XML Dynamic Key Encryption Approach

Shene Jalil Jamal [#1]

#1 Faculty of Science and Science Educations, Computer Department, University of Sulaimani, Kurdistan Region, Iraq.,
Phone no. 00964(7707723409)

ABSTRACT

Nowadays in the era of technology, security of data is being highly important to protect the data against misuse or data lost. Currently, XML formatted data is used in various applications and systems, such as e-commerce and business transactions, customer records, producing business management reports, e-government services etc. Therefore, the security of XML schemas is vitally important to ensure the protection of the data, data confidentiality, integrity, and access control. W3C and other standard organizations have tried to establish standard features that can solve security issues of the XML data that is transmitted on the internet. The XML Signature and XML Encryption are specifications recommended by W3C as the basis for all the XML security standards [1]. This paper discusses XML Encryption and XML Signature specifications that are recommended by W3C. Furthermore, this study examines a new encryption approach that uses dynamic key to encrypt XML documents. The new approach is called **D**ynamic **K**ey **E**ncryption **M**ethod, it has been implemented on an e-commerce website. **DKEM** uses a mathematical function to generate a dynamic key (depending on the length of the XML document to be encrypted) to perform the encryption process.

Key words: Encryption Algorithm, XML Encryption, XML Security, Dynamic Key and Dynamic Key Generator

Corresponding Author: Shene Jalil Jamal

INTRODUCTION

Data encryption is a mathematical method to convert digital data to a form which cannot be understood by unauthorized parties. A specific algorithm is used in this process to generate an encryption key. The encrypted form is called cipher-text which can only be read by the intended recipients if decrypted using the key. The purpose of this process is to protect the confidentiality of the digital data that is transmitted over the internet or stored in a server or in any other storage media [2].

With increasing the amount of data expressed in XML format, and increasing the popularity of data exchange in XML in the web service applications, strong data protection standards are needed to protect the data from unauthorized access. The security aspects are necessary to ensure that the XML data used on the web service applications is kept and only accessed by authorized recipients. The XML Encryption and XML Signature are the publications announced by W3C for securing XML data. In the Encryption process the data (which could be XML element or the content of the element) is encrypted, and the encrypted element contains the cipher data. The result of encrypting the data also expressed in XML format [3]. In the decryption process the XML Signature is enabled to separate the XML encrypted structures which must be decrypted from those that must not be decrypted [3,4]. In this study despite of discussing the existing XML Encryption approaches, a new encryption method to encrypt XML document will be discussed. This method is called **DKEM,** which depends of the size of the XML document to generate an encryption key. This key is used to encrypt the whole xml document. This approach is examined on an ecommerce web application to encrypt customer's profile data (which is stored in XML format). In this method if any change happens to the file (which contains the customer records and profile information) then the encryption key will be changed, as a consequence, the customer profile data will be encrypted again using the new key. This way each customer profile is encrypted using different encryption keys, and the encryption key for the same customer profile is dynamically changed (when the file size is changed). As the result, it makes it hard or even impossible for the unauthorized parties to discover the encryption keys.

