XML ENCRYPTION AND SIGNATURE FOR SECURING WEB SERVICES

Iehab ALRassan

Computer Science department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

ABSTRACT

In this research, we have focused on the most challenging issue that Web Services face, i.e. how to secure their information. Web Services security could be guaranteed by employing security standards, which is the main focus of this search. Every suggested model related to security design should put in the account the securities' objectives; integrity, confidentiality, non- repudiation, authentication, and authorization. The proposed model describes SOAP messages and the way to secure their contents. Due to the reason that SOAP message is the core of the exchanging information in Web Services, this research has developed a security model needed to ensure e-business security. The essence of our model depends on XML encryption and XML signature to encrypt and sign SOAP message. The proposed model looks forward to achieve a high speed of transaction and a strong level of security without jeopardizing the performance of transmission information.

KEYWORDS

Web Services, SOAP, SAML, XKMS, IDEA, RSA.

1. INTRODUCTION

Nowadays, Service Oriented Architecture (SOA) is being used widely between systems as a communication between loosely coupled services, which are functioning independent of the programming languages. Web Service is becoming, to a large extent, indispensable in many businesses. It plays an important role not because that it is a new generation technology, rather because it deals with the requirements of software development. While a variety of definitions of Web Services has been suggested, this paper will use the definition suggested by [1] who relied Web Services on a list of criterions to hold-up interoperability overall applications that have been developed in various languages and are running on various environments or operating systems. Web Services modify the notion of application communication from human-centric, where person takes the basic roles in communication, to application-centric, where emphasis is laid on application communication.

Web Service has been linked to various technologies, like SOAP, UDDL, and WSDL. SOAP message is used to transfer information, UDDI for discovering services, and WSDL for describing the services.

Web Service depends mainly on SOAP (Simple Object Access Protocol) for transmitting information among components. The specification of SOAP protocol could not provide any security. Thus, transmitting sensitive information by SOAP message may be intercepted and changed/modified via eavesdroppers. So, Web Service application is vulnerable to diverse

DOI: 10.5121/ijcsit.2020.12402

attacks. The security of Web Services depends basically on exchanging messages based on SOAP protocol. It employs SOAP protocol as a basic method to transmit XML information. Consequently, SAOP protocol provides support for the project that implemented applications of Web Services by ensuring that these applications are accessible to different e-business applications. The difficulty of the systems of Web Services is discovering an appropriate methodology to be compliant of the security needs of XML messages. For instance, if systems depend on SSL (Secure Socket Layer) only, they will not offer adequate security since SSL cannot achieve end-to-end security. SSL, regarding the security techniques, provides secured transport layer related to the Web Service, but Web Services security provides an upper degree of abstraction. A SOAP message with high level of security is an element of the foremost aims of Web Services security [2].

Web Services Security is a fundamental entity of the protocol stack of Web Services to assure end-to-end authentication capabilities, integrity, and confidentiality to Web Services among standards of XML. Security of end-to-end message contends the involvement of transport channel that is unsecure in message exchanges, which is a significant feature for service-oriented architectures and web systems. On the other hand, protocols of point-to-point security, e.g. Transport Layer Security, assure only bounded choices of security for Web Services. The point-to-point restriction is not constantly bad option for elementary architectures of services that demand protecting contents of message against alteration. For example, when two entities interact through HTTPS, keeping the security of service-oriented architectures necessitates much intermediary processing entity that shows the issue of an unreliable “man-in-the-middle” entity.
No new technique is offered by Web Services Security (WS-Security). It is just a grouping of current criterions like XML encryption, and XML signature [3]. Signature and encryption of XML document are methods to sign and encrypt the entire or a section of SAOP message. XML signature provides integrity of message and ensures that message exchanged is not modified or intercepted, and XML encryption ensures the message confidentiality. Encryption algorithms could be divided into two basic techniques: symmetric technique and asymmetric technique. In a symmetric cryptography, there is just one key called secret key that both sender and receiver use, and an asymmetric cryptography includes two keys: private and public key. The issue of a symmetric key is the key distribution, while an asymmetric key recovers such problem.

In this research work, we aim to propose a model to secure Web Services by XML encryption and decryption. The security model considers both levels of security; point-to-point security, and end-to-end security. To achieve point-to-point security, our paper used HTTPS, while in order to achieve the message level security, we relied on encryption and signature of XML file to support integrity and confidentiality. Also, XML signature provides non-repudiation [4].

