AJCSE
ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

www.ajcse.info

Asian Journal of Computer Science Engineering 2017; 2(6)-01-05

REVIEW ARTICLE

A FHE Algorithm used in Multi-Format Data in Collaborated Multi cloud Computing

***Mekala Bhaskar[1], Dr. M.Ashok[1], Dr. K.E.BalaChandrudu[1]**

[1]*CSE Dept., Malla Reddy Institute of Eng & Technology, JNTUH, INDIA*

Received on: 30/04/2017, Revised on: 18/07/2017, Accepted on: 01/10/2017

ABSTRACT

Now a day's health care data is growing vastly in the world, this health care data has to be stored and managed in proper format by the health care centers. Most of the work has been done in the area of cloud computing and still more work has to be done. But the main issue in multi cloud computing is providing security to the data. My research proposal is more concentrating on security issues and implementation aspects of security problems for multi-format data in healthcare multi cloud computing. The various security issues related to data security, privacy, confidentiality, integrity and authentication needs to be addressed. Most of the health care data cloud service provider stores the data in heterogeneous text format and user need to use their own encryption algorithm to secure their data if required. The data needs to be decrypted whenever it is to be processed. The encryption of remotely stored data has been the most widely used technique to bridge this security gap. The speculated vast usage of Cloud Computing solutions for data storage and with Big Data Analytics gaining strong foothold; the security on cloud is still at big risk. Fully Homomorphism Encryption is a good basis to enhance the security measures of un-trusted systems or applications that stores and manipulates sensitive data. The model is proposed on cloud computing which accepts encrypted inputs and then performs blind processing to satisfy the user query without being aware of its content, whereby the retrieved encrypted data can only be decrypted by the user who initiates the request. This allows clients to rely on the services offered by remote applications without risking their privacy.

Keywords: Data security; Cloud computing; Fully Homomorphism encryption; healthcare; security; privacy, Cloud computing.

INTRODUCTION

Cloud Computing has been the most promising innovation in the computing world in past decade. Its usage is still hindered by the security concerns related with critical data. The data is stored in DynamoDB of Amazon Web Service (AWS) public cloud. User's computation is performed on encrypted data in public cloud.

When results are required they can be downloaded on client machine. In this scenario user's data is never stored in heterogeneous text on public cloud. Security is major concern to the cloud computing. There is strong thrust to provide security at infrastructure - network level, Host level, application level and data. The data is associated with each level like network, host and Application level. In this paper security of cloud data at rest is focused. Cloud computing uses several technologies.

The information technology model for computing, which is composed of all the IT components (hardware, software, networking, and services) that are necessary to enable development and delivery of cloud services via the Internet or a private network. The prominent actors in Cloud Computing are Cloud Provider and Cloud User. Cloud Provider is the enterprise vendoring cloud services.

A Cloud User can vary from organizations, educational institutes to individual utilizing the cloud services. Cloud providers like: IBM, Google and Amazon use the virtualization in their Cloud platform, and in the same machine can coexist the storage space and treatment virtualized which belong to the concurrent enterprises. The aspect of security and confidentiality must intervene to protect the data from each of the enterprises.

Cloud Security

***Corresponding Author:** Mekala Bhaskar, **Email**: bhaskarmekala0206@gmail.com

AJCSE, Nov-Dec, 2017, Vol. 2, Issue 6

At present both in Public Cloud and Private Cloud; security ensures to encrypt the data stored. Also it is very easy to have secure transmission from a local machine to a cloud data store. The stored data being encrypted and the channel of data transmission well secured with key exchanges. But actually performing computations on that data stored in the cloud requires decrypting it first; this makes critical data available to the cloud provider. Data Mining and other Data Analysis onto the Encrypted Database is a far distant thing to achieve by using encryption standards available. The proposal here is to encrypt data before sending to the cloud providers. Thereby to enable a cloud computing vendor to perform computations on clients' data at their request, such as analyzing sales patterns, without exposing the original data. To achieve this it is also necessary to hold the cryptosystems based on Homomorphic Encryption either a Fully Homomorphic Encryption (FHE) or Somewhat Homomorphic Encryption (SHE).

Homomorphic Cryptosystems

Now we will see easy example of Homomorphic encryption Algorithm it explains clearly how it works. They are ones where mathematical operations on the cipher text have regular effects on the plaintext. A very simple demonstration of the mathematical consistency required: A user sends a request to add the numbers A=10 and B=20, which are encrypted to become the numbers 40 and 55, respectively. The server in the cloud processes the sum as 95, which is downloaded from the cloud and decrypted to the final answer, 30.A normal symmetric cipher -- DES, AES is not Homomorphic. The RSA algorithm is Homomorphic but only with respect to multiplication.

