A DATA ENCRYPTION STANDARD (DES) BASED WEB SERVICES SECURITY ARCHITECTURE

Mosadoluwa N. Daodu, Arome Junior Gabriel, Boniface Kayode Alese, Adebayo Adetunmbi

Department of Computer Science, The Federal University of Technology, Akure Ondo State, Nigeria

Corresponding author: Arome Junior Arome, ajgabriel@futa.edu.ng

ABSTRACT: Web Service Security (WSS) provides message level protection between two ends of clients and web services. This work proposes a ***WSS architecture*** whose security will be based on ***Data Encryption Standard (DES)***. It models a web service security based on DES, and evaluates the performance of the WSS using some ***standard metrics***. To enable evaluation of the system, experiments were conducted in a ***Windows Vista Operating System environment*** using relevant tools. The evaluation result revealed that, deploying security alongside web services comes with additional overheads like ***Extra Central Processing Unit (XCPU)*** Time Cost for the message that is encrypted and to be transmitted, the server CPU Time to Process Request with Encryption (SCTPWE) increases along with the Request/Respond Time with Encryption (TRRWE), the Server CPU Time to Process Request with Encryption (SCTPWE) is greater than the Server CPU Time to Process Request without Encryption (SCTPWOE).
KEYWORDS: Web Services, Web Services Security, Web Commerce, Information Security, Cryptography, Encryption.

1. INTRODUCTION

Web services are essentially changing the policies governing Web commerce. Today, we are witnessing better and more productive business transactions. This majorly is as a result of Web services connecting together programs at different remote locations across the globe, as well as ensuring cheaper and more efficient transportation of large amounts of data across enterprise networks.

Exchange of data between applications and/or nodes across networks are greatly facilitated by Web services. In fact [Eri00] opined that Web services enable program-to-program communication, integrating and interconnecting applications at diverse Internet locations as if they were part of a single, large Information Technology (IT) system.

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It is based on some common protocols like:

(a). Extensible Markup Language (XML), which include the Simple Object Access protocol (SOAP),

(b). Web Services Description Language (WSDL), which is an interface described in a machine-processable format, and

(c). Universal Description, Discovery and Integration (UDDI).

The three above are shown in the figure below:

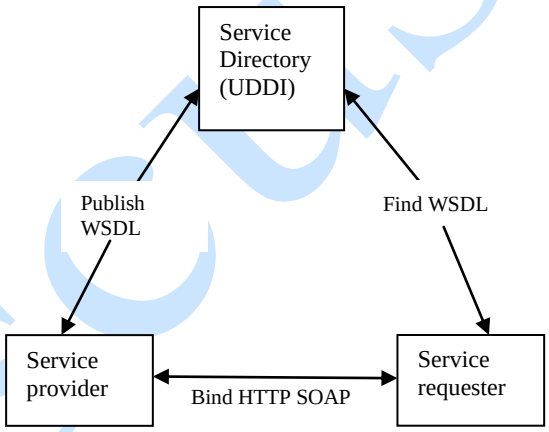

Figure 1: Web Service Architecture (WSA)

These protocols are not dependent on the machine architecture, the underlying operating system or even the programming language. Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, usually conveyed by means of HTTP, an XML serialization and other Web-related standards. A Web service provides a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service.

Existing researches shows that IT has been used to provide different electronic solutions in education, health ([Gar11]), governance ([RK11]), shopping ([AAH06]), medicine ([Gar11]) and even in democratic decision making ([O+13]).

Although it is a good idea to apply Web services to these systems, the security (privacy, confidentiality and integrity) of information (data) while in storage and even in transit, must be given serious consideration, especially if the data in question are sensitive. There is a serious need to include a security module into the Web Service Architecture. This would go a long way in preventing attacks such as spoofing, eavesdropping, falsification and/or repudiation.

The Transport Layer Security (TLS) is a simple and generally acceptable means of ensuring secure transactions for the Web security. However, TLS is originally intended for authenticating the server hosting the Web service; there is no means to authenticate a single service or sets of services running on the same machine. Consequently TLS can not sufficiently meet the security requirements of the Web Services system which are more intricate than Web applications.