The rest of the paper is structured as follows: Section 2 provides general review of Web Services with their components. Section 3 describes the overview of standards of security related to Web Services besides elaborating XML encryption, XML signature, XKMS, and SAML. Section 4 explains our proposed security model that depends on current methodologies to enhance security of Web Services in many fields. Section 5 summarizes the results, while section 6 concludes the research and illustrates the future work.

2. WEB SERVICES

Advancement in internet has revolutionized the way business is done today. In large businesses, the business processes need to be managed and integrated with other systems in an effective and befitting manner. The Business Process Management (BPM) aims at integrating and automating different systems in a business. The BPM of the enterprises needs to be improved in order to maintain reliable Business-to-Business partnerships. Web Services might seem to be a concept related to only technical IT operations. However, Web Services paradigm is proving to be very promising in the field of e-business. Web Services provide de-facto standards for computation and communication, hence e-business can benefit tremendously from it by linking and communicating its applications and services with customers, partners, and/or suppliers etc. "Web Services are self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces" [5] . Different techniques such as SOAP, UDD, and WSDL are linked to the Web Services. Figure 1 represents Web Services' roles and operations that comprise of the interaction among three basic roles: service registry, service provider, and service requestor. Usually, these three roles communicate with every other by the operations of publish, find, and bind. Typically, a provider of service is in charge of implementation of a Web Service or hosting a network module that could be accessed. The provider of the service then clarifies a description of the service and publishes it for the registry of service or requestor of the service. The requestor of the service employs the Find Operations to search or access locally the description of service or from registry of service and then employs this description related to the service to allow binding with the provider of service. After that, the service requestor interacts with the Web Service implementation module. A service is capable of taking up the roles of both service requestor and the service provider [6].

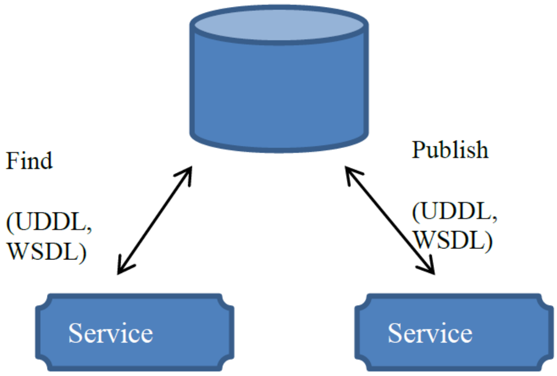

Figure 1. Web Services roles and interactions

2.1. XML Web Services

XML messaging is the most fundamental entity of Web Services architecture. XML has become a standard way to represent structured data. It has even been adopted as a message format in the form of SOAP. There are clear benefits of adopting SOAP for the messaging component of a middleware platform. SOAP is a protocol that allows exchanging information by XML messaging. The important thing is that it is very straightforward, offering not many conventions on the way to arrangement body and headers inside a message with XML format. SOAP is termed as transport independent, meaning that the messages of SOAP protocol could be post by any transport protocols that do not depend on any reason. SOAP is a straightforward XML-based procedure, which is used to interact between network applications by exchanging structured data between them [7].

2.2. Simple Object Access Protocol (SOAP)

SOAP is a simple protocol, which allows communications through XML Web Services. The simplicity of SOAP and the ubiquity of HTTP let them to be a prefect foundation for building XML Web Services, which could be requested by usually any platforms. The greatest feature of

SOAP could also be a negative side. Data included in SOAP message is transmitted as text in XML file to support data representation with standardized style. Changing whole data into text and then converting this text back to original data structures in the time they reach the other point could consume little of power for transformation [8].

3. WEB SERVICES SECURITY

Formal definition of web security states that "It is a set of procedures, practices, and technologies for protecting web servers, web users, and their surrounding organizations. Security protects you against unexpected behavior".

There are six security requirements, which need to be met for information security in general as well as Web Services Security in particular. These requirements include confidentiality, integrity, non-repudiation, authentication, authorization, and availability.

3.1. The Need for Security of Web Services

End-to-End: Providing the confidentiality and integrity of the message during transmission could be achieved by using secure transport protocols such as SSL and IPSec, but their security is for point-to-point only.

On the other side, since intermediaries receive messages of SOAP and forward, even if there is reliability among the communication links between them, secure interaction between end-to-end points is impossible when there is no confidence corporation through the complete intermediaries. End-to-end security is also compromised if the communication link is not secured. By accurate focusing at topologies of the Web Services, assurance security of transports is not adequate for end-to-end security of the XML SOAP message.

Independence of Middleware: The only method to support end-to-end security is at the level of the application or middleware. When the elements exchange message in the form of plain text, it could be a sensitive chance of attack. Integrating cryptographic functionality into a new or running application is not a simple or coveted function without admission of too much vulnerability of security and growing number of risks. In most circumstances, it is imperative to develop security implementation as near to the application as we can.

