ISSN No: 2456-2165

Secure Protocols for Data Exchange in XML-Based Applications

Alex Fabianne de Paulo[1], Ilmério Reis da Silva[2], João Nunes de Souza[2]
[1]School of Information and Communication, Federal University of Goiás, Brazil
[2]School of Computer Science, Federal University of Uberlândia, Brazil

Abstract:- Currently, a large amount of data is exchanged on Internet on a non-structuralized or half-structuralized way. The XML standard (eXtensible Markup Language) has been become the best solution to show and exchange data, making the structure and content of a document apart. However, some kinds of data are critical and need to be secure. This paper describes two new data exchange security protocols to XML-based applications in order to solve the problems of data heterogeneity and secure. Using the XML standard and concepts of cryptography, the protocols define rules that support different security levels according to the needs of each application. In addition, it presents a comparative evaluation between the two protocols for pointing the advantages and disadvantages of each one.

Keywords:- *Cryptography; Protocol; XML; Secure Data Exchange.*

I. INTRODUCTION

A large amount of data is available on Intranets and Internet on a non-structuralized or half-structuralized way. These data need to be accessible in an uniform and integrated way for both final users and software application layers. Because of HTML (HyperText Markup Language) limitations [6], these data are presented on an inadequate form, without clarity in separation between document structure and content. Consequently, the HTML markup is inadequate since the objective is to understand the data semantics.

In order to surpass the heterogeneity data problem, a great effort was made to provide a cautious markup technique that does not lose the HTML formatting and distribution potentialities. The main result of this effort for standardization is eXtensible Markup Language (XML), a solution for better representation and data exchange in the Internet specified by IETF/W3C XML Working Group [4]. The main characteristics of this language are simplicity, flexibility, legibility and interoperability.

In this context, the use of security mechanisms is essential under Web application, mainly regarding to confidentiality and authenticity. The data necessity to be authentic is evident under the focus of a commercial transaction. When the transaction involves different users, different parts of the system need different types of authentication [1].

Confidentiality is also important for many applications. Considering an application that makes critical information contained in a laboratorial exam result available. The portability and interoperability allow getting these data in different kinds of devices such as palmtops, mobile telephones or desktops, and manipulated them in different applications. The information secrecy is crucial to guarantee that the exam result is not going to be accessed or violated for someone else. The union of XML standard and some security mechanisms makes the data exchange through the Internet a more efficient and secure task.

This paper describes two new security protocols for XML-based applications in order to assure accessibility, interoperability and secrecy to data exchange on the Web. In addition, it presents a comparative analysis between these new protocols. In order to illustrate the protocols application, it is used a medical exam results exchange case study.

II. RELATED WORK

An authentication protocol such as Kerberos gets data security during its transmission but consider the main memories of user and server protected against intruders. Other secure protocols guarantee confidentiality, integrity and authenticity but providing this security on the transport layer, such as IPSec (Internet Protocol Security) or SSL (Secure Sockets Layer) [11]. In our protocols, the main memories of the machines are vulnerable to intruder's attacks. In order to avoid this, our protocol proposes a way to access and manipulate some encrypted data with no need to decrypt any data in the main memories of these hostile machines.

A multi-user protocol and data exchange protocol able to manipulate stored encrypted data with no need to decipher is proposed in [14]. This protocol is based on elliptic curves cryptography (ECC) [11, 15]. Our protocol is able to re-encrypt the data without decrypt it and expose the keys and the plain text. Instead of using ECC, a cryptographic protocol based on public key is used, more specifically, on the RSA scheme [11,16]. Although the ECC system has a better performance than RSA, we used the RSA scheme because it is based on exponential operations, which is essential to update the user keys without need to decrypt the data.

III. SECURITY REQUIREMENTS

Several Web applications need security, especially to support the following aspects:
- *confidentiality* to guarantee that the data contained in a document is not going to be accessed by non-authorized parts;
- *authenticity* to assure that the document proves a correctly identified origin, with the guarantee that the identity is not false;
- *integrity* to detect if some data contained in the document was modified;
- *non-repudiation* to guarantee that the sender does not deny the sending nor the receiver denies the act of receiving the document.