Web services offer a standard approach to interoperating different software applications, running on diverse platforms and/or frameworks. Figure 1 shows a Web Services Architecture (WSA). This architecture provides an abstract model and a background for understanding Web services as well as the relationships between the components of this model. The WSA describes both the minimal characteristics that are common to all Web services, and a number of characteristics that are needed by many, but not all, Web services. WSA is an *interoperability* architecture that identifies those global elements of the global Web services network that are required for guaranteeing interoperability between Web services ([BHM04]).

A Web Service can be defined as any service that is available over the Internet, using a standardized Extensible Markup Language (XML) messaging system, it is not tied to any operating system or programming language and it is self-describing via a common XML grammar [Hon10].

Web services enable application integration and data sharing on a neutral platform, language independent environment for both business and science. This increases the degree of exposure of critical resources which poses new challenges to securing data and service.

The Web services use XML (Extensible Markup Language) to pack data into XML messages defined by SOAP message, called WSDL (Web Services Description Language). With web services, applications owned by different organizations can be easily integrated; even if they are developed in different programming languages and deployed on different platforms (Middleware/OS). As a result, web services have been widely adopted in the industry as a standard platform independent middleware technology [C+07].

In this work, Web Service Security requirements were studied from different literatures and a security mechanism, Data Encryption Standard (DES) based security architecture was proposed for the Web Services.

2. GENERAL INFORMATION

Motivation

Many research papers on Web Service Security have been published by researchers all over the world. However, the security issue of the Web Service has often been considered as a crucial barrier to its application in many fields that conducts sensitive information, such as e-commerce [Hon10].

Booth et al., ([BHM04]) discovered that Web Service itself does not provide secure transmission protocol for messages; it brings high risks to both sides of the message exchanger. Although, traditional security technologies such as Secure Socket Layer (SSL) and Hypertext Transfer Protocol (HTTPS) can partially resolve this problem by encrypting messages transferred between two points, these point-to-point transport layer security technologies cannot insure end-to-end security along the entire path from client to a web service in a complicated multi-tiers distributed system. Furthermore, these point-to-point security technologies are all based on a specific transport protocol/layer, such as Transport Control Protocol/Internet Protocol (TCP/IP) for Secure Socket Layer (SSL) and Hypertext Transfer Protocol (HTTP) for HTTPS.

[Oas04] developed Web Services Security in other to provide message level protection between two ends (client and web service) through message integrity, message confidentiality and message authentications. Web Services Security (WSS) makes use of SOAP's composable and extendable architecture by embedding security related information (such as, security token and signature) in the SOAP header without affecting the data stored in the SOAP's body but may be encrypted or signed.

WSS enhances the security of web services but it has its own performance overheads. The overheads are:

(a). extra CPU times to process WSS related elements

(b). longer networking times to transport larger SOAP messages due to additional WSS contents.

[Hon10] introduced the Security Token Service into an existing WS-Security and present a Security Token Service based Security architecture, named STS-based Secure Web Services (STS-WS) for Web Services for higher security and higher performance services.

Chen et al., [C+07] carried out the performance evaluation on Web Service Security using XML Security technologies in other to evaluate the performance cost of applying Web Service Security and also developed a simple performance model for Ws-security.

So far, WSS enhances the security of Web services but it has its own performance overheads which is the extra CPU times to process WSS related elements. Hence, in this course of work, a Web Service Security performance will be modeled and the performance evaluation of WSS will be carried out using DES technology to check the cost effectiveness of security on web service taking WSS overhead into account.

This work is specifically;

(a) proposed a DES-based Web Service Security architecture;

(b) modeled a DES based Web Services Security Performance; and
(c) evaluated the performance of Web Services Security (WSS) using some standard metrics.