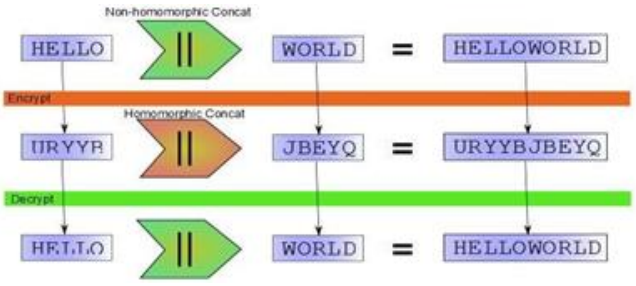
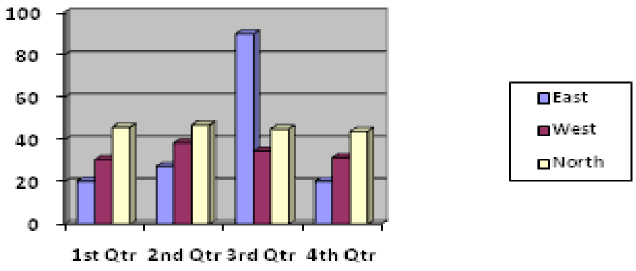


Figure 1. A string concate example of Homomorphic Encryption

At first, the notion of processing data without having access to it may seem paradoxical, even logically impossible. To convince you that there is no fallacy, and to give you some intuition about the solution, let us consider an analogous problem in the physical world. Sita owns a jewelry store. She has raw precious materials gold, diamonds, silver, etc. She wants her workers to assemble into intricately designed rings and necklaces. But she distrusts her workers and assumes that they will steal her jewels if given the opportunity. In other words, she wants her workers to process the materials into finished pieces, without giving them access to the materials. For that she uses a transparent impenetrable glove box, secured by a lock for which only she has the key. She puts the raw precious materials inside the box, locks it, and gives it to a worker. Using the gloves, the worker assembles the ring or necklace inside the box. Since the box is impenetrable, the worker cannot get to the precious materials, and ures he might as well return the box to Sita, with the finished piece inside. Sita unlocks the box with her key and extracts the ring or necklace. In short, the worker processes the raw materials into a finished piece, without having true access to the materials. Of course, Sita's jewelry store is only an analogy.

Implementation of Fully Homomorphic Encryption Algorithm

The implementation of a cryptosystem supporting arbitrary computations on encrypted bits. This includes the implementation of a somewhat Homomorphic scheme supporting a limited number of operations on encrypted bits, and its extension to a fully Homomorphic scheme by providing a Homomorphic version of the decryption function. New researches in the field of Homomorphic encryption schemes have made it possible to implement a variety of schemes using different techniques and programming languages. In 2009 Craig Gentry of IBM has proposed the first encryption system “Fully Homomorphic” that evaluates an arbitrary number of additions and multiplications and thus calculates any type of function on encrypted data. The internal working of this adds another layer of encryption every few steps and uses an encrypted key to unlock the inner layer of scrambling. This decryption "refreshes" the data without exposing it, allowing an infinite number of computations on the same.



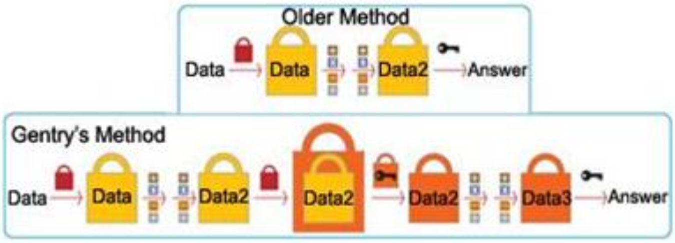


Figure 2. Craig Gentry implementation of FHE.

Fully Homomorphic Encryption Algorithm on Cloud

The application of fully Homomorphic encryption is an important brick in Cloud Computing Security; more generally, outsourcing of the calculations on confidential data to the Cloud server is possible, keeping the secret key that can decrypt the result of calculation.

In our implementation, we analyze the performance of existing Homomorphic encryption cryptosystems, and are working on a virtual platform as a Cloud server, a VPN network that links the Cloud with the customer, and then simulating different scenarios.

