International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 28 • 2017

Extended XML Security using Enhanced Digital Signature and Encryption

A. Murugan[a], P. Sigamaninathan[b] and K. Vivekanandan[c]

[a]*Asst. Prof. Department of Computer Science & Engineering, SRM University. Email: murugan.abap@gmail.com*
[b]*PG Student, Department of Computer Science & Engineering, SRM University. Email: siga@sigamani.in*
[c]*Professor, Department of Computer Science &Engg, Pondicherry Engineering, College. Email: k.vivekanandan@pec.edu*

Abstract: XML is the widely-used format for information exchange in the connected world and it is most extensively used in web services. Presently various mechanisms like WS-Security provide adequate security mechanism but implementation issues etc., are undermining the security of such systems. In this paper, the proposed security model takes into consideration most of the pitfalls like XML rewrite and XML encryption, non-repudiation etc., and also provides a higher degree or protection for XML based information exchange even in a less secured unencrypted network connections.

Keywords: {xml security, xml rewrite attacks, xml injection, web services security, xml digital signature}.

1. INTRODUCTION

Web Services have become an integral part of internet and they simplify the effort of information exchange. XML and SOAP has become the de-facto standard for information exchanges through web services and hence a highly attractive entity for malicious elements and hackers. XML files come under various type of attacks like XML rewriting/XML Signature wrapping, XML Injection, XML Signature – Key retrieval XSA, XPATH Injection, XML Flooding etc., to name a few common types. Various standards and protection mechanisms like TLS/SSL and even WS–Security **[1]** SOAP header with provision for confidentiality and integrity. But all this measure does have their own short falls like security loopholes, weak encryption, configuration issues, performance etc., Most of the available mechanisms offer foolproof security against XML signature wrapping or XML rewrite type attacks which are very difficult to detect with available tools. One mechanism is to use s complex deep learning for the detection but it is not an easy solution and involves great time and cost for implementation. An off the shelf and easy to implement solution for the XML rewrite attack and security must emanate from already available bullet proof encryption algorithms and available standards so that the security arrangement can be easily implemented with full confidence. In general security in connected world is achieved by the means of Transport Layer Security (TLS) and Secured Sockets Layer (SSL) combination. This mechanism of security creates an end-to-end security arrangement between the two points but the SOAP messages cannot be opened and routed or filtered by intermediaries and not suitable for the message based architecture of SOAP.

WS security Standards **[1]** can be implemented for overcoming the above problems but even though it can detect the attacks they are prone for lots of configuration problems/mistakes and may introduce vulnerabilities **[2].**

XML digital signatures can be done on selective elements of any XML file because at most of the times various parts of the XML file/message are populated/modified by various intermediaries inside the whole macro level system.

2. RELATED WORK

The goals of the XML signature are to protect the contents of the XML files from unauthorized change and malicious alteration. Even though the XML signature protocol fully achieves this goal, there are inherent weaknesses in the said protocol **[3]**.XML Elements along with their signatures can be copied and used in other XML files or locations. Web Service Security usage of XML DSS protects the node/elements name, attributes, and values etc., but not the position of the node within a typical XML document. This feature is designed for flexibility originally. But this feature leads to an important flaw that the signed XML elements can be moved from one part of the XML file to other and their location/ancestry can be easily changed. This change is perfectly acceptable to the XML DSS Signature standard **[3].** This change which cannot be detected by normal XML DSS validating mechanisms can create serious security related issues as mentioned in **[4].**

Currently XML rewrite attacks can be detected**[13][14]** and prevented by several ways. The easiest solution is to completely encrypt the XML message**[5],** to derive the location from simple ancestry, and as detailed in various examples detained in **[6].**Various techniques like Policy based approach for detection of XML signature wrapping attacks as detailed in **[6],** simulation based approach**[15],** and even an attack tolerant model is proposed in**[17]** but none is completely fail safe from sophisticated signature wrapping attacks. Another approach is the inline approach as detailed in **[7].**

In addition to the above in case if the contents of the node are to be protected it may be required to encrypt the same with an encryption algorithm but the selection of the algorithm and level of encryption plays an important role as the higher the encryption, more the security but poorer will be the performance as referred **[8].**