3. SYSTEM ANALYSIS AND DESIGN

Performance Evaluation of Web Service Security
WSS improves the security of web services. However, WSS incurs additional performance overheads to standard web services owing to additional CPU processing and larger messages transferred.
This WSS overhead was evaluated in [C+07]. The authors designed a web service that was benchmarked with and without WSS Security policies and their results were compared. A simple request/reply style messaging was used as benchmark. This consists of a client and a web service. The client sends the web service a request for a list of customer records. For flexible testing configuration, the request contains information required for a specific test such as message sizes and security setting was designed.
The web service receives the request and replies by sending back a specific amount of customer records with/without applying WSS policies according to the information in the request.
The prototyping was done in C# and deployed on Microsoft NET Framework 2.0 with WSE 2.0. In addition, XML security technology was used to carry out the performance evaluation on web service security [C+07].

3.1. The Proposed Model

In order to reach the goal of implementing this work, the work was carried out using a symmetric key cryptographic scheme (DES). This scheme provides for confidentiality and Integrity of messages, by encrypting such messages. DES also makes use of a secret key to preventing unauthorized access to data/information that has been encrypted. DES scheme is important for the protection of government, commercial, school and private information whether in store or on transit.

3.2 Data Encryption Standard (DES)

DES is a Symmetric cryptographic algorithm that enables the transformation of information from a clear plaintext form to an unintelligible coded form called cipher-text ([G+13]).
DES is a block cipher, so it operates on a single chunk of data at a time encrypting 64 bits (8bytes) of plain text which can produce 64bits of cipher-texts. The key length is 56 bits which is equivalent to 8 characters with extra bit used as parity check.

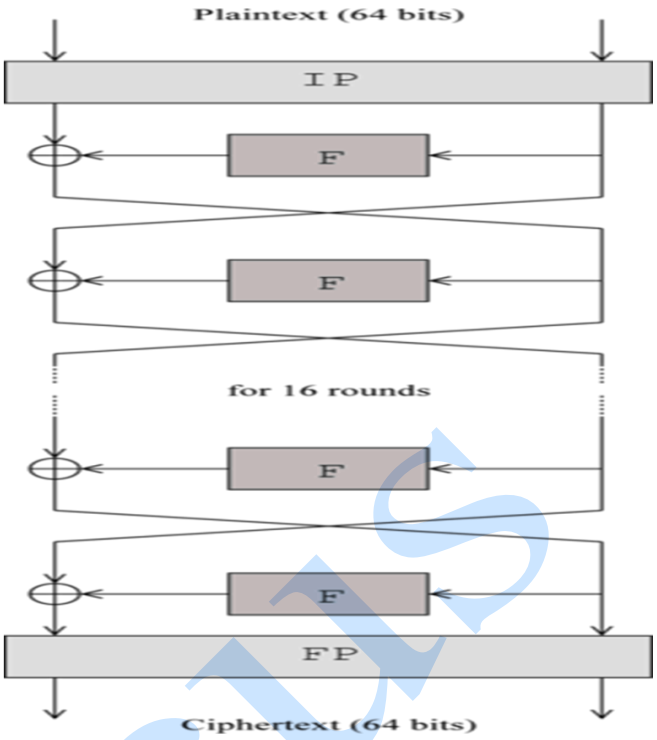

Figure 2: The overall Feistel structure of DES

The algorithm has 19 stages. The first stage record 64 bit input block by applying a fixed permutation while the last stage is the exact inverse of the permutation. During each iteration, the algorithm takes in two 32-bit inputs and produces two 32-bit outputs. The left output is simply a copy of the right input. The right output is the exclusive OR (XOR) of the left input [DMH01].

3.2.1 Mathematical Theory of Data Encryption Standard (DES)

DES exhibits the complementation property as follow:

$$E_K(P) = C \leftrightarrow E_{K'}(P') = C' \quad (1)$$

where;
E_K denotes encryption with key K
P & C denotes plaintext blocks.

3.2.2 DES-WS Architecture Overview

The DES-WS architecture in Figure 3 is made up of four (4) components; the UDDI, Web Service Requester (WSR), Web Service Provider with DES (WSP-DES), and a Database Management System of student record to be encrypted (DBMS).
DES allows DBMS information from the WSP to WSR to be secured.

