International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015
ISSN 2091-2730

SOAP Performance and Enhancement in WS-Security

Srinath KS[#1], Aditya Kumar[#2], Ankit Shrivastava[#3] , Kumar Pranav[#4], K Rohan[#5]

Sambhram Institute of Technology, Bangalore-97, India, solansrinath@gmail.com, +919036646801

Abstract— Web Services provides flexibility and interconnection between different systems, the communication in Web Services uses Simple Object Access Protocol (SOAP) a simple, robust, and extensible protocol that is the most widely used for communication protocol in the Web services model. SOAP message that is being sent to the client needs to be secured from the attackers over the open internet and should also make sure that the response time is minimal. In this paper, the SOAP message is encrypted whole or selective portion using AES (Advanced Encryption Scheme) encryption technique. The SAX parser is used to reduce the memory consumption and execution time for parsing SOAP request, response documents. To optimize the SOAP performance, response time is analyzed for both complete and selective encrypted soap document and results are found satisfactory for selective encryption technique.

.**Keywords**— SOA, Web Services, SOAP, WSDL, AES Encryption, SAX

INTRODUCTION

Changing the way organizations conduct business nowadays, modern technology dominates the business sector entirely. All transactions and managerial activities are being carried out via the Internet increasing application to application communication. Web Services simplify this business interaction by linking its applications with those of its business partners, customers and suppliers via the Internet. A Web Service is an independent self-contained application that can describe, publish and invoked remotely over the internet, thereby allowing smooth interoperability among heterogeneous systems and simplifies the business interaction for the organization

Security continues to be a top concern of Web Services and along with increased information exchange capabilities comes the significant considerations and challenges for the organizations. Security parameters like Authentication, Confidentiality, Integrity and Authorization are of prime concern to these organizations. *Authentication* involves verification of the user's identity based on the credentials presented. *Confidentiality* includes keeping the message safe from external entities while *Integrity* is the non-repudiation of data. *Authorization* allows access to only those users's who are authorized for the particular service [1]. Vulnerability in the application could lead to a breach in the system and could provide attackers with private information or system resources. Information can be credit card numbers, Pan Accounts and Passport leading to disparate complications causing significant damage to organizations. An attack as simple as DoS can cripple the organization's infrastructure and also panic among its customers.

The web services communicate through Simple Object Access Protocol (SOAP) which is robust and most widely used for communication. Since the SOAP messages needs to be secured from attackers various techniques are used for enhancing soap processing. In this paper, the SOAP message is encrypted using complete and selective portion using AES encryption technique. The SAX parser is used to reduce the memory consumption and execution time for parsing SOAP request, response documents.

2. SOA (Service Oriented Architecture) model for web service

Our idea of design is based on the SOA (Service Oriented Architecture) shown in Fig.1, wherein which application components provide services to other components via a communications protocol, typically over a network. The components involved in the model are labelled as Provider, Registry and Client [3]. The Provider creates a web service and publishes its interface on the Registry. The Client then finds the required web service on the Registry using various operations and binds to the provider invoking the required service. The Provider reads in the request from the Client and responds accordingly. SOA has been widely used in the field of web-based applications.

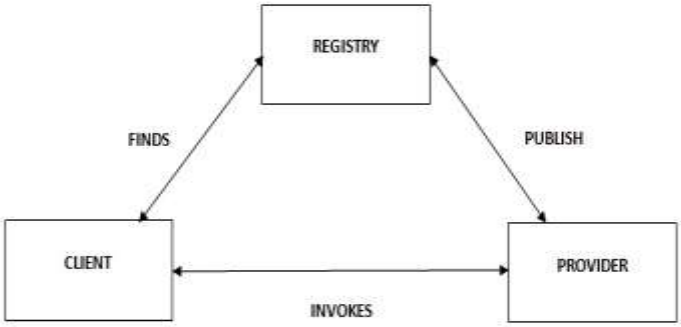

Fig. 1 SOA Architecture

The SOA helps in integrating wide heterogeneous applications by using multiple different platforms and provides a way for the consumers of the service. One of the key aspects of the SOA is that interactions occur with loosely coupled services that operate independently. A service reusability also exists which avoids unnecessary wastage of time and money by starting the development of the service from scratch.