Transport Independence: There are many employments of SOAP intermediaries; one of them is to resend messages (forward) to various elements in the networks. Security concepts, like the authenticity of the message generator, have to be interpreted to the following domain of security related to the transport protocol through the path of message, which would be tedious, and complicated, and would direct to faults related to integrity.

Asynchronous Messages of Multi-hop: Security of transport layer assures the information if it is moving on links of communication. It does not do anything with stocked data on intermediary point. Security of transport layer could not support efficiently in securing the information from forbidden accesses and sensitive modifications after a message is received and decrypted. In cases if messages are kept and then resent (continuous message queues), protection of message layer is required [9].

3.2. XML Key Management Services (XKMS)

XML Key Management Services (XKMS) is a trusted PKI service. It provides a trust relationship between clients. It is an XML interface for underlying PKI as illustrated in figure 2. Following are the benefits of XKMS:

- It has a small client footprint, implying that it occupies less space.
- Since it is based on XML, its implementation is simplified.
- It develops trust relationships between enterprises

New PKI features can be deployed without the requirement of explicitly deploying new clients [10].

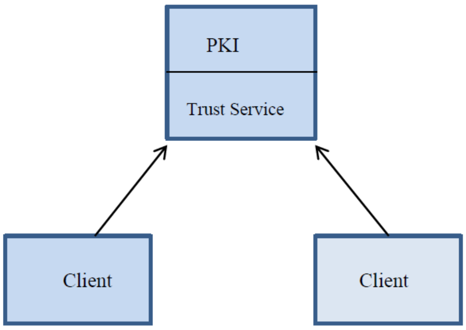

Figure 2. XKMS shields clients from PKI complexity

3.3. Security Assertion Mark-up Language

Security Assertion Mark-up Language (SAML) is a framework used for authorization and authentication of the request-response exchanges between applications. It is an XML based framework supporting the exchanges, such as the interaction between applications, which do not have common similar underlying infrastructure of authentication and authorization. These variations in infrastructure might be platform-based or organizational (such as Mac versus Windows). In these scenarios, SAML could be employed to ensure a Single Sign-On (SSO) among various platforms and systems as illustrated in figure 3 [11].

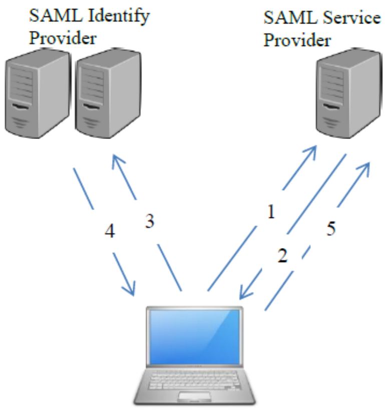

Figure 3. The process for service provider initiated single sign-on

3.4. XML Encryption

XML Encryption technology is required for the confidentiality of XML data. Its benefits include partial encryption, multiple encryptions, and complex encryption. Partial encryption deals with the encryption of certain tags of the XML documents, leaving the rest. Multiple encryption deals with encryption of data multiple times, whereas the complex encryption deals with activities like designation of recipients who are allowed permission to decrypt portions of data. XML encryption deals with confidentiality aspects of data security, such as eavesdropping [12].

3.5. XML Signature

Digital signatures in XML documents are used as a proof of a document's data integrity. The syntax of XML signature and specification of processing contain syntax of XML based signature for symbolizing associations among signatures of cryptographic and XML files. The specification contains functions for verifying and calculating the XML signatures. It is flexible to be applied on any type of data. Signature might be used for the whole XML tree, specific elements of the message, parts of elements, or arbitrary multiple XML tree parts.

XML digital signature could be done by using asymmetric encryption (private key and public key).To sign a message m, apply the encryption function with the private key to produce a signature s.To verify, apply the encryption function with the public key to the signature. Then, check that the result equals the expected message [13] [14].

$$\text{Sign}\ (m,k) = R(m,k)$$

$$\text{Ver}\ (m,s,K) = R(s;K) = = m$$

4. PROPOSED SECURITY MODEL

Design of security model related to Web Services poses a wide range of challenges, especially in the security perspective. Designers need to keep in mind certain requirements, like data confidentiality, data integrity, data authorization, and data authentication. Some of these are unique to Service Oriented Architecture (SOA), while others are common for all sorts of web application scenarios.

