or in a hard form. For the security point of view, encrypt the credentials property and information and store the encrypted form of information or data either in an **<appSettings>** element in Web.config or beneath a restricted registry key.

2) Message Level Authentication

This approach allows you to pass authentication tokens in a standard format or in standard way by using SOAP headers. When two parities are agreed to use web services , the defined format of the authentication token must also be granted upon. The following types of authentication token can be used:

- User name and password
- Kerberos ticket
- X.509 certificate
- Custom token

User Name and Password

Firstly you send your user name and password information in the SOAP header message. These information are sent in a plaintext format, so this approach be supposed to be used in conjunction with SSL due to the network eavesdropping threat. The credentials are sent as part of the **<Security>** element, in the SOAP header as follows.

```
<Security
xmlns="http://schemas.xmlsoap.org/ws/2002/12/secext">
  <Username Token>
    <Username>Bob</Username>
    <Password>YourStr0ngPassWord</Password>
   </Username Token>
</Security>
```

User Name and Password Digest

Instead of sending a user name and plaintext password, you can send a password digest. The digest is a Base64-encoded SHA1 hash value of the UTF8-encoded password. This approach is used over the secure channel but the data or information still be tempered and intercepted by the attackers using network monitoring mode and use this information to gain authenticated access to your web services. To protect this information from this replay attack threat, a nonce and a creation timestamp can be combined with the digest.

User Name and Password Digest with Nonce and Timestamp

Digest of any information is defined as the; SHA1 hash of a nonce value, a creation timestamp, and the password.

Digest = SHA1 (nonce + creation timestamp + password)

In this approach, the web services must store all the registered nonce values in the table and reject the nonce if they are duplicate in the table. This approach protects the password and prevents them from the replay attacks.

The main disadvantage of this is, it suffers from the clock synchronization issues between the sender and the receiver when calculating the expiration time and it does not prevent capturing of message by the attacker, in this case, the attacker modifying the nonce values and then replaying the message to the web services. To deal with this threat, the message must be digitally signed. You can sign a message using a custom token or an X.509 certificate. This provides tamper proofing and authentication, based on a public key and private key pair.

Kerberos Tickets

You can send a security token that contains a Kerberos ticket are as follows.

```
<Security
       xmlns="http://schemas.xmlsoap.org/ws/2002/12/secext">
  <BinarySecurityToken
         Value Type="Kerberosv5ST"
         Encoding Type="Base64Binary">
    U87GGH91TT...
  </BinarySecurityToken>
</Security>
```

X.509 Certificates

You can also provide authentication by sending an X.509 certificate as an authentication token.

```
<Security
       xmlns="http://schemas.xmlsoap.org/ws/2002/12/secext">
  <BinarySecurityToken
        Value Type="X509v3"
        Encoding Type="Base64Binary">
    Hg6GHjis1...
  </BinarySecurityToken>
</Security>
```

For more information about the above approaches, see the samples that ship with WSE.

3) Application Level Authentication

You can design and build your own custom authentication by using custom SOAP headers for your application. Before doing so, review the features provided by the platform. If you must use a custom authentication mechanism, and you need to use cryptography, then use standard encryption algorithms showing by the **System.Security.Cryptography** namespace.

- *Authorization*

After authentication, the restricted users are able to access the web services given by the web services provider, based on the user identity and role defined to him or her. You can restrict access to service endpoints, individual Web methods, or specific functionality inside Web methods.

1) Web Service Endpoint Authorization

If your Web service is configured for Integrated Windows authentication you can configure NTFS permissions on your Web service (.asmx) files to control access, based on the security context of the original caller. This authorization is performed by the ASP.NET **FileAuthorizationModule** and impersonation is not required.

2) Web Method Authorization

You can use declarative principal permission demands to control access to individual Web methods based on the identity or role membership of the caller. The caller's identity and role membership is maintained by the principal object associated with the current Web request (accessed through **HttpContext.User**.)

```
[Principal Permission (SecurityAction.Demand, Role=@"Manager")]
[Web Method]
Public string QueryEmployeeDetails (string empID)
{
}
```

- *Improving Web Service efficiency using SOAP Message:-*

Secure Message

In addition to requiring the use of addressing, requires the use of transport-level security for protecting messages.

```
<Soap: Envelope>
  <Soap: Header>
    <wss: Security soap: must Understand="1" >
      <wsu: Timestamp u:Id="_0">
        <wsu: Created>2006-01-19T02:49:53.914Z</u: Created>
        <wsu: Expires>2006-01-19T02:54:53.914Z</u: Expires>
      </wsu: Timestamp>
    </wss: Security>
    <wsa: To>http://real.contoso.com/quote</wsa:To>
    <wsa:
Action>http://real.contoso.com/GetRealQuote</wsa:Action>
  </soap: Header>
  <soap: Body>...</soap: Body>
</soap: Envelope>
```

The SOAP message in the example above includes security timestamps that express creation and expiration times of this message. It requires the use of security timestamps and transport-level security - such as HTTPS – for protecting messages. (The prefixes wss and wsu are used here to denote the Web Services Security and Utility namespaces.) [9]

III. SCENARIO TO SECURE WEB SERVICES:-

A. *XKMS Service*:-It stands for XML Key Management Service.

1) XKMS uses the web services framework to make it easier for developers to secure inter-application communication using PKI.
2) XKMS is a protocol, which is developed by W3C.
3) It describes the distribution and registration of public keys pairs.