The SOA is thus a valid approach to solve many architectural problems that are faced by the enterprises today. With wide use of web services today SOA is the best way to bring the architectural agility to the enterprise.

3. Proposed model

The proposed model is shown in Fig. 2 which implements the architecture over a primitive outline of SOA model. The provider creates the web service and publishes its WSDL files onto the registry. Client then request for the needed service from the registry. If the service is found the required information about the service is sent to the client that is the API's regarding service invocation and other general information. The client then invokes the service and sends in the SOAP request to the Provider.

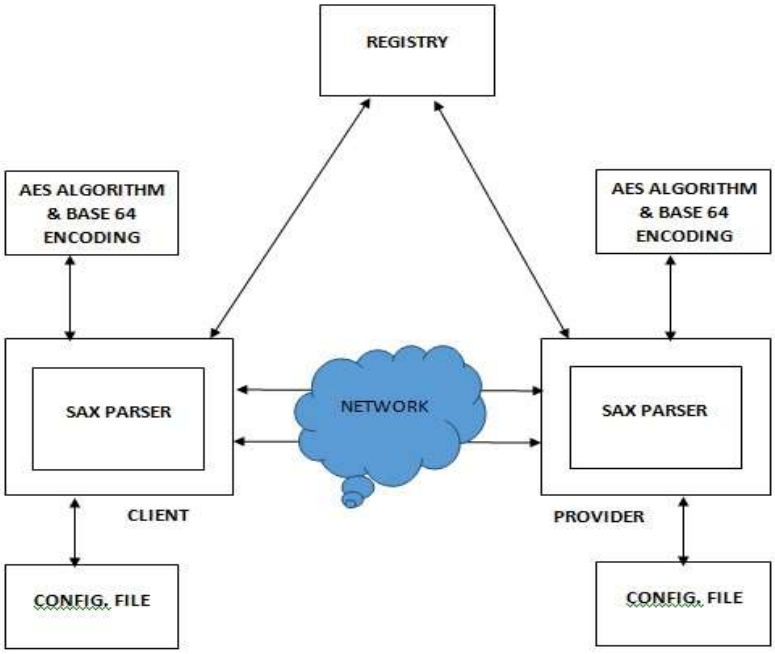

Fig. 2 System Model

The provider then processes this request and formulates an appropriate response and sends back the result to the Client thereby completing an interaction successfully. All the interactions are kept confidential by using AES encryption algorithm.

3.1 Provider Model

The Provider is an application which exposes its service publicly and can be invoked by the requestors of the service. In the proposed provider model shown in Fig. 3 we take the WSDL files of web services and firstly pass them as input to a XML parser. The parsed document is then transferred to the application for further processing. SAX parser is used instead of a DOM parser as SAX is an event based parser which triggers events on occurrence of tags whereas the DOM parser creates a DOM tree of all the tags which incurs extra overhead of memory for traversing each node of the tree and making the encryption process more tedious. The parsed file is encrypted using the AES (Advanced Encryption Scheme) algorithm which provides a considerable amount of security and is stronger than DES as it provides various key lengths ranging from 128-bit, 192-bit or 256-bit than 56-bit respectively [2]. Also AES is used in this context as it is faster than RSA (Rivert Shamir Aldeman) algorithm. A configuration files which helps in selective encryption. This configuration file consists only of those tags which need to be encrypted and is read by the application at the start itself. This successfully results in an encrypted WSDL file[4].