Figure 4 shows the proposed design that focuses on the security aspect of SOAP messages, which are the main means of information, upon which a web service relies. For securing SOAP messages, the proposed design considers end-to-end and point-to-point security. The main focus in this design model is on the security goals of Web Services. In this scenario, a hybrid encryption scheme has been used for XML encryption. A hybrid encryption uses public-key cryptography for secret key distribution and secret-key cryptography for data transmission. We have used this scheme because it guarantees the speed of secret key encryption and high security with public key encryption during transmission. If we use just symmetric key encryption, then there is a need to secure secret key from eavesdropper during transmission. On the other side, if we use asymmetric key encryption, it will cost high processing power. So, we will be using both methodologies to get the most benefit from both.

For encryption secret key, we choose RSA since it is so efficient and demand less time than AES or ECC [15]. RSA depends upon presupposing hardness of discovering factoring huge number. Our proposed model focuses mainly on the security performance of application of Web Services. Regarding encryption of XML data, we select IDEA since many papers concluded that IDEA is fast and ensures high security as compared to other encryption algorithms [16]. Internet and web have made the entire world come together [17].

In the proposed methodology, XML encryption and XML signature have been used for addressing the integrity and confidentiality issues of Web Services respectively. “XML Key Management Specification (XKMS)” is a specification used for easy management of the security infrastructure of a web service application. We chose XKMS for the following reasons:

- It works with the Web Services framework to let it easier for designers and programmers to secure transmissions between applications using public key infrastructure (PKI).
- It has a small client footprint, implying that it occupies less space.
- Since it is based on XML, its implementation is simplified.
- It develops trust relationships between enterprises.
- New PKI features can be deployed without the requirement of explicitly deploying new clients [12].

4.1. Requestor Scenario

Summary: The requestor wants to request the services from the provider by encrypting the SOAP message using a secret key, which is enciphered by public key encryption and is sent over HTTPS.

Actors: A requestor

Precondition: Requestor has registered the public key using the XKMS server.

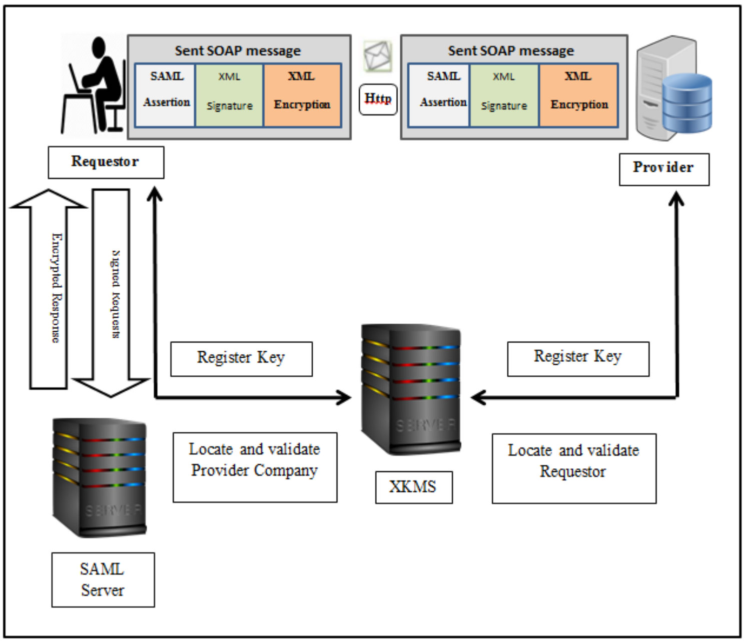

Figure 4. Proposed design model

Description:

a) Requestor logs in to the system using SAML server for a single sign-on.

b) The authority will receive a request from the requestor and will authenticate the request with a document containing SAML assertion.

c) The authority returns the SOAP message to the requestor.

d) The requestor will sign the SAML request using XML signature via RSA and send it to SAML server.

e) SAML server verifies the identity of the requestor. SAML response is encrypted and sent to the requestor.

f) Requestor will decrypt the SAML response. SAML response consists of authentication, attribute, and authorization. To avoid the reply attacks, attribute security will be contained in the header of SOAP message.

g) SOAP body is encrypted using XML encryption so that confidentiality can be ensured. Hybrid encryption algorithm will be used for this. For more details, RSA will be used for secret key encryption, and IDEA will be used for XML message encryption. SOAP message is signed using XML signature via RSA.

h) Signed and encrypted message is transmitted through HTTPS.

Post-condition: Encrypted XML message and encrypted secret key have been sent.

4.2. Provider Scenario:

Summary: The provider receives a request from the requestor, checks its validity, and decrypts the encrypted XML message.

Precondition: Provider has registered the public key using the XKMS server.

