Implementing DNA Encryption Technique in Web Services to Embed Confidentiality in Cloud

Gunjan Gugnani, S.P. Ghrera, P.K. Gupta, Reza Malekian and B.T. J. Maharaj

Abstract Hottest buzzword of this decade is "Cloud Computing"—which has been considered as one of the potential solutions to our increasing demand for accessing, processing, storing, and using provisioned resources over the internet. However, with so many boons, it comes along with some curses as security and trust issues. There are many security issues within the cloud, and Information disclosure is a big threat to a cloud user when the information is transferred over the network using web services. In this paper, we have focused on to provide confidentiality to the user while using cloud-based web services, and an approach has been proposed for selective encryption of XML elements so as to provide confidentiality and prevent XML document form improper information disclosure. The technique used for encrypting the selective XML elements is Deoxyribonucleic Acid (DNA) Encryption. Proposed technique selectively encrypts the elements of XML file using DNA sequencing.

Keywords Cloud security · Web services · Encryption · DNA sequencing · SOAP

G. Gugnani (✉) · S.P. Ghrera
Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat 173234, India
e-mail: Gugnani.Gunjan@gmail.com

S.P. Ghrera
e-mail: Spghera1@gmail.com

P.K. Gupta · R. Malekian · B.T.J. Maharaj
Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria 0028, South Africa
e-mail: Pkgupta@ieee.org

R. Malekian
e-mail: Reza.Malekian@ieee.org

B.T.J. Maharaj
e-mail: Sunil.Maharaj@up.ac.za

S.C. Satapathy et al. (eds.), *Proceedings of the Second International Conference on Computer and Communication Technologies*, Advances in Intelligent Systems and Computing 381, DOI 10.1007/978-81-322-2526-3_42

1 Introduction

Cloud Computing is a way of computing in which resources can be easily supplied to the client as a service over the internet as per the requirement of the client. The client may request any service or resource over the internet to the cloud service provider, and the cloud service provider will respond to the requested service or resource by the client. This request can be of any type for example request for some application, for processing, some storage space, etc.

There are major benefits of cloud for manageability, availability, scalability, on-demand, pay-per-use, and consolidation. Cloud resources are available over the internet, and these capabilities can be accessed through many standard mechanisms that are used by different platforms, e.g., personal computers, mobile phones, laptops, tablets, and workstations. Cloud is easily flexible and scalable according to the consumer demand. Resources can be easily scale-in and scale-out as per the consumer use. Cloud also provides measure service of pay-per-use so that the resource usages can be easily examined, managed, and controlled for both user and the provider. Cloud can easily be consolidated with any other unit with minimal effort.

There are many economic benefits as cloud promises to reduce the capital and operational expenditure. Along with these many benefits, cloud also comes along with many issues like no proper Service level Agreement (SLA) agreement, involvement of the third party that raises the trust issues, and attacks and threats that raise security concerns. There are many threat full events that compromise security of cloud and internet, e.g., Resource exhaustion—due to DOS attacks, malicious insiders—due to improper access control given to the cloud authorities, information leakage—due to disclosure of some confidential information or encryption failure, loss of encryption keys while exchanging of key, and injections made by attackers to extract or alter the information. Above all, security and trust concerns are the foremost issues in cloud as companies may hesitate to completely rely on their own records, data, information, and execution, processing jobs to some outside enterprise whose location is in different countries and makes enterprises to attach to the old local data center approach [1] as shown in Fig. 1.

Here, we have focused on the communication scenario of business transactions between client and the cloud which takes place through web services via SOAP messages in an XML document [2]. But still web services are in their infancy, and there is no proper research on the vulnerabilities of web services [3]. There are different threats like Information disclosure, leakage, and many injection attacks which could take place on an XML document over the network need to be removed from the web services. This shows that there is a huge potential for the improvement of confidentiality and integrity in the web services in cloud environment. In our work, we have embedded confidentiality using XML encryption in the cloud. XML encryption ensures not to disclose the information contained in the XML document while communicating over the network. XML Encryption grants integrity and confidentiality to the messages being transferred between the cloud user and the cloud provider. These messages can be of any type like request and response