There are two types of XKMS services

XKISS: - it stands for XML key information service specification. it is used for the management of the component of the public in public key pair.

XKRSS: - it stands for XML Key Registration service specification. It concern for the management of the component of the private key in private key pair.

B. *SAML*: -It stands for the Security Assertion Markup Language.

1) It is based on the XML message called Assertions. Assertion contain information and statements that determine whether the user is authenticated user who claims that he is authenticated and authorized to use the resources.
2) SAML provide Single Sign-On capability so that the agent in the different domains must communicate with each other across the trusted domains such as web sites or web services.

There are four Steps for securing the Agent on the Semantic Web using Cryptographic Algorithm, XKMS, SAML, and PKI, PKI is an infrastructure to protect internet transaction as a whole system. It is the dual public-key cryptosystem which means there are two keys, public-key and private-key pairs for each PKI entity [11].

Step 1:

In the first step, agent/client request for the generation of pair of keys. The pair of keys (public and private) is generated using RSA algorithm. When keys are generated then it will sent to the agent.

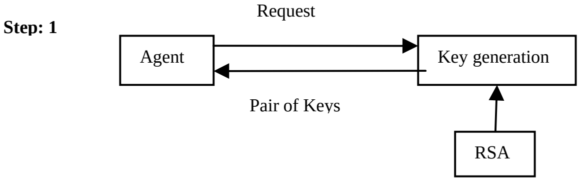

Figure 2. Generation of keys

Step 2:

Now agent sends the pair of keys to the XKMS services to registration process. This information's are stored in the PKI database.

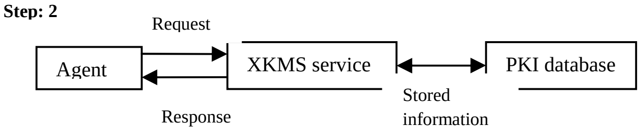

Figure 3. Registration Process

Step 3:

Now, when the registration process is complete, agent creates a SOAP message. SOAP message contain request, signature, and other security parameters and encrypt the message using a session key and send the XKMS Request to the SAML Security Authority or for the secure communication over the internet. Security authority decides that the client or agent who sends the request for the web services are authenticated person and authorized to use the web services. When it validates then it send XKMS Response to the Agent and provide single sign-on to the agent.

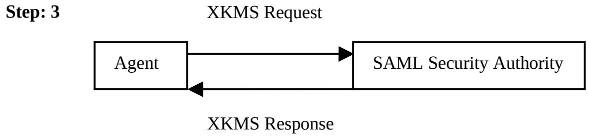

Figure 4. Authentication Process

Step 4:
Now the agent/client in one domain is able to securely communicate with the other agent/client in other domain using single sign-on.

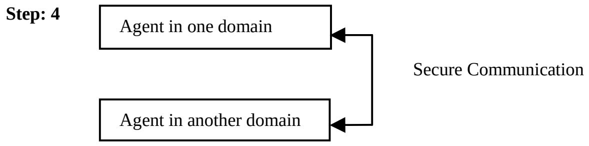

Figure 5. Single Sign-On

IV. Proposed Scenario: -

Our propose system basically deal with the multi Agent System and to secure the web services which is requested by the agent on the behalf of their client. When the semantic web services retrieve the data or information from the database, it is more prone to attack. The attacker analyzes and captures the data which transfer from one place to another and this will produce replay attack, DDOS attacks, probing attack etc, so this will needs high security consideration.

The main constituent of the proposed system as shown in the figure:-

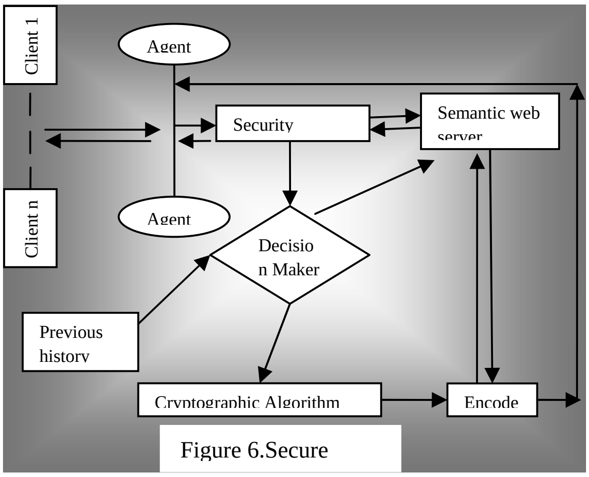

Figure 6.Secure

A. *Agent/Client*

Agent sends the request with the time stamp for the Web services using his/her X.509 certificate to the security authority.