In order to support the security requirements as described in the previously, the following services are available:
- *IPSec (Internet Protocol Security) or SSL (Secure Sockets Layer):* these services guarantee confidentiality, integrity and authenticity, providing this security in the transport layer;
- *XML encryption/signature:* it guarantees the data confidentiality, authenticity and non-repudiation in the application layer.

The use of protocols as IPSec or SSL provides security in the communication through the Internet, but it is not applied to situations, which the data needs to be protected before and after been transmitted. Specifically, in case proposed on this paper, the use of cryptography assures the data security through the XML encryption [2] and digital signature [3]. The main motivation for the use of cryptography based on XML syntax instead of using a binary or text-based syntax is the necessity to have the encrypted or signed data as structures that can be created, manipulated and analyzed with XML tools [1].

IV. SECURE PROTOCOLS FOR DATA EXCHANGE IN XML-BASED APPLICATIONS

In this section, the new two security protocols to XML-based applications are presented. First, the case study of medical exam results exchange used to illustrate an application of the protocols is explained. Second, the notation used for protocols specifications is described. Following, the new two protocols to data exchange and the evaluation of them are presented.

A. Case study: secure medical data exchange

The continuous technological advances have provided a revolution in the medicine. The use of computer in the hospital evolved from a situation that the computer was used just for simple and isolated tasks, to the current global integration level, in which it wants to join the diverse points of generation and use of the information inside and outside of the institution [8].

It is increasing the number of medical institutions and professionals who try to offer a better attendance to the users, as much in time as much in cost. In this direction, the use of computer science resources is an essential stage. The worldwide trend points in direction to the digitalization of the clinical record of the patient through the electronic record of the patient. However, in lots of countries, the inexistence of an exclusive number that identifies to all the citizens since its birth, the lack of government support and the absence of a legislation that gives legal validity to the electronic record, can be some of the factors that make it difficult the evolution of the medical sector to the creation of the patient electronic record [9].

Searching continuous evolution, but still distant of the electronic record idea, there are several initiatives to make the laboratory exam results available on the Web. This process contributes mainly in accessibility and agility of exam result for the interested people either it, medical or patient.

```
<?xml version “1.0”?>
<ExamResult xmlns=‘http://www.hc.ufu.br/examresult’>
  <Identification>
      <ID>João da Silva</ID>
      <Cod>151233</Cod>
  </Identification>
  <Result> ... </Result>
</ExamResult>
```

Fig. 1: XML data structure of an exam result

However, some kinds of exams are critical and decisive results for the patient life. Ahead of this, this paper presents two new secure protocols that define rules to exchange exam results using XML standard. The figure 1 shows the data structure of an XML exam result.

B. Notes

In the next two sections, the new protocols applied in a case study of security medical exams results exchange are described. Some important notes are pointed on this section.

As first notation is the syntax used in the protocols. The logical propositional symbols $\vee$, $\wedge$ (or, and) are used in some flows of the protocols. The characters “C” and “D” express respectively encryption and decryption functions. The response value “ex” refers to the plain text of exam result and “exXML” indicates the plain text of exam result converted to XML standard. Other two kinds of response values are permitted: “N” represents a null value used to indicate that the data is invalid or not found, and “Cod” value that refers to exam identification code. The public and private keys are represented in KU and KR. These keys are different for user and laboratory following the “U” and “L” indexes.

Second important note, all operations over XML data, like canonicalization [7], encryption or digital signature following the syntax specified on IETF/W3C Working Group. Third, the way in which the data of the exam result will be encrypted is another point. In both protocols will be encrypted only the element <Result>, leaving the user identification and the exam identification code opened. Fourth, due to the portability and interoperability characteristics of XML language in both protocols, the user can access the laboratory servers through different devices like palmtops, mobile telephones or desktop computers.

Last, the use of public key certificates [8], will allow that the user and laboratory can exchange keys in trustworthy way without having to directly interact with a public key certificate authority. The certificates are previously gotten by each system entity (user and laboratory) together to a certificate authority.

The authority provides the certificate in the following form:

$C = E_{KRAUT}$ [T, ID, KU],

Which, C is the solicitant entity certificate, E_{KRAUT} is the certificate authority private key, T defines the validity of the certificate, ID is solicitant entity identification (name or code) and KU is the solicitant entity public key. C is obtained through the encryption of [T, ID, KU] with the private key KR_{AUT} of certificate authority.