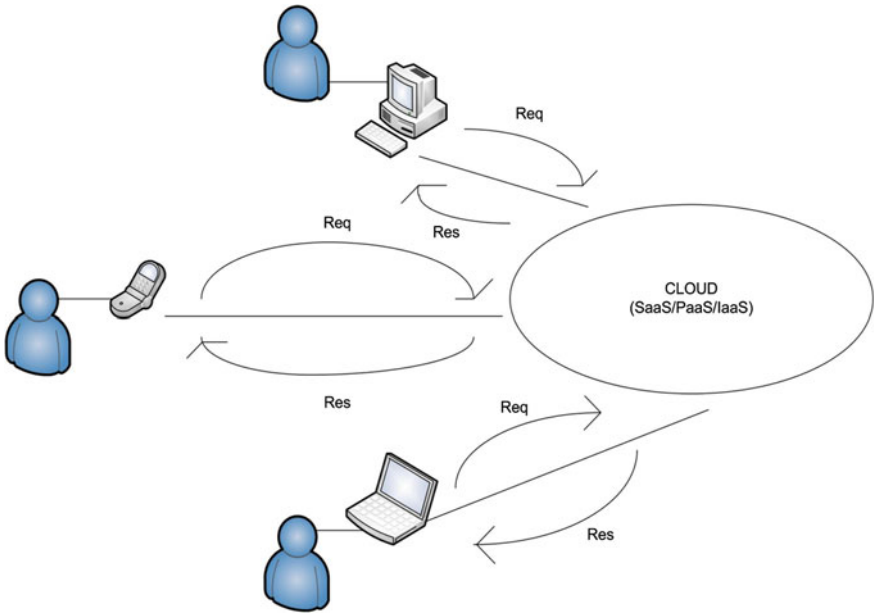

Fig. 1 Clients communicating with the cloud through various means

messages, notifications, etc. The cryptographic technique that we have used in our work is DNA encryption and decryption. DNA encryption is based on some basic but interesting features of DNA molecules which are generally formed by two biopolymer strands known as polynucleotides. Each nucleotide consists of nitrogen containing nucleo-base Adenine (A), Guanine (G), Cytosine (C), or Thymine (T). DNA encryption makes use of both these base pairing rule and also complementary pairing rule.

This paper is divided into various sections, where Sect. 2 represents the literature survey of existing frameworks, web services, and encryption and decryption techniques. Section 3 discusses about the proposed methodology and proposed implemented algorithms. Section 4 shows the result analysis of the discussed scenario and represents the results in tabular form consisting of encryption and decryption times of variable length of text. At last, Sect. 5 represents the conclusion of the work.

2 Literature Survey

Cloud Computing can thought to be an internet model that can offer a suitable and easy on-demand online network access to the pooled resources with nominal administration or any service provider interaction. Here, we have discussed about

the security of web services, methods to implement confidentiality, and about the encryption techniques.

In [4], Chou has discussed about many security threats and attacks and categorized them in two broad terms that are malware injection attacks and wrapping attacks. The malware injection attack covers cross-site scripting attacks and SQL injection attacks. Wrapping attack covers the attack on SOAP messages that break confidentiality between client and server. Subashini and Kavitha [5] have discussed about the nature of SaaS applications and emphasized on the confidentiality of SOAP messages. They have found that one of the biggest challenges which are still needed to be addressed within web services is managing the transaction. Currently, there are many standards available like WS-Transaction and WS-Reliability, but they are not mature enough as the clients are communicating to the cloud via internet; then, all communication is done through SOAP messages, so these SOAP messages are needed to be kept confidential which are in XML format. In [6], Yue-Shen et al. have described about the use of core web security technologies as XML signature and encryption. Integrity in the document can be realized through XML signature and can also be realized as an identification of authentication and focused on the requirement of the confidentiality. In [7], Thakur and Gupta have implemented the idea of dividing the data into multiple clouds according to the level of integrity and confidentiality required to them and then applied the encryption techniques like Advanced Encryption Standards (AES), Data Encryption Standards (DES), and Digital Signature Standards (DSA). Authors have also compared four of these algorithms. Liu and Lin [8] have discussed about the cryptographic technique, i.e., DNA encryption technique, and embed it in a word document to assure confidentiality. Firstly, the plain text has been encoded by a DNA sequence, and the equal length DNA sequence is created by Chebyshev maps which are used to encrypt the already considered DNA reference sequence, and then they have attached the result to the elementary DNA sequence. In the next step, they have shifted the whole DNA sequence for infinite times and inserted them into the word file by amending the fore color of the characters. In [9], Terec et al. have discussed the implementation of various cryptographic techniques in Java, Matlab, and BioJava, and also explain how DNA encryption is implemented in three of them. Based on confidentiality properties, these algorithms are used. Rana et al. [10] have represented the combined and improved framework of Infrastructure as a service and Platform as a service, and described about the simulation of key techniques like data storage technology and data management technology. In [11], Thakur et al. analyzed the performance of encryption algorithms RSA, Bcrypt, and AES to ensure the data integrity issue in SaaS cloud.