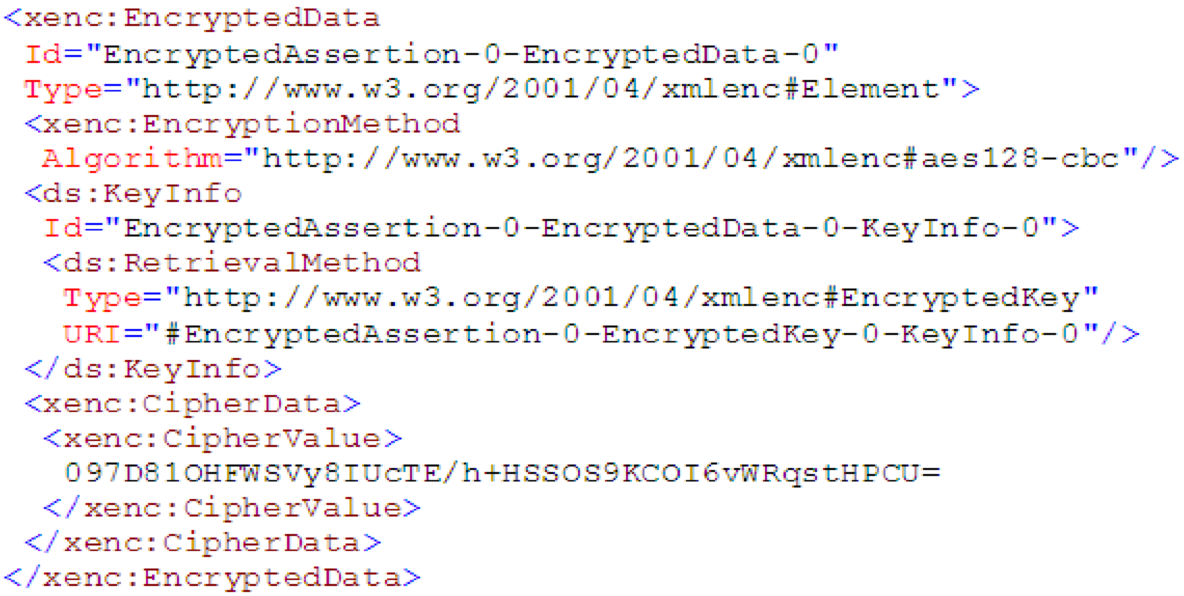
MATERIALS AND METHODS

This DKEM is an approach invented in this study to generate dynamic key for XML and text file encryption. A mathematical method used to make this approach that mainly depends on the size of the file which will be encrypted. An ecommerce website was developed only to examine this DKEM. This website was created using technologies such as Apache Tomcat, Servlet and JSP. On the website the customer profiles are expressed in XML format. Directly after registering a customer on the website the DKEM encrypts the customer profile. To execute queries on the customer profiles (such as retrieving information, adding and editing information) XQuery is used. Saxon-HE is used as XQuery processor. Any change made in a profile causes the length of the file to be changed, this will change the encryption key and a new key will be generated to encrypt the file.

XML Encryption and XML Signature

With increasing the popularity of XML formatted data, demands for securing the data also increased. In this section some XML encryption standards are explained.

XML Encryption was proposed by W3C and IETF in 2002 as standard for encrypting XML data in a document. This is widely used in server to server communications, with the idea of encrypting only the sensitive part of a document that is mixed with non-sensitive data. Researches have been done on using this method for encrypting parts of a document with different keys, and permitting recipients to only decrepit the part that is applicable to them. For example using this method to encrypt the credit card information, it would allow accessing the general information, but the financial information (which is the sensitive portion) can only be accessed by the authorized users [4, 5]. When a portion of document is encrypted using this method, the encrypted information enclosed by a tag in the beginning and the end appeared in the document [3, 6]. Bellow figure -1-shows the structure of an encrypted XML document.