Thus, C can be passed to any other entity, which could read and verify the certificate, decrypting C with the public key KUAUT of the certificate authority and getting [T, ID, KU]. This process follows this equation:

D_{KUAUT} [C] = D_{KUAUT} [E_{KRAUT} [T, ID, KU]] = [T, ID, KU]

Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority.

C. *On-demand protocol*

In the on-demand protocol, the process of data encryption occurs in real time and follows the demand of users as shown in the figure 2.

Suppose a user, either it patient, responsible doctor or another person assigned by the patient, send to laboratory a public key certificate (C_U) to get its respective exam result. In the laboratory front-end server, the user certificate (C_U) is decrypted with the authority public key (KU_{AUT}); getting the user identification (ID_U), its public key (KU_U) and the certificate validity (T_U). The front-end server returns to user the laboratory certificate (C_L) or a null value message (N), characterizing that the user certificate expired. It can be seen in figure 2 as indicated by [$C_L \vee$ N].

In case that user does not receive the null value message, it gets from C_L the laboratory public key (KU_L). At this moment, the user must send to laboratory the exam identification code (Cod). For this, the user signs Cod with its private key (KR_U), later encrypted it with the laboratory public key (KU_L) and sends the encrypted Cod to the laboratory front-end server, as indicated in the flow by $E_{KUL}[E_{KRU}$[Cod]].

To get the exam identification code (Cod), the front-end server forwards $E_{KUL}[E_{KRU}$[Cod]] and ID_U to back-end server that deciphers it using laboratory private key (KR_L) and verify The user signature using user public key (KU_U). Known Cod and ID_U, the back-end server queries to the relational database of exam results to obtain the exam result of corresponding Cod and ID_U. It is shown in the flow [Cod $\wedge$ ID_U]. If the exam identification code or the user identification is not found, the back-end forwards a null value message (N) to front-end server that forwards N to user.

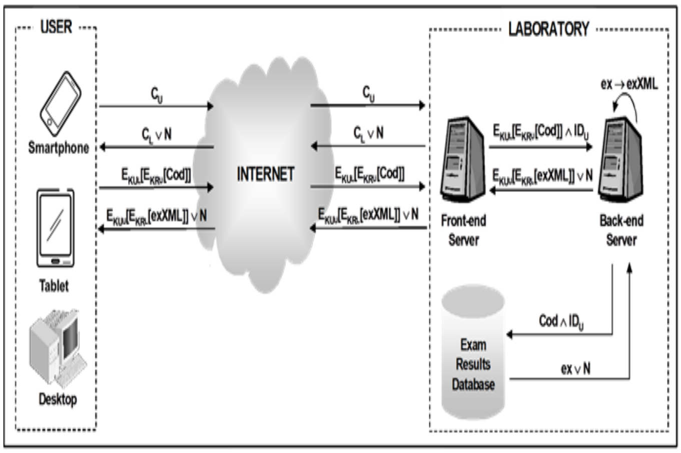

Fig. 2: On demand protocol scheme

But in case that a corresponding result to Cod and ID_U is found, the back-end server obtains "ex", the plain text of exam result searched. Then, the back-end server converts the plain text (ex) to a plain text XML version (exXML). Also, into back-end server, exXML is first signed using the laboratory private key (KR_L) and later it is encrypted using the user public key (KU_U), getting the encrypted exXML as it is shown in the flow by $E_{KRU}[E_{KUL}$[exXML]]. These two procedures assure confidentiality, authenticity and non-repudiation to the final result emitted by the laboratory. Figure 3 shows the encrypted exXML exam result.