3 Proposed Methodology

All the communication between these cloud services and cloud clients takes place through web services. Web-based applications using the XML, WSDL, SOAP, and UDDI open standards over an internet protocol backbone. For tagging, the data

XML is used; SOAP is used for transferring of the data; WSDL is used for describing the available services, whereas for listing of available services, UDDI is used [12]; and to safeguard these messages, generally SSL is used.

The steps of web services architecture in cloud are as follows: (1) Firstly, the cloud client makes a request and system figures out who is the provider of the required data by contacting the UDDI. (2) Secondly, it uses the SOAP protocol and contacts the cloud service provider. (3) Further, the cloud service provider validates the client's service request and then again uses the SOAP protocol and sends structured data in an XML file to cloud service provider as shown in Fig. 2.

The whole communication among a customer and a cloud service provider over the network takes place using SOAP messages. As these messages travel through the network, they become vulnerable to different types of threats and attacks like man in the middle attack, injection attack, spoofing attacks, etc. Sometimes this conversation may contain very crucial and confidential information like passwords, credit card details, patient's data, or bank transaction information. Therefore, the cloud system requires high integrity and confidentiality. This is the reason why we need to encrypt these SOAP messages which are available in XML format. To embed the confidentiality in the interaction between client and cloud, the approach we are using here is XML DNA encryption/decryption. The proposed DNA encryption will selectively encrypt the required information and hence provides the confidentiality in the cloud environment. In XML DNA encryption, we can secure the communication among cloud and client. It will also secure the communication from improper information disclosure and information leakage over the network.

3.1 Proposed Algorithm

In this algorithm, firstly, we have extracted the required element from the XML file. In the next step, we have converted the extracted element in binary form and then assigned a binary combination to the DNA bases (A = 00, T = 01, C = 10, G = 11) to the extracted text. Furthermore, we have applied complementary pairing rule on this DNA sequence which replaces the bases with their complements (A = C,

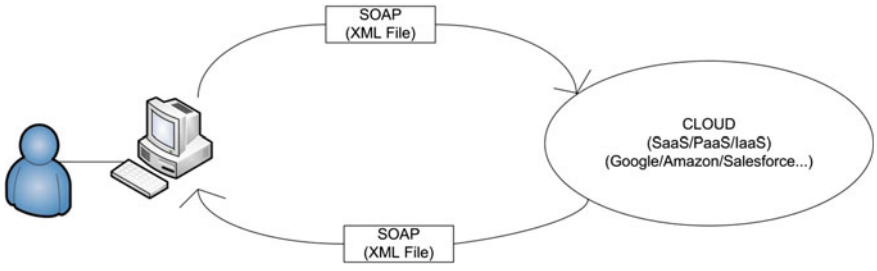

Fig. 2 Communication between cloud and client using web services via SOAP messages

Input=Selective Plain Text from XML file, DNA Reference Sequence
Output-Encrypted Text in XML file
Step1: Replace Plain Text (PT) by its binary

BinaryText ← PlainText,

Step2: Assign binary to DNA bases (A ←00, C ←01, T ←10, G ←11)
Step3: Replace binary digits of both the text and key by its DNA bases

DNASeq1 ← BinaryText

Step5: Replace DNA sequence with complementary bases (A ←C, G ←T, T ←A, C ←G)