3. ENHANCED XML SIGNATURE

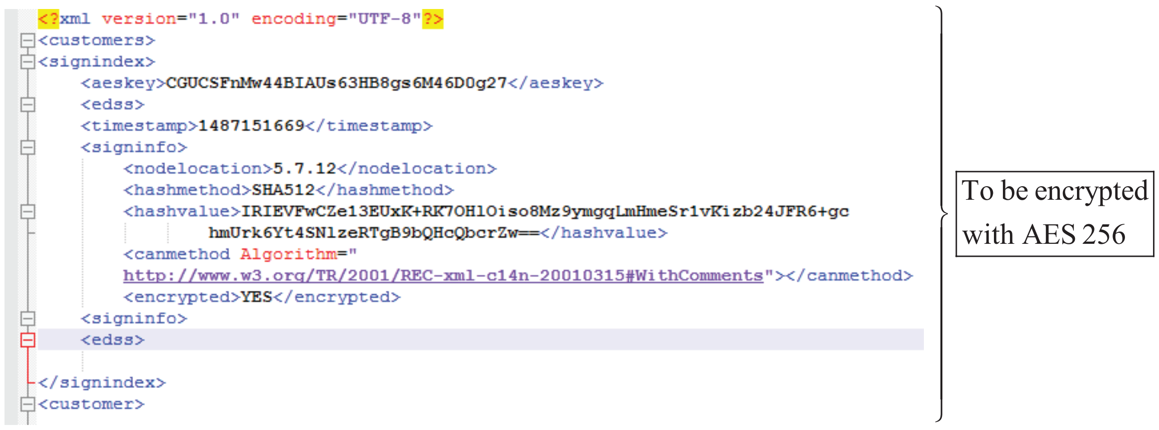
In this section, it is proposed to have an enhanced security algorithm and framework. The aim of this framework is to prevent most of the security loop holes narrated in web service security as shown in Figure 1. After all the individual XML nodes are signed by different entities in the system, before transmission of the request to the web service provider or the server the client system will have a 'finalize' procedures.

One XML node named <signindex></signindex> is created as an immediate child to the root node and it must appear as the first child to the root node. This may contain any number of <signInfo> </signInfo> nodes which encompasses. The location of the signed node relative to the root as walked from the root. An ordinary dashed approach like 3-2-14 is proposed to reduce the computational complexity involved (as in the case of XPath etc.,) in locating the node at the server side.The hash value of the signed content. The type of hashing used, the canonicalization method usedand the contents are fully encrypted. A gist of the unencrypted node as proposed is available in Figure 1.

After entering all the signed nodes inside the <signindex> element, the next step is to encrypt the entire <signindex> node with a symmetric key encryption algorithm. It is proposed to us a proven algorithm. So, this paper proposes to use AES-256 as the symmetric key encryption for encryption of the <signindex> element as well as the optional encryption required for encryption complete nodes. AES-256 is proposed because of its proven security records and any custom encryption algorithm may have inherent weakness by the way compromising the security of this entire algorithm. Other algorithms like 3DES, Blowfish may be used but they are supposed to be implemented completely in software coding whereas AES-256 is already implemented in hardware by the way of AES-NI instructions in almost all the current generation of microprocessors thereby boosting the performance of AES encryption/decryption**[9][10].**

A key with entropy of at least 2^{240} is proposed as the AES key and generated on the fly by the client side. As the key generation is part of the client side load, the server side number crunching is limited to just decryption and validation of the encrypted contents. Complete architecture of the proposed system with all key components and processes is shown in Figure 2.

Since symmetric key encryption is used, there must be a mechanism for secure key exchange but this presents another complex set of requirements. In order to avoid another key exchange/issue infrastructure this paper proposes to bundle the AES key in the same XML file by introducing a <keyvalue> element inside the <signindex>node.To make the AES key secret, it is proposed to utilize RSA public key of the Web Service Provider (Server) to encrypt the Symmetric key so that the Symmetric key can only be decrypted by the Web Service Provider. This virtually eliminates the possibility of key leakage/compromise during the key exchange mechanism and because of the strength of the RSA algorithm the entire XML signature process is secure from any type of XML Signature wrapping or XML Rewrite type attacks.If required any sensitive XML elements may be completely encrypted by using the same AES key so that the contents of the element is also protected, which is unbreakable using current level of computing power within reasonable time.



```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
<signindex>
    <aeskey>CGUCSFnMw44BIAUs63HB8gs6M46D0g27</aeskey>
    <edss>
    <timestamp>1487151669</timestamp>
    <signinfo>
        <nodelocation>5.7.12</nodelocation>
        <hashmethod>SHA512</hashmethod>
        <hashvalue>IRIEVFwCZe13EUxK+RK7OH1Oiso8Mz9ymgqLmHmeSr1vKizb24JFR6+gc
                hmUrk6Yt4SNlzeRTgB9bQHcQbcrZw==</hashvalue>
        <canmethod Algorithm="
        http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"></canmethod>
        <encrypted>YES</encrypted>
    <signinfo>
    <edss>

</signindex>
<customer>
```

Figure 1: Sample <signindex> node

The table below is used to represent the analysis of various parameters compared with existing system.