Description:

a) Provider company checks the validity of the SOAP message received. If the message has valid security attributes, it is then responded.

b) Signature of the SOAP message is verified.

c) SOAP message is decrypted using the cipher key.

d) SAML assertion is processed for ensuring the identity of the user.

e) Decision is made for response or denial of the message.

Post-Condition: The message is decrypted and responded.

5. Experimental Results

Finally, in our proposed model, we achieved authentication by using SAML, XKMS, and XML signature. The goal of SOAP message integrity has been reached via XML signature. Also, we worked for getting the aim of SOAP message integrity by employing SSL, however it just functioned once the message is in channel during transition. By XML encryption, we achieved message confidentiality. For non-repudiation goal, we achieved it by using XML signature.

The implementation phase of the system takes into account the design aspects of the system previously defined. Keeping that in mind, we have developed a bank transaction client-server application. This application mainly focuses on the security aspect of the SOAP messages, with additional emphasis on transport security. Two encryption techniques i.e. RSA and Elgamal encryption have been used and their performance is compared at the end through experimental results. SAML assertion has been used in order to authorize and authenticate user credentials through Security Token Service (STS). We have used Windows Communication Foundation (WCF) service for service-end implementation and ASP.NET web forms for client-side

development with Microsoft Visual Studio 2010 as a platform. In addition to that, we have used WCF tracing tool for tracing the XML, and Windows Identity Foundation for facilitation in SAML authentication.

5.1. RSA key Exchange

Table 1. shows the time of encryption in the case of RSA exchange algorithm, which was measured in the case of all four methods that were requested from the service i.e. GetName(), GetDate(), GetNum(), and GetResp(). The time of end-to-end encryption is the total time including message encryption time, sending time, receiving time, and decryption time. In case of RSA key exchange algorithm, the end-to-end encryption time of the GetName() method is 11:38:31.30 to 11:38:32.73 (1.43 seconds), the end-to-end encryption time of the GetDate() method is 11:38:32.56 to 11:38:32.74 (0.18 seconds), the end-to-end encryption time of the GetNum() method is 11:38:32.61 to 11:38:32.76 (0.15 seconds), and the end-to-end encryption time of the GetResp () method is 11:38:32.736 to 11:38:32.774 (0.04 seconds).

The calculated time of sending, time of receiving and time of decryption of the four WCF methods i.e. GetName(), GetDate(), GetNum() and GetResp() with the RSA key exchange method. The total processing time (Sending + Receiving + Decryption) = 1.474 seconds.

Table 1. RSA key exchange Time

Method	Encryption Time	Sending Time	Receiving Time	Decryption Time	Total Duration
GetName()	11:38:31.30 to 11:38:31.35	11:38:31.36	11:38:32.48	11:38:32.63 to 11:38:32.73	1.43 seconds
GetDate()	11:38:32.56 to 11:38:32.73	11:38:32.738	11:38:32.740	11:38:32.740 to 11:38:32.743	0.183 seconds
GetNum()	11:38:32.61 to 11:38:32.74	11:38:32.747	11:38:32.763	11:38:32.763 to 11:38:32.765	0.15 seconds
GetResp()	11:38:32.736 to 11:38:32.769	11:38:32.769	11:38:32.770	11:38:32.771 to 11:38:32.774	0.04 seconds

5.2. Diffie-Hellman Key Exchange

The time of encryption in the case of Diffie-Hellman exchange algorithm was measured in the case of all four methods that were requested from the service i.e. GetName(), GetDate(), GetNum() and GetResp().

Table 2. shows the time of sending, time of receiving, and time of decryption of the four WCF methods i.e. GetName(), GetDate(), GetNum() and GetResp() with the Diffie Hellman key exchange method. The total processing time (Sending + Receiving + Decryption) = 0.44 seconds.

Table 2. Diffie-Hellman key exchange

Method	Encryption Time	Sending Time	Receiving Time	Decryption Time	Duration
GetName()	11:17:35.22 to 11:17:35.30	11:17:35.53	11:17:35.54	11:17:35.55 to 11:17:35.63	0.41 seconds
GetDate()	11:17:35.53 to 11:17:35.63	11:17:35.642	11:17:35.643	11:17:35.644 to 11:17:35.646	0.116 seconds
GetNum()	11:17:35.55 to 11:17:35.64	11:17:35.649	11:17:35.65	11:17:35.651 to 11:17:35.653	0.103 seconds
GetResp()	11:17:35.57 to 11:17:35.65	11:17:35.656	11:17:35.657	11:17:35.658 to 11:17:35.660	0.09 seconds

5.3. RSA Encryption vs. Decryption Comparison

Figure 5 shows the comparison between encryption and decryption of RSA key exchange algorithm. We have to keep in mind that this is the performance of the same algorithm, but different functions i.e. encryption and decryption. All the charts generated consist of time in seconds on the y-axis and the service methods on the x-axis. It can be seen from figure 2 that for RSA, decryption is faster than encryption as encryption takes relatively more time as compared to decryption for three out of the four methods.