NewDNASeq ← DNASeq1
Step6: Take the input reference sequence and replace DNASeq by the base pair occurrence number

NumericalEncrypted text ← NewDNASeq
Step7: Replace it in same XML file

Fig. 3 DNA encryption

C = G, G = T, and T = A). In the next step, we have taken a DNA reference sequence from a gene sequence database. As we know that one complete DNA sequence consists of 20 pairs of bases, we have two DNA sequences: (1) Encoded DNA sequence and (2) Actual DNA sequence from genes database. Finally, we have replaced the encoded DNA sequence base pairwise, with the number of occurrence of the corresponding base pair in the actual DNA sequence. The algorithm steps are shown in Figs. 3 and 4.

This encryption technique is very effective rather than other old encryption techniques, as now the attackers are well aware of the old techniques. In our case, the probability to judge the correct plain text is 0.0000000061 because there are approximately 163 million of DNA sequences accessible freely [13].

Input- EncryptedXML file, DNA reference Sequence
Output: plain text in XML file
Step1: Take encrypted text from XML file
Step2: Replace encrypted text by DNA bases according to the reference sequence

DNASeq1 ← Encrypted text
Step3: Replace DNASeq by its Complements

DNASeq2 ← DNASeq1

Step4: Assign binary to DNA bases (A ←00, C ←01, T ←10, G ←11)
Step5: Convert DNA text to binary

BinaryText ← DNASeq2
Step6: Convert Binary to its actual text

Plain Text ← BinaryText
Step7: Replace the plain text in XML document

Fig. 4 DNA decryption

4 Results

The encryption technique as discussed in Sect. 3.1 to embed security in web services has been implemented in NetBeans IDE using Java, and to provide the cloud environment, CloudSim is integrated within the NetBeans. As the user provides the details of ATM card, these details are submitted to the cloud using web services via SOAP file. The structure SOAP file is as follows:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap:Envelope
xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  ……
    <soap:Body>
  ……
        <AccountPin>1234</AccountPin>
  ……
    </soap:Body>
  </soap:Envelope>
```

According to the approach, selective encryption has to take place to encrypt the confidential information to increase the performance of the system. In the given file, the ATM pin needs to remain confidential to secure the transaction of the user; therefore, we have encrypted the element AccountPin. After selective DNA encryption, the file will be as follows:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap:Envelope
xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  ……
    <soap:Body>
  ……
        <AccountPin>201420182020209</AccountPin>

  ……
    </soap:Body>
  </soap:Envelope>
```

The steps of DNA encryption performed over the provided ATM pin are represented in Table 1. We have also computed the average encryption and average decryption time for the given text with different bit lengths of the text. It is observed that with the increase in length of the text to be encrypted, the required encryption time increases proportionally.

The other observation is that the decryption time is little more than that of encryption time. The average encryption/decryption time for different bit lengths of the text is shown in Fig. 5.

The DNA reference sequence dataset used in our work is taken from Berkeley Drosophila Genome Project (BDGP) site. The dataset file contains set of contiguous

Table 1 DNA encryption steps

S. no.	Input	Output
1	XML-SOAP File	1234 (selected element)
2		00110001001100100011001100110100 (binary conversion)
3		AGATAGACAGAGAGTA (G = 11, C = 10, T = 01, A = 00)
4		CTCACTCGCTCTCTAC (A = C, C = G, G = T, T = A)
5	DNA Reference: $[TA_1]$, $[GC_2]$, $[TG_3]$, $[AG_4]$, $[CT_5]$, $[CT_6]$, $[TT_7]$, $[TG_8]$, $[AC_9]$, $[TC_{10}]$, $[TC_{11}]$, $[TA_{12}]$, $[AT_{13}]$, $[CA_{14}]$, $[CC_{15}]$, $[CC_{16}]$, $[TC_{17}]$, $[CG_{18}]$, $[TG_{19}]$, $[CT_{20}]$	201420182020209 (encrypted text)