B. *Security Authority*

This meticulous component supervises the request send by the Agent, and it also work as the first interface for the Agent. It also decides the level of authorization. Security Authority also generates the Token T_1 for the complete process. Token will be generated by the authorization level. Token also capture the complete process from one end to another end. Now the complete information and the Token T_1 will be sending to the Decision Maker.

C. *Decision Maker*

Decision Maker decides which agents require more attention for security. This will be totally depending on the Previous History. If any of the data requested by the Agent is more confidential and private data, then this Decision Maker will decide to provide the data or not, and send it to the Cryptographic Algorithm.

D. *Cryptographic Algorithm*

Cryptographic Algorithm (Symmetric and Asymmetric) is responsible for providing algorithm for web services which is requested by the Agent/client and information given by the Decision Maker. It also generates key and generates Token T_2 and send it to the Encoder. It also sends the key to the original requestor, so that the Agent who is the original requestor is able to decipher the encoded information.

E. *Encoder*

Encoder encodes the semantic web services provided by the Server and send the encoded services along with token T_2 to the original requestor. Now the agent is able to view his/her information or data securely because the web services are in encoded form. When Agent receives response by the encoder, the information is change according to the token T_2, now the original requestor is able to decipher it using the key generated by the cryptographic Algorithm.

V. Conclusion and the Future work

Security has become a primary anxiety in every field so as in Semantic Web Services. Moreover day by day new threats are emerging and it is difficult to detect them, and there will be attacks on the web services requested by the Agent/client. Therefore there is a requirement for securing the web services. Section III describes the Authentication process, Authorization process and security of the SOAP message. The proposed Web services and Agent secure model is based on the XKMS and SAML can ensure secure communication and authentication with PKI. In future work, the algorithm and the coding are proposed for this process.

VI. ACKNOWLEDGEMENTS

AIT-G, Especially Faculty of Computer Science & Technology supported this work.

REFERENCES

[1] www.w3c.org

[2] Bhavani Thuraisingham, “confidentiality, privacy and trust policy enforcement for the semantic web”, Eighth IEEE international workshop on Policies for Distributed System and Networks, IEEE, 2007

[3] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 2004.

[4] T. Takse and K. Tajima “Efficient web services message exchange by SOAP bundling Framework” IEEE-2007, pp.65-67.

[5] Diego Zuquim Guimarães Garcia, Maria Beatriz Felgar de Toledo,” Web Service Security Management Using Semantic Web Techniques”, ACM, *SAC’08,* March 16-20, 2008

[6] Web Services Policy Framework, http:// www. ibm.com /developerworks/library/specification/ws-polfram/, 09 Mar 2006 (Published 01 May 2003)

[7] A.Gomez-Perez et al,”ODE SWS: A Framework for Designing and Composing Semantic Web Services”, University of Madrid, July/August 2004

[8] European Telecommunication Standards Institute, “Mobile Commerce (M-COMM); Mobile Signature Service; Web Service Interface”. Technical specification ETSI TS, 2003-08, pp.102-204

[9] S. Fugkeaw, P. Manpanpanich, S. Jantra premjitt.”A Robust Single Sign-On Authentication Model based on Multi-Agent System and PKI”, Proceedings of IEEE International Conference on Networking, ICN-2007.

[10] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan,” Building Secure Web Services ”,chapter-12,” http://msdn.microsoft.com/en-us/library/ff648643.aspx”,publised- June 2003

[11] T. Kataoka, “Federation of Campus PKI and Grid PKI for Academic GOC (Grid Operation Center)”, 24th APAN Meeting-2007, Vol.93, pp.733-866.

SHORT BIODATA OF ENTIRE AUTHOR

Rashmi Pandey received the B.Tech degree in Information Technology from Institute of Technology and Management, Gida, Gorakhpur (U.P. Technical Univ., Lucknow) India, in 2009 and pursuing M.Tech in Information Security from Ambedkar Institute of Technology, Govt. of NCT Delhi, Geeta Colony, New Delhi (Guru Govind Singh Indraprastha University, New Delhi), India.

Suresh Kumar received the M.Tech degree in Computer Science & Engineering from Department of Computer Science & Application, Kurukshetra University, Kurukshetra, Haryana, India in 2002 and pursuing Ph.D. from Faculty of Engineering & Technology, Maharshi Dayanand University, Rohtak, Haryana, India. His major field of study is Semantic Web. His current research interest includes Secure Semantic Web Services, Semantic Search, Cloud Computing, Cryptography and Network Security, Biometric Security.

He has more than nine years teaching experience. He is working as Assistant Professor in the Department of Computer Science & Engineering, Ambedkar Institute of Technology, Govt. of NCT Delhi, Geeta Colony, New Delhi, India. He is the author/co-author of more than 13 publications in International/National journals and conferences.

DR. MANJEET SINGH is currently working as an Associate Professor (CE) at YMCA University, Faridabad, Haryana, India. He has completed his M.Tech from GJU, Hissar and PHD from MDU University. Rohtak, Haryana, India. His areas of interests are Natural Language Processing and Internet Technology.