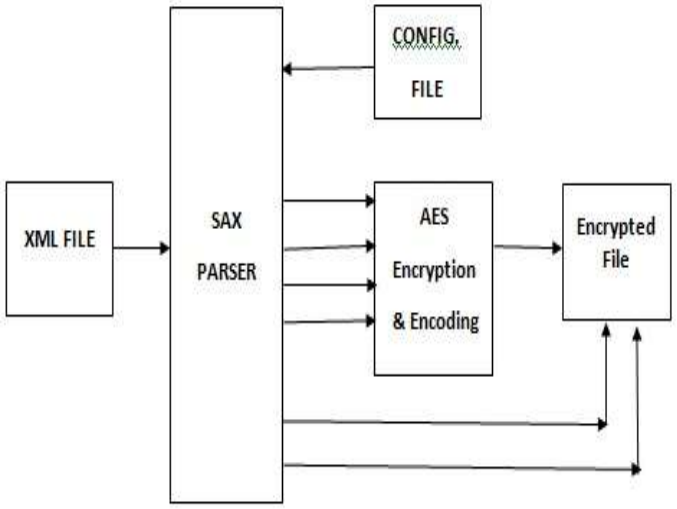

Fig. 3 Provider Model

After creation of Web service and successful encryption of WSDL files, they are published onto the Registry for Clients to find and invoke. We establish a connection from the provider to the Registry and send these encrypted WSDL files. Further the Provider interacts with the Client as the latter requests the service. The Provider processes this SOAP request and provides the suitable SOAP response[5] [6] [7].

3.2 Client model

The Client is a service consumer which invokes the services provided by the Provider. Client finds the required service from the Registry by sending a request to the registry. Fig. 4 shows the how the WSDL file is decrypted when it reaches client. The file is passed through a SAX parser wherein all the encrypted tags are processed by event triggering. Next this file is decoded and decrypted using the BASE64 and AES decryption algorithm. Decryption is done with respect to configuration file which consists of only those tags that are to be decrypted. If all fields are encrypted then it takes more amount of time processing each tag. This way an encrypted WSDL file is successfully searched and retrieved.

The next step of the Client is to establish a connection to the service provider through a network. The Client then sends across a request demanding a particular service via a SOAP envelope. This request is sent via the network using the HTTP protocol. Since the SOAP request is encrypted hence there is no need of using HTTPS, HTTP would suffice. The provider processes this request and then formulates a response, suitably received by the Client.

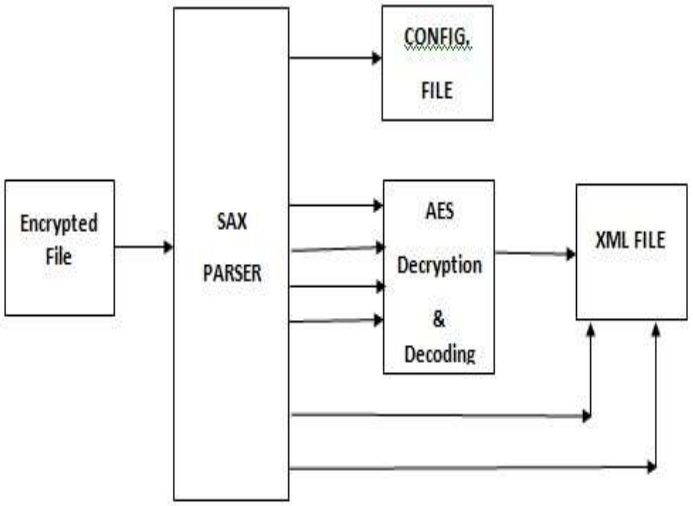

Fig. 4 Client Model

4. RESULTS Analysis

Results are analysed for both complete and selective encryption. In *complete encryption, all the elements of SOAP document is encrypted and encoded* using the AES algorithm and BASE64 encoder. In *selective encryption, the specific elements of SOAP document are encrypted and encoded* with the help of configuration file which indicates which elements should be encrypted; Fig. 5 shows the input SOAP document that is to be encrypted. Fig 6 shows the cipher file obtained after encryption [8][10]. The decryption process is exactly the reverse and we get the original file back using the AES decryption algorithm and BASE64 decoder[9][11][12].

```
▼<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 ▼<soapenv:Body>
  ▼<sub xmlns="http://rohan">
     <a>15</a>
     <b>3</b>
   </sub>
  </soapenv:Body>
</soapenv:Envelope>
```