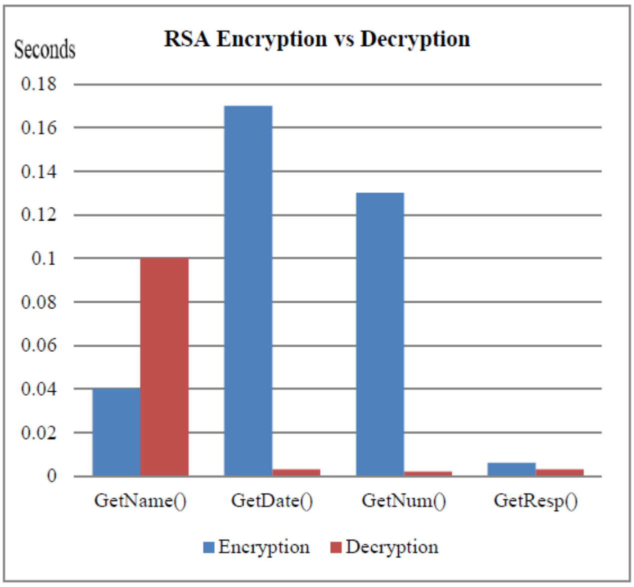

Figure 5. RSA Encryption vs. Decryption

5.4. Diffie-Hellman Encryption vs. Decryption

Figure 6 shows the comparison between encryption and decryption of Diffie-Hellman. The encryption process is relatively much slower than the decryption process. We can see from the figure 6 that encryption and decryption processes take the equal amount of time only for the GetName() method. For the other three methods, encryption takes relatively more time as compared to decryption by a visible margin.

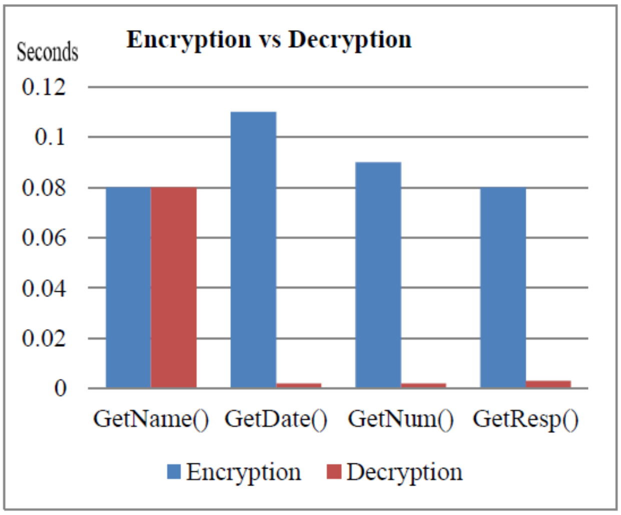

Figure 6. Diffie-Hellman Encryption vs. Decryption

5.5. RSA vs. Diffie-Hellman Encryption

Figure 7 shows the comparison between the performances of RSA encryption vs. Diffie-Hellman encryption. For two methods, the Diffie-Hellman encryption is faster than the RSA encryption and vice versa. However, if we take a look at the complete duration of encryption including all the four methods, the time taken by RSA turns out to be 0.346 seconds, while that of Diffie-Hellman exchange turns out to be 0.36 seconds, which makes RSA encryption process slightly faster.