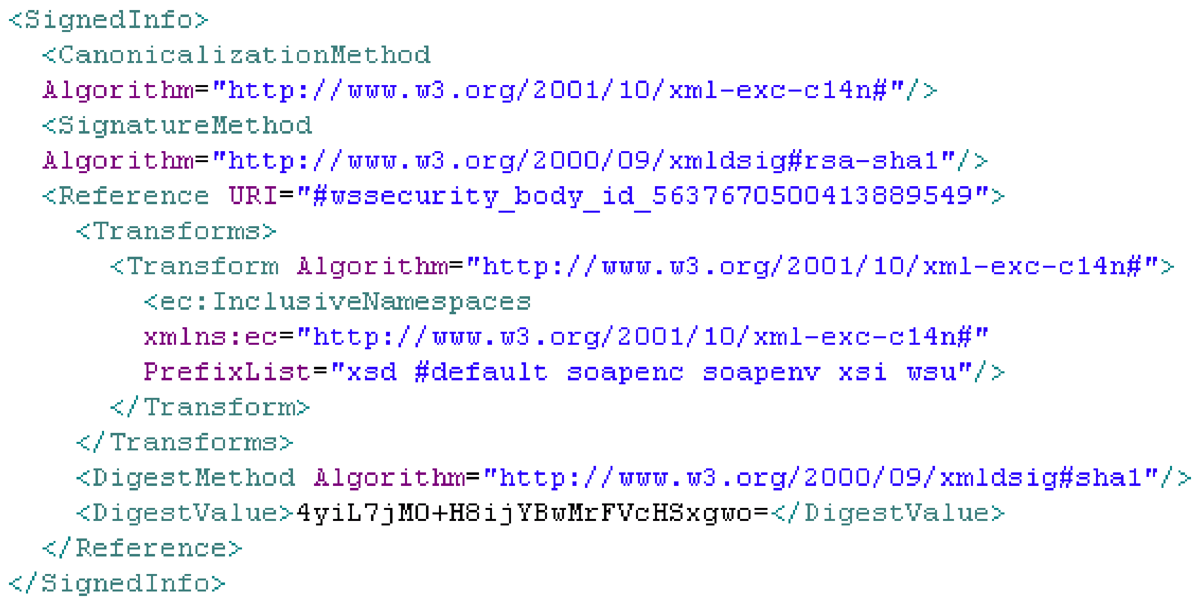


```
<xenc:EncryptedData
 Id="EncryptedAssertion-0-EncryptedData-0"
 Type="http://www.w3.org/2001/04/xmlenc#Element">
 <xenc:EncryptionMethod
  Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
 <ds:KeyInfo
  Id="EncryptedAssertion-0-EncryptedData-0-KeyInfo-0">
  <ds:RetrievalMethod
   Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"
   URI="#EncryptedAssertion-0-EncryptedKey-0-KeyInfo-0"/>
 </ds:KeyInfo>
 <xenc:CipherData>
  <xenc:CipherValue>
   097D81OHFWSVy8IUcTE/h+HSSOS9KCOI6vWRqstHPCU=
  </xenc:CipherValue>
 </xenc:CipherData>
</xenc:EncryptedData>
```

Fig 1: Structure of an encrypted XML document

Furthermore, Implementing XML Signature has become the main aim for a number of researches, with the purpose of a secure XML data transmission. This technique is used to verify that the data expressed in XML has not been modified after it is received by the intended recipients [7].

XML Signature and XML Encryption are very similar in solving security problems, as XML Signature also supports partial signature; hence, the Signature can be applied to specific tags in the XML document [3, 4]. This security technique uses the concept of canonicalization, when the content of an XML document is signed, a unique signature is made by canonicalization from the tags and the contents of the document to sign the document before it is sent. When it is received, XML Signature decryption is executed in the recipient system to separate the content encrypted after signing from those encrypted before signing, contents encrypted after signing is decrypted and the same canonicalization method is applied to the decrypted content. This way data integrity is achieved when the result shows that the data sent is the data received [8]. Bellow figure -2- shows the structure of a signed XML document.



```
<SignedInfo>
  <CanonicalizationMethod
  Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
  <SignatureMethod
  Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="#wssecurity_body_id_5637670500413889549">
    <Transforms>
      <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <ec:InclusiveNamespaces
        xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
        PrefixList="xsd #default soapenc soapenv xsi wsu"/>
      </Transform>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>4yiL7jMO+H8ijYBwMrFVcHSxgwo=</DigestValue>
  </Reference>
</SignedInfo>
```

Fig 2: Structure of a signed XML document.

Dynamic Key Encryption Method

The algorithms that have been invented for encryption digital data comes in two main types: Symmetric and Asymmetric. With Symmetric algorithms, a single key is used to convert the plain text data to cipher text. **Asymmetric** algorithms use two keys, public and private keys to perform encryption and decryption respectively [9]. The type of the encryption method performed in this study is Symmetric. This Encryption method is proposed for protecting data stored on a server, and for a secure data transmission especially in the business-to-customer e-commerce web services. To examine the DKEM encryption method a website which provides business-to customer sales services is developed. In this website all the data is expressed in XML format. The online service creates a profile document in XML for each registered customer. This document contains the information about the customer including their financial information. The DKEM encryption method only encrypts the customers profile data on the server side. The encryption key is generated based on the length of the XML file. Hence, each customer profile xml document is encrypted using different encryption key. In addition, if the customer makes any update to his/her profile document. And this update changes the length of the document, then a new key will automatically be generated and the document will be encrypted again using the new key. The stored data (customer profile) on the server is encrypted by the key, and it only can be read by the authorized recipient (the customer). This algorithm can be implemented to encrypt any text file to cipher text. However, in this paper the algorithm is only executed to encrypt XML files.