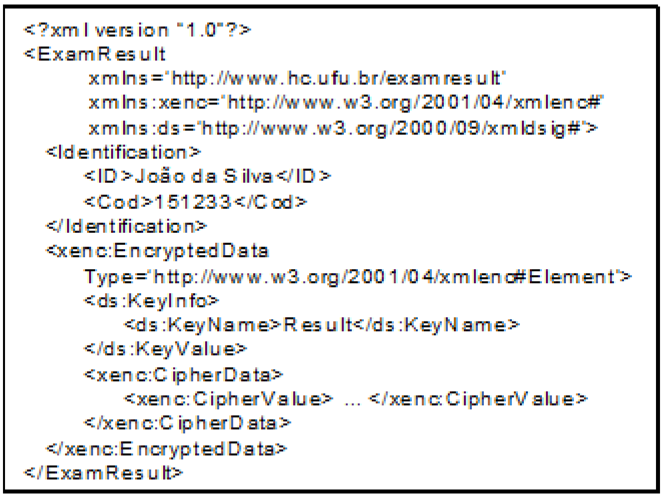
```
<?xml version "1.0"?>
<ExamResult
        xmlns="http://www.hc.ufu.br/examresult"
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
   <Identification>
        <ID>João da Silva</ID>
        <Cod>151233</Cod>
   </Identification>
   <xenc:EncryptedData
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        <ds:KeyInfo>
             <ds:KeyName>Result</ds:KeyName>
        </ds:KeyValue>
        <xenc:CipherData>
             <xenc:CipherValue> ... </xenc:CipherValue>
        </xenc:CipherData>
   </xenc:EncryptedData>
</ExamResult>
```

Fig. 3: XML encrypted data

Finally, the back-end server forwards to front-end server the encrypted exXML and then it is sent to the user. The user gets the exam result plain text after decrypting it using KR_U and verify the laboratory signature using KU_L.

D. Anticipatory protocol

The anticipatory protocol can be seen at figure 4. In this protocol, all procedures for conversion of the exam results for XML, as well as the encryption of these data occurs in an off-line way, before the user access request to the exam result. Moreover, the exam results will be always encrypted, not being available to the eventual spy's action. For such, the user must leave in the laboratory its public key certificate (C_U) when it will be doing the exams.

Analyzing the laboratory off-line procedures, the exam results data entry is made through application in the back-end server, as it can be seen in the flow "ex". Into the proper back-end server, the exams results plain texts are converted to XML standard and then encrypted. To execute the encryption operation, the back-end server signs the XML plain text (exXML) using the laboratory private key (KR_L). After it encrypts exXML using the user public key (KU_U) and stores it in a XML database exam results, as indicated for $E_{KUU}[E_{KRL}[exXML]]$, everything occurring off-line. In this way, the XML encryption data of exam results will be enclosed in the XML database daily, or as defined by the system's administrator. It's still the system's administrator responsibility defines how much time an exam result must remain available for access in the XML database.

Related to the user and the front-end server, the flows for exam results exchanging are the same ones presented in the on-demand protocol. A particularity of anticipatory protocol is accord to the back-end server that can operate in either off-line or on-line way. In off-line way, it supports all the data entry process, conversion for XML, encryption and storage of the exam results in the XML database. In on-line way, it assures the users request attendance came from front-end server.

E. Protocols evaluation

For both presented protocols, the use of two servers guarantees greater security and becoming the laboratory environment less vulnerable to attacks. This occurs because all the procedures involving the laboratory private key and the exam results plain texts manipulation are carried through in the back-end server. This server is not connected to the Internet and can be protected by firewall or other security devices.

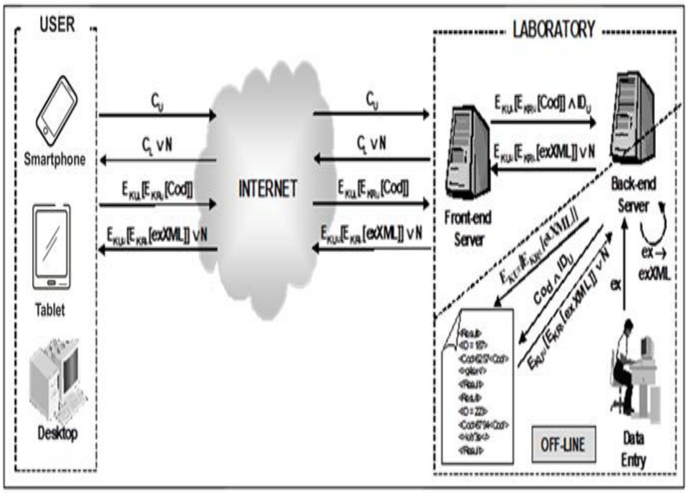

Fig. 4: Anticipatory protocol scheme

But some aspects distinguish the two protocols as it can be seen in table 1. The on-demand protocol is not affected by the user certificate expiration, the user requests to its exam result, as well as the execution of all procedures to guarantee data security, happens in on-line way. But the on-demand protocol has two weaknesses. The first one is related to the exam results storage. In this protocol, the exam results are always available in a database without being encrypted. The second weakness is a consequence of previous one: this protocol is vulnerable to intruder's attacks. By the fact that all procedures occur in on-line way, it makes attacks possible when the exams results data are being accessed and manipulated.