Table 1
Feature Comparison

Parameter	*Existing XML DSS*	*Proposed System*
Non-Repudiation	Not Secure	Complete
XML Rewrite Attacks	Possible	Protected, Secure
Encryption of Sensitive Info	Not Possible	Implemented
Key Management	Only Hashing	Highly Secure (RSA)

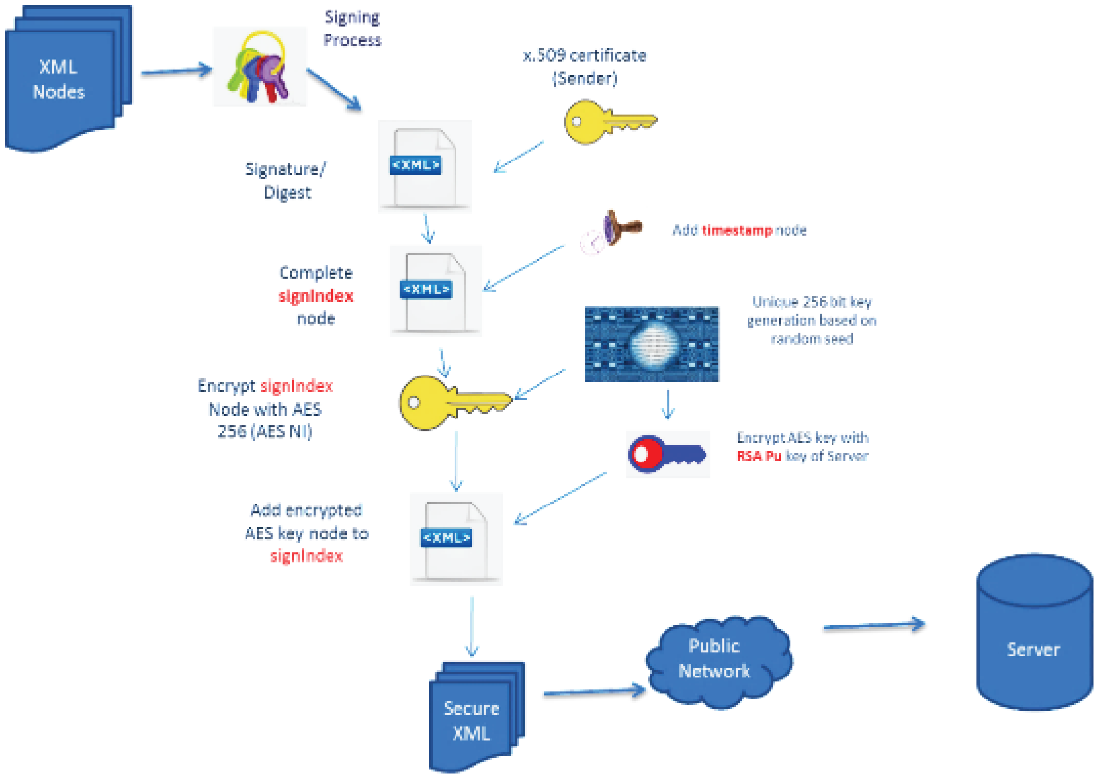

XML Nodes
Signing Process
x.509 certificate (Sender)
Signature/ Digest
Add timestamp node
Complete signIndex node
Unique 256 bit key generation based on random seed
Encrypt signIndex Node with AES 256 (AES NI)
Encrypt AES key with RSA Pu key of Server
Add encrypted AES key node to signIndex
Secure XML
Public Network
Server

Figure 2: Proposed System Architecture

4. CONCLUSION AND FUTURE WORK

Even though the proposed algorithm is clearly secure in nature from all types of attacks that can be envisaged on an XML webservice request it may slightly a performance bottleneck not because of the AES encryption (as AES NI**[10]**hardware implementation is used) but because of the RSA encryption and decryption required at the client and server side. It can be implemented and tested using various custom algorithms like proposed in **[8]**, RSA small e **[11],** Blowfish**[12]**, etc. it may also be explored that an encryption algorithm with small key size say 56/128 bits can be used and the timestamps can be compared and the xml may be discarded with a timeout value just a bit lesser than the time required to break the encryption with all known type of attacks on that encryption algorithm. Suitable mechanisms can also be devised to mitigate and restore the XML signed nodes to original locations from the values in the <signindex> node, with minimum computational efforts.

The main feature of this algorithm is the resultant XML file cannot be tampered by any means (i.e) neither the contents of the file can be modified by even a bit not the nodes can be juggled for creating any type of rewrite/ node manipulation attacks. Another important feature is as the hash value of the AES encrypted <signIndex> node is signed with the sender's RSA private key, the server of receiver while receiving the XML file will select the corresponding sender's public key based on the sender id in the XML file. If any man-in-the-middle style attack tries to replace even the entire file with a look alike legitimate one, the private/public key of the sender will not match hence the file will be marked as not legitimate and further investigations on the sender can be initiated after collecting required forensics.