For example a Database-Server communicating with Client using FHE Cryptosystem is as shown in the figure below.

Encryption schemes that support operations on encrypted data (aka Homomorphic encryption) have a very wide range of applications in cryptography.

This concept was introduced by Rivest et al. shortly after the discovery of public key cryptography, and many known public-key cryptosystems support either addition or multiplication of encrypted data. However, supporting both at the same time seems harder, and until very recently all the attempts at constructing so-called “fully Homomorphic” encryption turned out to be insecure.

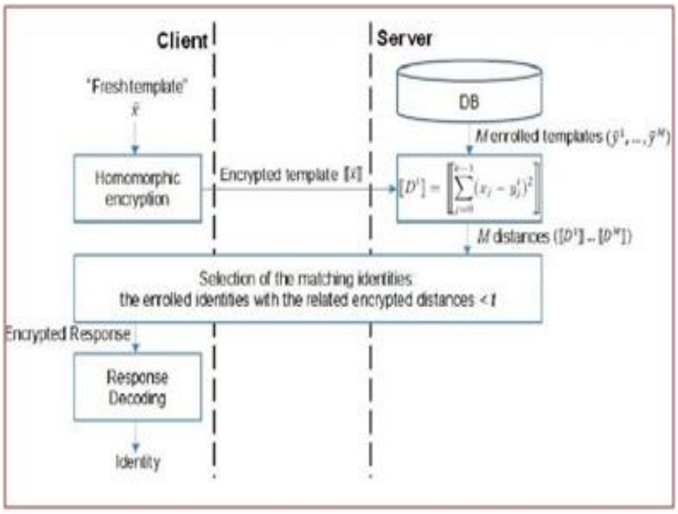
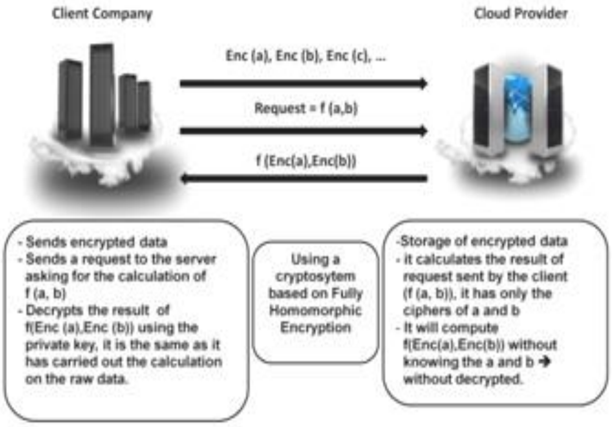

Client
Server
"Freshtemplate"
$\vec{x}$
DB
M enrolled templates $(\vec{y}^1, \ldots, \vec{y}^M)$
Homomorphic encryption
Encrypted template $[\vec{x}]$
$[D^i] = \left[\sum_{j=0}^{k-1}(x_j - y_j^i)^2\right]$
M distances $([D^1] \ldots [D^M])$
Selection of the matching identities
the enrolled identities with the related encrypted distances < t
Encrypted Response
Response Decoding
Identity

Figure 3. A Database-Sever & Client implementing Homomorphic Encryption

On similar lines; the Cloud Computing scenario can be illustrated as below:


Client Company
Cloud Provider
Enc (a), Enc (b), Enc (c), ...
Request = f (a,b)
f (Enc(a),Enc(b))
- Sends encrypted data
- Sends a request to the server asking for the calculation of f (a, b)
- Decrypts the result of f(Enc (a),Enc (b)) using the private key, it is the same as it has carried out the calculation on the raw data.
Using a cryptosytem based on Fully Homomorphic Encryption
-Storage of encrypted data
- it calculates the result of request sent by the client (f (a, b)), it has only the ciphers of a and b
- It will compute f(Enc(a),Enc(b)) without knowing the a and b → without decrypted.

Challenges of Fully Homomorphic Encryption Algorithm on Cloud

The double layer of encryption causes the system runs too slowly for practical use. We are working on optimizing the same for specific applications such as searching databases for records reduce the time complexity. Also to trust a very new encryption scheme for confidentiality is not feasible and it requires considerable (~10 yrs) of usage exposure. A team from MIT’s Computer Science and Artificial Intelligence Laboratory, who worked in conjunction with the University of Toronto and Microsoft Research, sought to combine multiple schemes to solve these challenges. The system starts with Homomorphic encryption, with a decryption algorithm embedded in a garbled circuit which is itself protected by attribute-based encryption this ensures the process stays encrypted.