In the anticipatory protocol, the problem of user certificate expiration is evident, since the user's certificate is supplied previously, and it can expire before the user requests its exam results. A solution for this problem is in the use of atomic proxy concept [12]. However, the anticipatory protocol has two great advantages: it stores XML data in encrypted form and it is not vulnerable to the attacks, since that all data entry conversion process, to XML, encryption and storage of the data occurs in way off-line. To guarantee more security, these tasks can be executed with the back-end server detached from the front-end server.

Evaluation resources	**On demand protocol**	**Antecipatory protocol**
Problem of the user's certificate expiration	No	Yes
Encrypted data storage	No	Yes
Susceptibility to intruders attacks in the moment that data is being manipulated	Yes	No

Table 1: Protocol evaluation

Based on the protocol's comparison analysis, it is clear that the anticipatory protocol is safer than the on-demand protocol because it supports encrypted data storage and avoids that unauthorized people access secret data.

V. CONCLUSION

Besides of the large heterogeneous data dispersed on the Internet, it is basic the uses of a standard that allows the data are shared of simple form for any different of application. The XML standard has been adopted due to its simplicity, portability and interoperability. Nevertheless, due to individuality and criticism of determined data, for example, a HIV exam result, cancer or tuberculosis, services that guarantee confidentiality, authenticity and non-repudiation of the data are essential to save the user over eventual faults.

This paper has presented two new data exchange security protocols to XML-based application in order to solve the problems of data heterogeneity and secure. Using the XML standard and concepts of cryptography, both protocols define rules that support different advantages as according to the needs of each application. Through the two protocols evaluation, it can be concluded that the anticipatory presents safer than the on-demand protocol because it keeping the data always encrypted. Moreover, the anticipatory protocol is not vulnerable to intruder’s attacks because it manipulates private keys and plain text in way off-line.

As future works, the goal is to improve the anticipatory protocol using the concept of atomic proxy function [9] to solve the problem of user certificate expiration. In addition, it is planned to this protocol can be used in an environment of hostile machines, where an encrypted data can be manipulated without necessarily being decrypted.

REFERENCES

[1]. Eastlake III, D.E., K. Niles, Secure XML: The New Syntax for Signatures and Encryption, Addison-Wesley, Boston, 2002.

[2]. W3C, "XML Encryption Syntax and Processing", http://www.w3.org/TR/xmlenc-core/, December 10, 2014.

[3]. W3C, "Decryption Transform for XML Signature”, http://www.w3.org/TR/xmlenc-decrypt, December 10, 2014.

[4]. W3C, "XML-Signature Syntax and Processing”, http://www.w3.org/TR/xmldsig-core/, February 12, 2015.

[5]. W3C, "Extensible Markup Language (XML) 1.0 (Second Edition)”, http://www.w3.org/TR/REC-xml, February 4, 2015.

[6]. W3C, "HTML 4.01 Specification”, http://www.w3.org/TR/REC-html40, December 24, 2014.

[7]. W3C, “Canonical XML Version 1.0”, http://www.w3.org/TR/2001/REC-XML-c14n-20010315, March 15, 2015.

[8]. H. Johanston, “Sistemas de Informação Hospitalar: Presente e Futuro”, Revista Informédica, in http://www.epub.org.br/informed, São Paulo, v.1, n.2, 1993.

[9]. E-Health Latin America, “Há um Futuro Promissor na História Clínica Eletrônica”, Bibliomed, in http://www.bibliomed.com.br, November 2000.

[10]. L. Kohnfelder, “Towards a Practical Public-Key Cryptosystem”, Bachalor’s Thesis, M.I.T., May 1978.

[11]. Stallings, W., Cryptography and Network Security: Principles and Practice, Prentice Hall, New Jersey, 1999.

[12]. M. Blaze, M. Strauss, “Atomic Proxy Cryptography”, AT&T Labs-Research, February 1998.