In XML Encryption standard (recommended by W3C) the encrypted data is represented in XML structure. This standard encloses the encrypted data and all encryption specifications in the "EncryptedData" XML element (if only the data is encrypted) or in the "EncryptedData" XML element and "EncryptedKey" element (if the data and the encryption key are encrypted) [10]. In contrast, this DKEM algorithm encrypts the whole file including all the tags and elements, and the encrypted data will not be expressed in XML format.

The generated encryption-key consists of eight characters. The following mathematical expression is used to generate the key.

KeyCharacter $= X - (|\frac{3*X}{Y}|)$

Where:

X= The length of the XML document that is calculated using java method **java.io.File.length()**

Y= A variable that takes its value in a for loop iterator. The loop starts from 4 to 12. In each step, Y takes a value increasingly from 4 to 8. Hence, 8 key characters will be produced which compose the 8-lengh character encryption key.

How it works

This algorithm is implemented in Java. The procedures of creating the key, encrypting and decrypting a file are explained in the following steps:

Step1: Constructing the Encryption Key:

To create the encryption key, a java method is used to get the length of the file. This value is used in an eight steps loop with a mathematical expression, the loop starts from four and increases to twelve to generate a key of eight character length. This key is stored in a storage collection (ArrayList is used in this practice).

```
for (int i= 4; i<12 ; i++) {
            KeyIndexes = XMLFile length() - (Math.abs(3*all.length()/i));
             IndexesStorage.add(KeyIndexes);
            keyCharecter = XMLFile.length() - (Math.abs(3* XMLFile. length()/i));
             keyCharStorage.add(keyCharecter);
        }
    KeysStorage.add(IndexesStorage);
    KeysStorage.add(keyCharecter);
```

Step2: Encrypting the data:

To encrypt the data, the XML file is read and put in a string object. Every character in the string is checked to determine if it’s a number or a letter (or special character). And the character is shifted using the key and some mathematical expressions.

```
String encryptedText="";
int  keyIndex=0;
for (int i=0; i < fileText.length(); i++) {
      int shift= KeysStorage.get(keyIndex);
      char fileChar = fileText.charAt(i);
      if ((fileChar >= 32 && fileChar <= 127) && ! IndexesStorage.contains(i)){
              int temp = fileChar - 32;
               temp = (temp + shift) % 96;
                if (temp < 0)  temp += 96;
                encryptedText += (char) (temp + 32);
                  keyIndex ++;
        } else{
              encryptedText += fileChar;
               }
            If (keyIndex == KeysStorage.size()-1)  keyIndex =0;
             }
return encrypted;
```

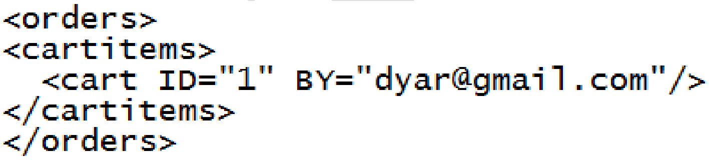
Step3: Decrypting the cipher text:

The same generated key is used to decrypt the cipher text. The code presented in ….. can be used to decrypt the file; with changing the eighth line to

temp = (temp + shift) % 96;

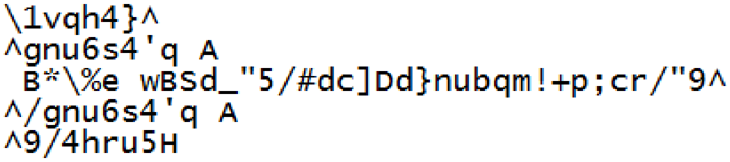
The following is a sample of XML document that is encrypted using DKEM method.