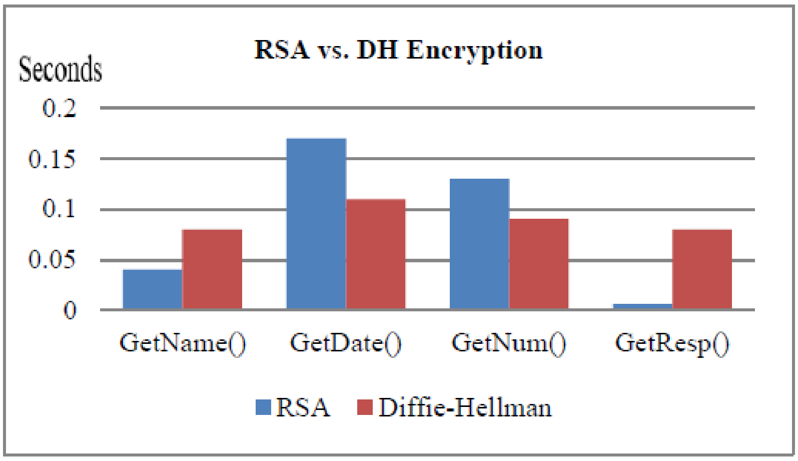

Figure 7. RSA vs. Diffie-Hellman Encryption

5.6. RSA vs. Diffie Hellman Decryption

Contrary to the comparison in encryption, figure 8 shows the comparison between the performances of RSA decryption vs. Diffie-Hellman decryption, which reveals slightly different results. As seen from the figure 8, the Diffie-Hellman algorithm turns out to be more time-efficient for all four cases. In addition to that, if we look at the complete duration of decryption including all the four methods, the time taken by RSA turns out to be 0.108 seconds, while that of Diffie-Hellman exchange turns out to be 0.086 seconds, which makes Diffie-Hellman decryption process slightly faster.

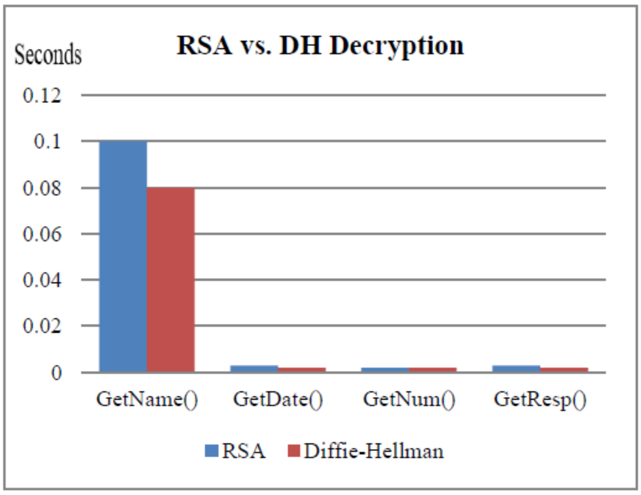

Figure 8. RSA vs. Diffie Hellman Decryption

5.7. RSA vs. Diffie-Hellman End-to-End Encryption

Figure 9 shows the overall performance comparison chart between the RSA and Diffie-Hellman key exchange method, which is the end-to-end encryption chart.

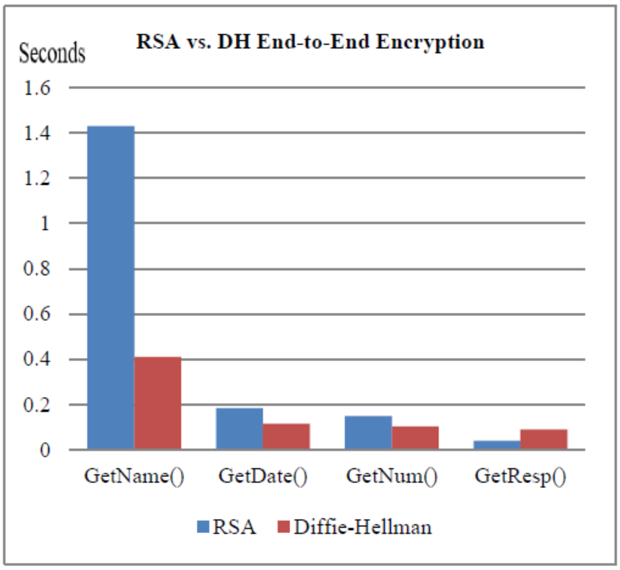

Figure 9. RSA vs. Diffie-Hellman End-to-End

Figure 9 shows that there is a slight variation between performances of the two algorithms for four methods. We can see that the RSA key exchange algorithm takes more end-to-end encryption time in three methods out of four. In addition to that, if we take a look at the overall end-to-end encryption considering all four methods, RSA key exchange algorithm takes more time (1.803 seconds) than the Diffie-Hellman key exchange algorithm (0.719). This relatively stark difference in time reveals that Diffie-Hellman key exchange algorithm shows better performance in terms of time for the end-to-end encryption. However, the difference in performance is mainly in the duration of message sending and receiving i.e. time for the encryption of the initial time request. The message procession (including decryption) and its comparison show a varying result as seen in encryption and decryption comparison. When we take a look at the encryption part, which is the main function of the whole process, RSA key exchange algorithm, in this part, shows better performance than Diffie-Hellman key exchange algorithm although Diffie-Hellman key exchange algorithm is computationally more advanced. However, its computational advancement gives rise to slower encryption process as compared to RSA key exchange algorithm. In case of decryption, Diffie-Hellman key exchange algorithm consumes less time than RSA key exchange algorithm.