The salient feature is, this high level of security is obtained with not encrypting the contents of the XML file completely but only utilizing a combination of hashing, AES and RSA algorithms to selected parts suitably designed to mitigate any type of attacks on any given XML document.

Another feature is the most of the computation intensive process is distributed, (i.e) done at the client side and the server infrastructure is spared from at least half of the computational complexity involved as the server side is not required to handle all the security mechanisms single handedly.

REFERENCES

[1] Nadalin, Kaler, Hallam-Baker, Monzillo. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004) OASIS Standard 200401, March 2004.

[2] Rahaman, Schaad, SAP Research. SOAP-based Secure Conversation and Collaboration, IEEE International Conference on Web Services (ICWS 2007), 0-7695-2924-0/07

[3] XML Signature Syntax and Processing (Second Edition), W3C Recommendation 10 June 2008

[4] Juraj Somorovsky, Mario Heiderich, Meiko Jensen, Jörg Schwenk, Nils Gruschka, Luigi Lo Iacono. All Your Clouds are Belong to us – Security Analysis of Cloud Management Interfaces, CCSW'11, October 21, 2011, Chicago, Illinois, USA. ACM 978-1-4503-1004-8/11/10

[5] Anupkumar M Bongale, Nithin N, Nirmala C R. LRXE: Lite-RSA for XML Encryption Suitable for Computational Constraint Devices.2014 IEEE Global Conference on Wireless Computing and Networking (GCWCN) 978-1-4799-6298-3/14/

[6] Te-Shun Chou. SECURITY THREATS ON CLOUD COMPUTING VULNERABILITIES. International Journal of Computer Science & Information Technology (IJCSIT) Vol. 5, No 3, June 2013. pp. 79-88

[7] Mohammad AshiqurRahaman, Maarten Rits, Andreas Schaad SAP Research. An Inline Approach for Secure SOAP Requests and EarlyValidation. OWASP Europe Conference, Leuven, Belgium, May 30-21, 2006.

[8] V. Sankar, Dr. G. Zayaraz. SECURING CONFIDENTIAL DATA IN XML USING CUSTOM LEVEL ENCRYPTION. 2016 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC). 978-1-5090-0901-5/16/

[9] Abhi Basu, Abhilasha Bhargav -Spantzel. Intel® AES - NI Performance Testing on Linux/Java Stack. Intel Corporation.

[10] Vilas V Deotare, Dinesh V Padole Ashok S. Wakode. Performance Evaluation of AES using Hardware and Software Codesign. International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 2 Issue: 6. ISSN: 2321-8169, 1638 – 1643

[11] Faraz Fatemi Moghaddam, Maen T. Alrashdan, and Omidreza Karimi. A Hybrid Encryption Algorithm Based on RSA Small-e and Efficient-RSA for Cloud Computing Environments. Journal of Advances in Computer Network, Vol. 1, No. 3, September 2013. DOI: 10.7763/JACN.2013.V1.47. 238-241

[12] Bruce Sctmeier. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994, pp. 191-204.

[13] Aziz Nasridinov, Jeong-Yong Byun and Young-Ho Park. A Study on Detection Techniques of XML Rewriting Attacks in Web Services. International Journal of Control and Automation Vol. 7, No. 1 (2014), pp. 391-400.

[14] Sebastian Gajek, Meiko Jensen, Lijun Liao, JorgSchwenk. Analysis of Signature Wrapping Attacks and Countermeasures. IEEE International Conference on Web Services. 978-0-7695-3709-2/09 pp. 575-582

[15] Pham, Phuoc Hung, Aziz, Nasridinov, Lin, Qing, Byun, Jeong-Yong. A Solution for Injection and Rewriting Attacks on SOAP Messages in Web Services Security. Journal of KIISE: Computing Practices and Letters, Volume 18, Issue 3, 2012, pp. 244-248, Korean Institute of Information Scientists and Engineers

[16] Azzedine Benameur, Faisal Abdul Kadir, Serge Fenet. XML rewriting attacks: existing solutions and their limitations. IADIS Applied Computing 2008. [arXiv:0812.4181. Cornell University Library]

[17] Nasridinov, Aziz; Hung, Pham Phuoc; Qing, Lin; Byun, Jeong-Yong;.XaT-SOAP: XML-based Attack-Tolerant SOAP Messages. Journal of KIISE:Computing Practices and Letters. Volume 18, Issue 6, 2012, pp. 489-493. Korean Institute of Information Scientists and Engineers.