Fig. 5 SOAP document

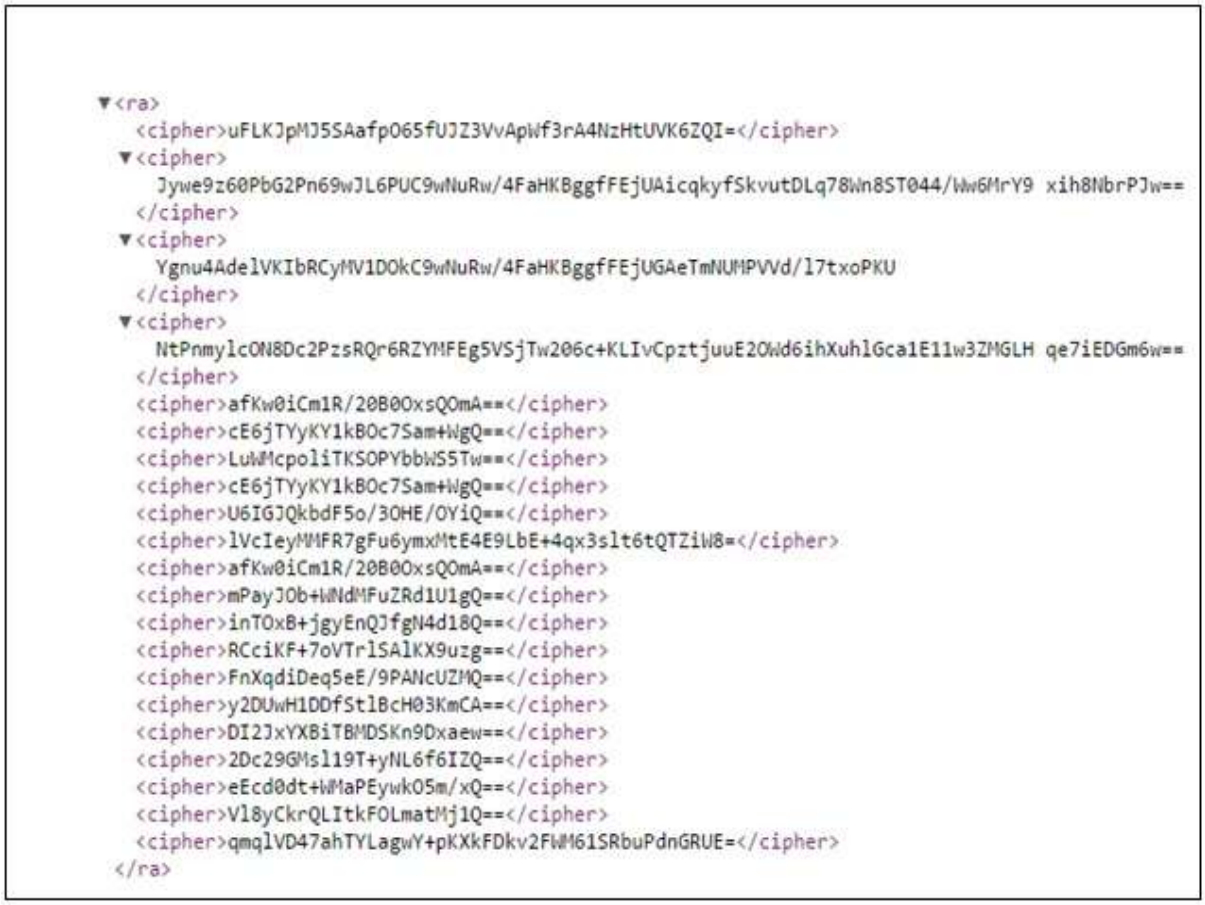

Fig. 6 Cipher file

The time taken for complete and selective encryption and decryption of SOAP document are tabulated for varying the number of tags from 5 tags to 25 tags is shown in Table (i) and Table (ii).

Fig. 7-8-9 shows graphs plotted against tabulated values, from the obtained results it can be seen that both selective and complete encryption takes almost the same time for encrypting and decrypting the XML document.