6. CONCLUSION

The security is an important field in Web Services. With Web Services, we can get interoperability of e-business, which can support multiple platforms. There are various standards to ensure security for Web Services. XML encryption supports encrypting portion or the entire message and proves confidentiality of the message. XML signature provides the opportunity to sign the entire or part of message. SAML offers assertion, which includes user information security. Employing SAML provides authorization and single sign-on. XKMS is a Web Services that manages keys of sender and receiver and ensures authentication and authorization.

Every suggested model should support the security objectives; confidentiality, integrity, authorization, authentication, and non- repudiation. In our model, we have proposed HTTPS that has been used to accomplish point-to-point security. XML signature and XML encryption are used to support message level security. The hybrid cryptosystem algorithm is used since it gets the most benefit from secret key and public key algorithms. SAML assertion is appended to SOAP header in order to prevent reply attacks; it contains sequence number and timestamp. XKMS is used to achieve authorization and authentication of sender and receiver.

The proposed model achieved high speed of transaction and strong level of security without jeopardizing the performance of transmission information. The experimental results show that the Diffie-Hellman gets better performance in terms of time for the end-to-end encryption. When we take a look at the encryption part, which is the main function of the whole process, RSA in this part, shows better performance than Diffie-Hellman algorithm. In case of decryption, Diffie-Hellman consumes less time than RSA algorithm.

In the future, compression technique will be applied to SOAP message since SOAP message has bigger size. This compression model will be applied after the proposed Web Services security model. We expect that the proposed compression model accompanied by the security is going to be an efficient solution with good performance.

REFERENCES

[1] Minder Chen, Andrew N. K. Chen, Benjamin B. M. Shao, "The Implications and Impacts of Web Services to Electronic Commerce Research and Practices" , Journal of Electronic Commerce Research, VOL. 4, NO. 4, pp. 128-129, 2003.

[2] N. A. Nordbotten, "XML and Web Services Security Standards," Communications Surveys & Tutorials, IEEE, vol. 11 ,pp. 4-21, 2009.

[3] Iehab Alrassan , Maha Alrashed , " Enhancing Web Services Security in e-business " , The International Journal of Soft Computing and Software Engineering [JSCSE], vol. 3 , pp. 502-506 , 2013

[4] Gu Yue-sheng, Zhang Bao-jian, Xu Wu , "Research and Realization of Web Services Security Based on XML Signature" , International Conference on Networking and Digital Society, 2009, pp. 116-118.

[5] Nils Agne Nordbotten, "XML and Web Services Security Standards" , IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 11, NO. 3, THIRD QUARTER 2009

[6] F. Prevention and P. Technologies, “XML Signature / Encryption —,” vol. 2, no. 1, pp. 35–39, 2002.

[7] E. Bertino and E. Ferrari, “Secure and selective dissemination of XML documents,” ACM Trans. Inf. Syst. Secur., vol. 5, no. 3, pp. 290–331, 2002.

[8] T. Imamura and A. Clark, “A Stream-based Implementation of XML Encryption,” Architecture, pp. 11–17, 2002.

[9] Heather Kreger, “Web Services Conceptual Architecture”, IBM, May 2001.

[10] M. Humphrey, M. R. Thompson, and K. R. Jackson, “Security for Grids,” Proc. IEEE, vol. 93, no. 3, pp. 644–652, 2005.

[11] F. Leymann, D. Roller and M. Schmidt, 'Web services and business process management', IBM Syst. J., vol. 41, no. 2, pp. 198-211, 2012.

[12] Web Services Security', Network Security, vol. 2003, no. 5, pp. 14-16, 2003.

[13] M. Chen, “Factors affecting the adoption and diffusion of XML and web services standards for e-business systems,” International Journal of Human-Computer Studies, vol. 58, no. 3, pp. 259–279, 2013.

[14] Web Services Security', Network Security, vol. 2003, no. 5, pp. 14-16, 2003.

[15] B. Kaliski, “The Mathematics of the RSA Public-Key Cryptosystem,” 1989.

[16] Chen, D. Xue and X. Lai, 'An analysis of international data encryption algorithm(IDEA) security against differential cryptanalysis', Wuhan University Journal of Natural Sciences, vol. 13, no. 6, pp. 697-701, 2008.

[17] C. Sireesha , G. Jyostna , P. Varan , and P. Eswari "PROP - Patronage of PHP Web Applications ", International Journal of Computer Science & Information Technology (IJCSIT) Vol 7, No 2, April 2015