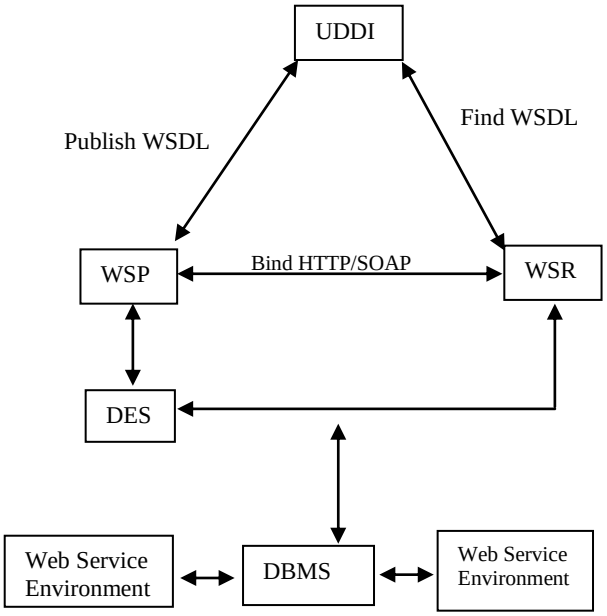

Figure 3: The Proposed DES-WSS Architecture Overview

3.3 Performance of Web Services Security

In determining the performance of web services security, the additional time cost on message transmission as well as the additional time cost on processing the security content of the message is considered.
The performance of web services security can be captured as follows;

$$P_{wss} = P_{ws} + T_{smt} + T_{scp} \quad (2)$$

where:
P_{wss} = Performance of web services security
P_{ws} = Performance of web services without security
T_{smt} = Additional time cost on SOAP message Transmission
T_{scp} = Additional time cost on processing the secured content of messages

3.4 Web Service Request and Response Interface

This is the Interface which shows the result of the requested information and the respond to the information with or without encryption. It also shows the extra CPU time cost for processing the encryption. The following data are shown on the Interface:

- REQINFO: Number of requested information
- LRCWOE: Length of requested content without encryption
- TRRWOE: Request/Respond time without encryption
- SCTPWOE: Server CPU time to process request without encryption (milsec)

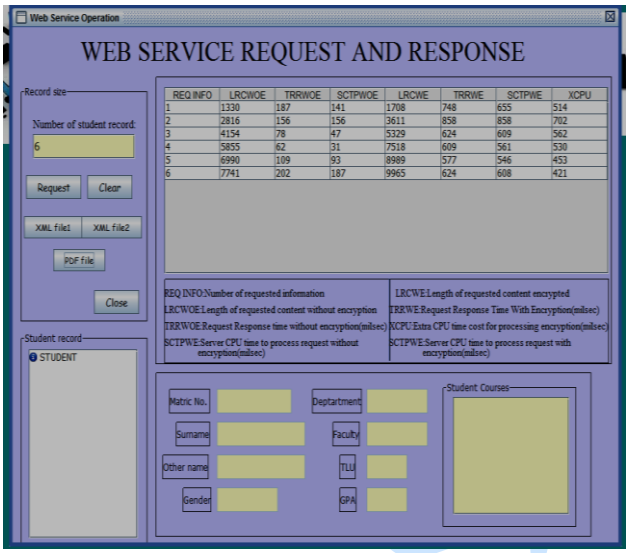

Figure 4: Web Service Request and response Interface

- XCPU: the extra CPU time cost for processing the encryption=SCTPWE-SCTPWOE
- LRCWE: Length of requested content with encryption
- TRRWE: Request/Respond time with encryption
- SCTPWE: Server CPU time to process request with encryption (milsec).

3.5 Performance Evaluation

Table 4.1 shows the performance evaluation of the Web Service Request and Respond Output of seven (7) records of students obtained with or without encryption.
Table 4.1 Web Service Request and Response Output table. We further evaluated with the aid of graph. The graphs showing the performance evaluation are printed out from the PDF file which is on the Web Service Request/Response Interface**.**