CONCLUSION AND FUTURE WORK

Security of cloud computing based on fully Homomorphic encryption is a new concept of security which is to enable to provide the results of calculations on encrypted data without knowing the raw entries on which the calculation was carried out respecting the confidentiality of data. Our work is based on the application of fully Homomorphic encryption to the security of Cloud Computing: a) Analyze and improve the existing cryptosystem to allow servers to perform various operations requested by the client. b) Improve the complexity of the Homomorphic encryption algorithms and study the response time to requests according to the length of the public key. Homomorphic Encryption will bring a new dimension to cloud storage. It provides confidentiality to the data as in no stage data is exposed in heterogeneous text. The proposed algorithm is simplified, efficient version applied in AWS public cloud. The proposed algorithm can be used for various applications such as online auctioning, medical purposes and business purposes. There is need to carry out research in reducing the size of cipher text for efficient data processing. There is also a need to evolve various algorithms for searching and querying on encrypted data under FHE scheme.

REFERENCES

1. Tebaa, M.; El Hajji, S.; El Ghazi, A., "Homomorphic encryption method applied to Cloud Computing," in Network Security and Systems (JNS2), 2012 National Days of , vol., no., pp.86-89, 20-21 April 2012.
2. Mather, Tim, Subra Kumaraswamy, and Shahed Latif. Cloud security and privacy: an enterprise perspective on risks and compliance. " O'Reilly Media, Inc.", 2009
3. Samyak Shah, Yash Shah, Janika Kotak, “Somewhat Homomorphic Encryption Technique with its Key Management Protocol”, Dec 14 Volume 2 Issue 12 , International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 4180 – 4183.
4. Ramaiah, Y. Govinda, and G. Vijaya Kumari. "Efficient public key homomorphic encryption over integer plaintexts." Information Security and Intelligence Control (ISIC), 2012 International Conference on. IEEE, 2012.
5. Gentry, Craig. "Computing arbitrary functions of encrypted data." Communications of the ACM 53.3 (2010): 97-105.
6. Atayero, Aderemi A., and Oluwaseyi Feyisetan. "Security issues in cloud computing: The potentials of homomorphic encryption." Journal of Emerging Trends in Computing and Information Sciences 2.10 (2011): 546-552.
7. Catteddu, Daniele, and Giles Hogben. "Cloud computing." Benefits, Risks and Recommendations for Information Security/European Network and Information Security Agency, ENISA (November 2009) (2009).
8. Deyan Chen; Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," in Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on , vol.1, no., pp.647-651, 23-25 March 2012
9. Pearson, Siani. "Taking account of privacy when designing cloud computing services." Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE Computer Society, 2009.
10. Rivest, Ronald L., Len Adleman, and Michael L. Dertouzos. "On data banks and privacy homomorphisms." Foundations of secure computation 4.11 (1978): 169-180.
11. Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009 http://crypto.stanford.edu/craig/craig-thesis.pdf.
12. Understanding Homomorphic Encryption - http://en.wikipedia.org/wiki/Homomorphic_encryption
13. FHE implementation with garbled circuit - http://eprint.iacr.org/2010/145.pdf
14. New encryption method promises end-to-end cloud security, by Kevin McCaney Jun 13, 2013 - http://gcn.com/Articles/2013/06/13/Encryption-end-to-end-cloud-security.aspx?Page=1
15. Homomorphic Encryption Applied to the Cloud Computing Security by Maha TEBAA, Saïd EL Hajji, Abdellatif EL Ghazi - http://www.iaeng.org/publication/WCE2012/WCE2012_pp536-539.pdf
16. Vic (J.R.) Winkler, ”Securing the Cloud, Cloud Computer Security, Techniques and Tactics”, Elsevie r
17. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic , volume 1592, 1999
18. Julien Bringe and al. An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, Springer-Verlag , 2007. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2):120-126, 1978. Computer Science, pages 223-238. Springer, 1999.
19. Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 469-472, 1985.

20. WiebBosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. J. Symbolic Comput., 24(3-4): 235 -265, 1997. Computational algebra and number theory, London, 1993.
21. Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On Data Banks and Privacy Homomorphisms, chapter On Data Banks and Privacy Homomorphisms, pages 169-180. Academic Press, 1978.
22. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Theory of Crypto gr aphy Conference, TCC'2005, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.