TABLE VIII

Complete Encryption & Decryption Time (in milliseconds)

Number of Tags	Complete Encryption Time (in milliseconds)	Complete Decryption Time (in milliseconds)
5	273.58	7.13
10	277.63	9.535
15	293.9	12.429
20	306.2	14.893
25	318.82	17.678

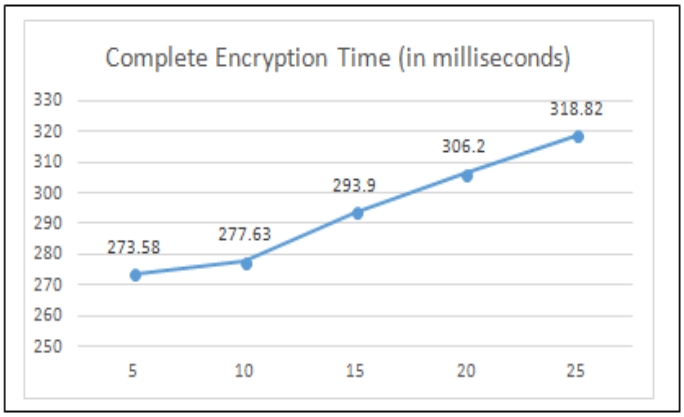

Fig. 7 Complete Encryption Time (in milliseconds)

TABLE II

Selective Encryption & Decryption Time (in milliseconds)

Number of Tags	Selective Encryption Time (in milliseconds)	Selective Decryption Time (in milliseconds)
5	272.32	4.164
10	274.81	4.328
15	288.63	5.615
20	293.25	6.814
25	302.76	7.4

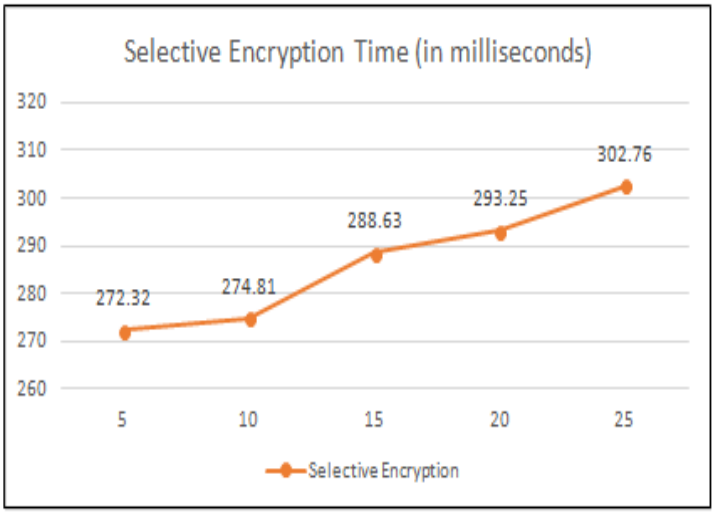

Fig. 8 Selective Encryption Time (in milliseconds)

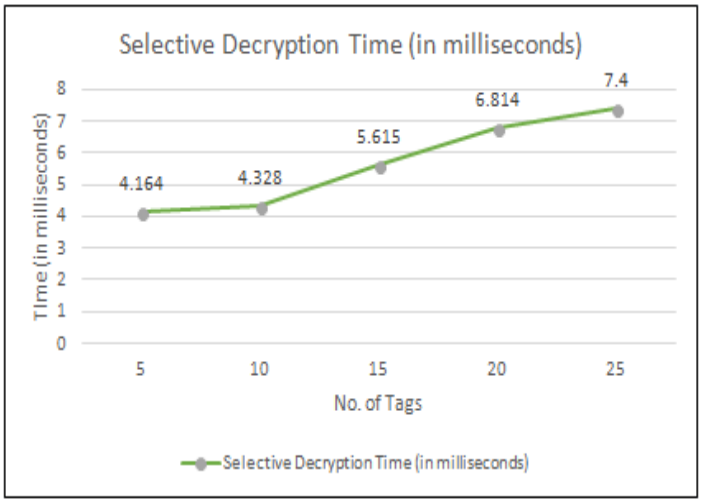

Fig. 9 Selective Decryption Time (in milliseconds)

Axis 1.4 server is integrated on Kepler eclipse IDE, the server side response time was recorded when the client sends SOAP request, Table (iii) and Fig. 10 shows the results obtained.