3.5.1 Graph of Extra Server CPU Time Cost for Processing Encryption (XCPU)

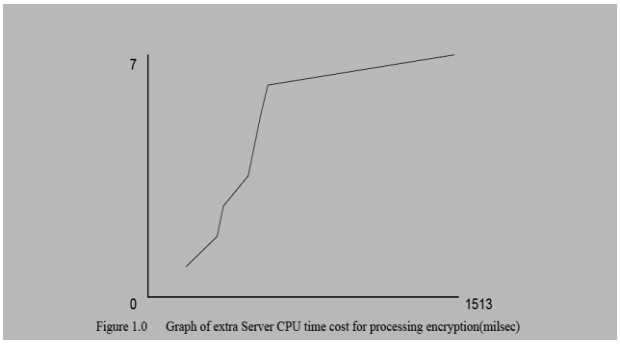

Figure 5. Graph of Extra Server CPU time cost for processing encryption (milsec)

This graph shows the XCPU for the seven (7) records that are requested for.

3.5.2 Graph of SCTPWE and TRRWE

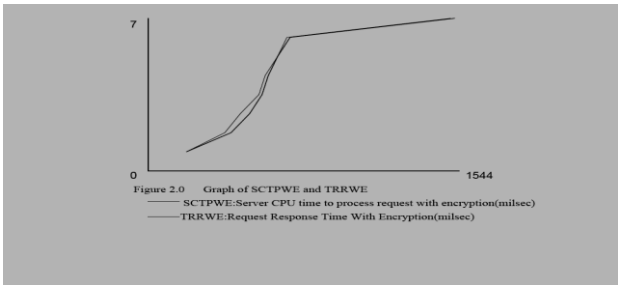

Figure 6: Graph of SCTPWE and TRRWE

This graph shows the Server CPU Time to Process Request with Encryption (SCTPWE) in conjunction with the Request/Respond Time with Encryption (TRRWE). The dotted line represents SCTPWE while the thick line represents TRRWE. While SCTPWE is increasing, the TRRWE is also increasing.

3.5.3 Graph of SCTPWOE and TRRWOE

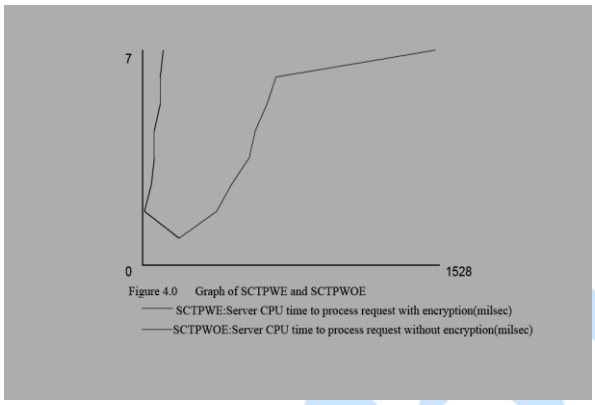

Figure 7: Graph of SCTPWE and SCTPWOE

This shows the graph of Server CPU Time to Process Request without Encryption (SCTPWOE) and Request/Respond Time without Encryption (TRRWOE). The time it took the Server CPU to process the request without encryption is smaller than the Request/Respond time.

3.5.4 Graph of SCTPWE and SCTPWOE

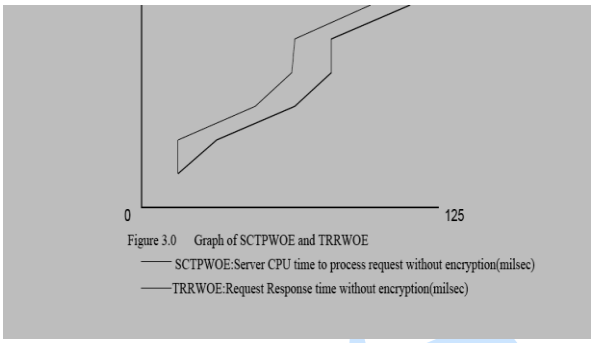

Figure 8: Graph of SCTPWOE and TRRWOE

This graph compares the Server CPU Time to Process Request with Encryption (SCTPWE) and Server CPU Time to Process Request without Encryption (SCTPWOE). From the graph, it is clear that the SCTPWE execution time is greater than the SCTPWOE execution time and this difference gives what is known as the XCPU time which is major challenge in the Web Service Security (WSS). Though the advantage of WSS is greater when compare to its disadvantage.