- Before Encryption



```
<orders>
<cartitems>
  <cart ID="1" BY="dyar@gmail.com"/>
</cartitems>
</orders>
```

- After encryption



```
\1vqh4}^
^gnu6s4'q A
 B*\%e wBSd_"5/#dc]Dd}nubqm!+p;cr/"9^
^/gnu6s4'q A
^9/4hru5H
```

CONCLUSION

Data protection has become vitally important in nowadays tech world. With increasing the amount of sensitive data expressed in XML, demands for protecting the XML data is increased. To satisfy this demand various security methods and techniques are developed. Encrypting XML data is one of those methods that provide a high level of security to the data expressed in XML. XML Encryption and XML signature are standards recommended by W3C for protecting XML data. XML Encryption aims protecting XML data in server to server communication, and it allows different parts of a document to be encrypted using different keys, and allows recipients to only decrypt the relevant part to them. Similarly, XML signature can be applied to specific parts of XML documents with the aim of proving that the content of the file is not changed after its received by intendant recipients [3, 4, 7].

The concept of using ‘Key(s)’ is used in all encryption techniques, and mathematical methods are used to generate the Key with the purpose of converting the XML data from a plan text to cipher-text. The encryption method discussed in this paper is a new approach of using mathematical methods to create a sophisticated Key to be used for XML data encryption. This method depends on the length of the XML file. It uses this length in a mathematical operation to generate eight-character length key for encrypting the file. The Symmetric encryption approach is used in this experiment; hence the same key is used in the recipient side to decrypt the data. This generated key could be used to encrypt any text file, however in this study it is only used to encrypt XML documents. This encryption method is examined on a e-commerce website to encrypt XML documents that contain profile information about customers. Because each customer document might have different length, the generated encryption key for each document is different. Furthermore, if customer makes any changes to his/her profile information that might cause the length of the profile document, then the key will automatically be changed and the document is encrypted again using the new key. This makes the encryption to be more secure than using the same static key to encrypt all the documents in a server or a system.

ACKNOLEDGMENT

I would like to show my gratitude to Dyar Ibrahim for providing his insight and expertise that greatly improved the research.

REFERENCES

[1] Saravanaguru, RA. K. et al, ‘Securing Web Services Using XML Signature and XML Encryption’ , School of Computer Science and Engineering, VIT University, Vellore, India, 2013, Available at: http://arxiv.org/ftp/arxiv/papers/1303/1303.0910.pdf .

[2] Lackey, E., Encryption and Decryption, 2012, Available at: https://developer.mozilla.org/en-US/docs/Archive/Security/Encryption_and_Decryption.

[3] MIYAUCHI, K., ‘XML Signature/Encryption the Basis of Web Services Security’, NEC Journal of Advanced Technology, 2005, 2 (1), pp. 35-39. Available at: http://www.nec.com/en/global/techrep/journal/g05/n01/pdf/a035.pdf.

[4] Yue-sheng, G., Meng-tao, Y. et al, ‘Web Services Security Based on XML Signature and XML Encryption’, JOURNAL OF NETWORKS, 2010, 5 (9), pp. 1092-1097. Available at: file:///C:/Users/Shene/Downloads/2595-8394-1-PB.pdf.

[5] Imamura, T., Dillawa, B. et al, XML Encryption Syntax and Processing Version 1.1., 2013, Available at : http://www.w3.org/TR/xmlenc-core1/.

[6] Imamura, T., Dillawa, B. et al, XML Encryption Syntax and Processing, 2002, Available at: http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview.html.

[7] Bartel, M., Boyer, J. et al, XML Signature Syntax and Processing (Second Edition), 2008, Available at: http://www.w3.org/TR/xmldsig-core/.

[8]Eastlake, D., Reagle, J. et al, XML Signature Syntax and Processing (Second Edition), 2008, Available at: http://www.w3.org/TR/xmldsig-core/Overview.html#sec-c14nAlg.

[9]Agrawal, M., Mishra, P., 'International Journal on Computer Science and Engineering', A Comparative Survey on Symmetric Key Encryption Techniques, 2012, 4 (5), pp 877-882. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.433.2037&rep=rep1&type=pdf.

[10]Yang, H., What is XML Encryption Syntax and Processing, 2016, Available at : http://www.herongyang.com/Web-Services/X509-Token-XML-Encryption-Syntax-and-Processing.html.

Figures References:

[1]Erdős, P., Annex C (informative): Generation of RSA modules, 2015, Available at: http://www.kormanyablak.org/it_security/2015-02-14.php.

[2]Liu, J., XML representation of a digest value, 2004, Available at : http://www.ibm.com/developerworks/websphere/library/techarticles/0403_liu/0403_liu.html.