TABLE III

Selective Encryption & Decryption Time (in milliseconds)

No. of Tags in SOAP	Response Time (in ms)
5	15
10	15
15	16
20	16

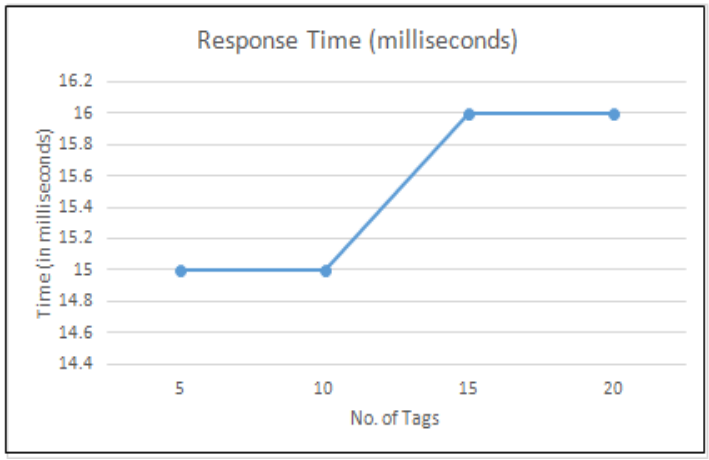

Fig. 10 Response Time from Axis server

5. CONCLUSION

In this paper, streaming-based complete and selective XML encryption and decryption have been designed and implemented. Experiments were conducted to demonstrate the comparison between the response time of selectively encrypted XML document and completely Encrypted XML document. The results shows streaming-based XML encryption and decryption technique provides a better way of resisting XML attacks. It also includes hash code generation for the SOAP request and SOAP response to maintain the integrity of data during transmission over the network. The analysis reveals the use of stream based selective XML encryption and decryption is faster and less memory consumption compare to stream based complete XML encryption and decryption. The SOAP document can further optimised in its header field for faster response time.

REFERENCES:

[1] Esmiralda Moradianvand Anne Håkansson, "Possible attacks on XML Web Services", IJCSNS International Journal of Computer Science and 154 Network Security, VOL.6 No.1B, January 2006, pp 154-170.

[2] Hashizume Keiko, Eduardo B Fernandez, "Symmetric encryption and XML encryption patterns", Proceedings of the 16th Conference on Pattern Languages of Programs, pp.270-298, 2009.

[3] Meiko Jensen, Nils Gruschka, Ralph Herkenhoener and Luttenberger, N., (2007), "SOA and Web Services: New Technologies, New Standards – New Attacks" Fifth European Conference on Web Services, 0-7695-3044-3/07, 2007.

[4] Michael Schrefl, Katharina Grun, Jurgen Dorn, "SemCrypt - Ensuring Privacy of Electronic on Data Engineering Workshops (ICDEW)", 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW) 2005, pp. 1191, doi:10.1109/ICDE.2005.280 Documents Through Semantic-Based Encrypted Query Processing", ICDEW, 2005, 2013 IEEE 29th International Conference.

[5] Taflan I. Gundem and Mustafa F. Celikel, "Structure Encryption in XML", Computer Engineering Dept., Boğaziçi University, 34342 Bebek, İstanbul, Turkey, pp. 1-13, 2008.

[6] Rupesh Kumar, Mario mumoz organero, rajat agarwal, "XML Secure Documents for a Secure E-commerce Architecture", pp. 35-45, 2010.

[7] Benjamin Sanno, "streaming-based encryption technique", pp. 1-48, 2010.

[8] R.D. Cameron, K.S. Herdy, and D. Lin, "PARABIX: High Performance XML Parsing Using Parallel Bit Stream Technology," Proc. Conf. Center for Advanced Studies on Collaborative Research: Meeting of Minds (CASCON '08), vol. 17, pp. 222-235, 2008.

[9] J. Cheney, "Compressing XML with Multiplexed Hierarchical PPM Models," Proc. Data Compression Conf., pp. 163-173, 2001.

[10] R. Chinnici, J.J. Moreau, A. Ryman, and S. Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C recommendation, http://www. w3.org/TR/wsdl20, Aug. 2009.

[11] K. Chiu and W. Lu, "A Compiler-Based Approach to SchemaSpecific XML Parsing," Proc. Workshop High Performance XML Processing, 2004.

[12] K. Chiu, M. Govindaraju, and R. Bramley, "Investigating the Limits of SOAP Performance for Scientific Computing," Proc. ACM Int'l Symp. High Performance Distributed Computing (HPDC), pp. 246-254, 2002