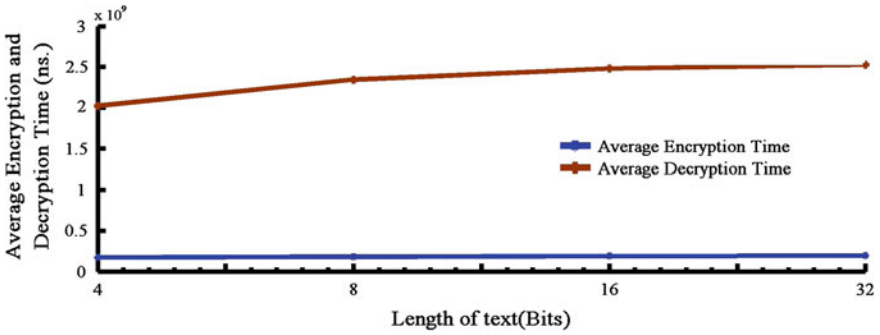

Fig. 5 Average time required for encryption and decryption

DNA sequences and the file was in Fasta format. There are approximately 163 million of DNA sequences which are accessible freely, so it is very complex to recover the plain text from the encrypted text. Moreover, this technique gives the option to selectively encrypt only the confidential information rather that the whole information as in the case of SSL.

5 Conclusion

A DNA-based encryption/decryption technique has been proposed to embed the confidentiality in the communication between client and the cloud service provider. The advantage of the technique is that it is very difficult to decrypt a DNA ciphertext without knowing the correct reference DNA sequence. We have implemented this technique on a SOAP file and then computed the average

encryption and decryption time for the selected text that needed to be encoded. This technique will embed confidentiality in the cloud environment with better effective performance as compared to SSL, because it gives us the option to selectively encrypt the confidential information rather than encrypting the whole information which is an overhead in SSL.

Acknowledgments This work was supported by the National Research Foundation of South Africa (NRF) grant funded by the South African government (No. KIC 150323115773).

References

1. Jensen, M., Schwenk, J., Gruschka, N., Iacono, L. L.: On technical security issues in cloud computing. In: IEEE International Conference on Cloud Computing, pp. 109–116 (2009)
2. Saravanaguru, R.A., Abraham, G., Ventakasubramanian, K., Borasia, K.: Securing web services using XML signature and XML encryption. arXiv preprint arXiv:1303.0910, 1–6 (2013)
3. Moradian, E. Anne H.: Possible attacks on XML web services. Int. J. Comput. Sci. Netw. Sec. **6.1B**, 154–170 (2006)
4. Chou T.S.: Security threats on cloud computing vulnerabilities. Int. J. Comput. Sci. IT, **5**, 79–88 (2013)
5. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. **34**, 1–11 (2011)
6. Gu, Y.S., Ye, M.T., Gan, Y.: Web services security based on XML signature and XML encryption. J. Netw. **5**, 1092–1097 (2010)
7. Thakur, A.S., Gupta, P.K.: Framework to improve data integrity in multi cloud environment. Int. J. Comput. Appl. **87**, 28–32 (2014)
8. Liu, H., Lin, D., Kadir, A.: A novel data hiding method based on deoxyribonucleic acid coding. Comput. Electr. Eng. **39**, 1164–1173 (2013)
9. Terec, R., Vaida, M.F., Alboaie, L., Chiorean, L.: DNA security using symmetric and asymmetric cryptography. Int. J. New Comput. Archit. Their. Appl. **1**, 34–51 (2011)
10. Rana, P., Gupta, P.K., Siddavatam, R.: Combined and improved framework of infrastructure as a service and platform as a service in cloud computing. In: Babu, B.V., Nagar, A., Deep, K., Pant, M., Bansal, J.C., Ray, K., Gupta, U. (eds.) SocPros 2012. LNCS, vol. 236, pp. 831–839. Springer, India (2014)
11. Thakur, A.S., Gupta, P.K., Gupta, P.: Handling data integrity issue in SaaS cloud. In: Satapathy, S.C., Biswal, B.N., Udgata, S.K., Mandal, J.K. (eds.) FICTA 2014. LNCS, vol. 328, pp. 127–134. Springer International Publishing (2015)
12. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services. Springer, Berlin Heidelberg (2004)
13. Shiu, H.J., Ng, K.L., Fang, J.F., Lee, R.C., Huang, C.H.: Data hiding methods based upon DNA sequences. Inf. Sci. **180**, 2196–2208 (2010)