3.5.5 Graph of TRRWE and TRRWOE

This graph shows the time of request/respond for information with encryption and the information without encryption. The execution time of the request/respond of information with encryption is greater than the execution time of the request/respond of information without encryption.

Table 1. Performance evaluation of Web Services on the R

REQ INFO	LRCWOE	TRRWOE	SCTPWOE	LRCWE	TRRWE	SCTPWE	XCPU
1	1332	31	15	1708	1544	1528	1513
2	3783	125	94	4853	515	468	374
3	5121	63	47	6571	422	390	343
4	6822	78	63	8760	577	562	499
5	7957	78	62	10231	609	593	531
6	8708	110	94	11207	655	655	561
7	9090	125	109	11698	718	702	593

4. CONCLUSIONS

Lack of security can bring about threats to the message/information requested for, and the entire network infrastructure. Keeping messages secure by encryption will to a great extent, ensure their Integrity, Privacy and Confidentiality is not compromised.

In this work, a DES based WSS architecture was proposed. Its performance was then modeled and evaluated. Although deploying security with web services will incur some additional overhead, it will guarantee end-to-end security of information especially when in transit across enterprise networks.

REFERENCES

[AAH06] **C. P. Adelina, K. Ali, Hishamnddin** – *E-Commerce: A study on online shopping in Malaysia*. Journal of Social Science. 3(3), 231-242.

[BHM04] **D. Booth, H. Haas, F. McCabe** – *Web Services Architecture, W3C Working Group Note*.http://www.w3c.org/TR/ws-arch/ 2004.

[C+07**]** **S. Chen, B. Yan, J. Zic, R. Liu, A. Ng**, *Evaluation and Modelling of Web Services Performance*, In Proceedings in the IEEE International Conference on Web Services (ICWS'07). 2007.

[Dou04] **E. Douglas -** Computer Networks and Internets with Internet Application P. p542-596, 2004.

[DMH01] **O. Donal, P. Michael, T. Hitesh -** *Electronic Payment Systems for E-commerce*. Second Edition. Pp22. 2001.

[Enc12] **C. Encyclopedia** - *The Colombia Electronic Encyclopedia*, sixth edition. Colombia University Press. 2012.

[Eri00] **N. Eric** - *Understanding Web Services.* Addison Wesley Professional. p.7, 2000.

[Gar11] **A. Gartner** - *E-health for a Healthier Europe*. Retrieved online at www.calliope-etwork.eu/Linkclick.aspx, 2011

[G+13] **J. A. Gabriel., B. K. Alese, A. O. Adetunmbi, O. S. Adewale** – *Post-Quantum Crystography: A Combination of Post-Quantum Cryptography and Steganograph*. International Conference for Internet Technology and Secured Transactions, United Kingdom. December 2013.

[Hon10] **K. Hongzhao** - *A Study on the Security Mechanism for Web Services*. In Proceedings of the World Congress on Engineering and Computer Science 2010 Vol. I WCECS 2010, October 20-22, 2010, San Francisco, USA

[OASIS04] **OASIS Web Services Security** - *Web Services Security: SOAP Message Security 1.0* (WSSecurity 2004), 01 March 2004. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsssoap-message-security-1.0.pdf.

[OMA11] **O. M. Olaniyan, T. Mapayi, S. A. Adejumo** – *A Proposed Multiple Scan Biometric-Based System for Electronic Voting*, African Journal of Computing and ICT (Journal of IEEE Nigeria Computer Section), 4(2), 9 – 16. 2011.

[O+11] **O. M. Olaniyi, O. T. Arulogun, E. O. Omidiora, O. Adeoye** - *Design of Secure Electronic Voting System Using Multifactor Authentication and Cryptographic Hash Functions*, International Journal of Computer and Information Technology (IJCIT),2 (6),pp 1122-1130. 2013.

[RK11] **A. Roy, S. Karforma** - *Risk and Remedies of E-governance Systems*, Oriental Journal of Computer Systems and Technology, 4(2), 329-339. 2011.