International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference on Communication Computing & Virtualization 2016

Intelligent Homomorphic Encryption for Cloud Data

Manish M Potey
Associate Professor
K J Somaiya College of Engineering, Mumbai

C. A Dhote, PhD
Professor
PRM Institute of Technology & Research, Badnera, Amravati

Deepak H Sharma
Associate Professor
K J Somaiya College of Engineering, Mumbai

ABSTRACT

As cloud computing represents a new computing model, there is a great deal of uncertainty about how security at all levels can be achieved. Data security becomes more important when using cloud computing at all "levels" is addressed.

Users are allowed to store large amount of data on cloud storage. The various security issues related to data security, privacy, confidentiality, integrity and authentication needs to be addressed. Most of the cloud service providers store the data in plaintext format and user need to use their own encryption algorithms to secure their data if required. The data needs to be decrypted whenever it is to be processed. This paper focuses on storing data on the cloud in the encrypted format using fully homomorphic encryption and proposes an extension to partial homomorphic scheme. It results in low size cipher text and choice is taken from the user. Intelligence has beenproposed to this scheme by the selecting automatically any of the algorithms – partial, fully or small cipher text size depending on the data or application.

Keywords

Data security; Cloud computing; Fully Homomorphic encryption; Public Cloud; Cipher text.

1. INTRODUCTION

Security is major concern to the cloud computing. There is strong thrust to provide security at infrastructure - network level, Host level, application level to the 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 security issues related to different type attacks related to several technologies needs to be addressed. Some security issues discussed in [6][7] regarding cloud computing are -

- Availability –availability of data is an important security issue. Whenever it is required it must made available to user. Also user must have control over its data. Availability issue needs to attend, when service is required from another cloud service provider. There are presently three major threats to availability. The first threat is network based attack [2]. The second threat is cloud service providers availability and third backup of stored data by cloud service provider. There is need to provide effective and efficient techniques for access control, authentication and authorization of significant data.

- Data remanence - It is an issue when data gets exposed after deletion to the unauthorized party. A data security lifecycle [8] refers to the entire process from data creation to disposalis shown in Fig. 1. Care must be taken when the data needs to be removed.

Third-Party Control- Cloud Service provider is managing the user data. Third party access may lead to leakage of sensitive information and trade secrets.

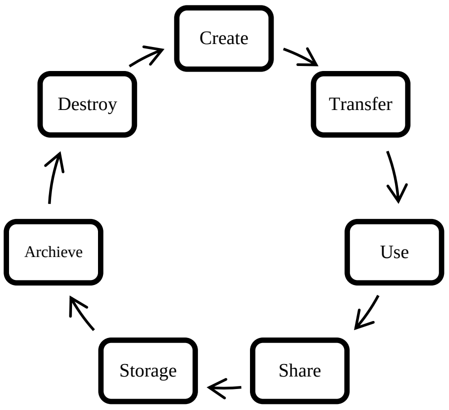

Fig. 1 Data Life Cycle

- Legal Issues and Privacy- User is unaware about where datais stored in cloud. In each country cyber laws are different. There is great concern about legality, confidentiality of data. User is also concerned about its data privacy. Major privacy issues related to cloud computing are sighted in [9] by Pearson.

- DataLineage–In cloud computing the data in cloud is moving from one location to other. Following the path of data is known as data lineage [2], and it is important for an auditor's assurance.

- Data provenance – Provenance means not only that the data has integrity, but also that is computationally accurate that data was accurately calculated.

Generally when data is encrypted it is not easily understood by unauthorized people and to get plain text back decryption is used. For any kind of computation one needs to perform the decryption first. Encryption solves major issues. But the power of cloud can be exploited if user is able to carry out computation on encrypted data.

Homomorphic Encryption technique enables computing with encrypted data. It means, one is able to perform the operations on this data without converting into the plaintext. Data is in encrypted state in its most of the stages on the cloud.Homomorphic Encryption (HE) technique allows user to perform multiple types of operations on encrypted data. Only one kind of operation is allowed in a partially homomorphic encryption technique [3]. Fig. 2 shows the proposed system.

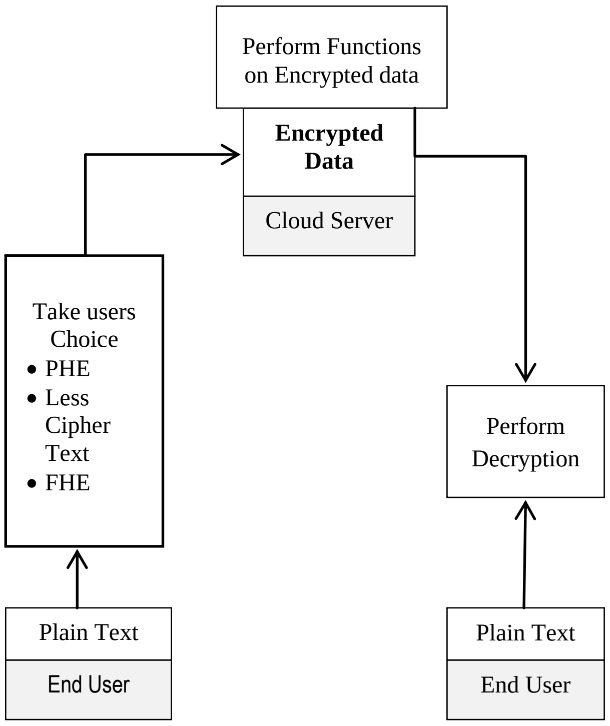

Fig 2 Proposed Scheme

The proposed system takes input from the user in plain text form. Depending on the user's choice the plain text is encrypted by using any of the three algorithms – Partial Homomorphic Encryption (PHE), Fully Homomorphic Encryption (FHE), and Low cipher text HE. The experimentation of Low size cipher text is in process. It is also proposed the program will not take such choice from the user about the selection of HE algorithm but it will be automatically selected by the program depending on level of confidentiality required to that data. Such an intelligence to this homomorphic scheme is under experimentation.

The paper is divided in five sections. Section 2 gives brief outline about various homomorphic encryption algorithms. Section 3 explains the proposed scheme. The implementation of FHE is explained in section 4, the paper conclusion and future work is in section 5.

2. RELATED WORK

The first homomorphism suggested by Rivest, Adleman and Dertouzos in [10]. Multiplicative homomorphism is given by RSA [11]. Partial homomorphic encryption scheme is suggested by Yao [12], Goldwasser and Micali [13], ElGamal [14] and Paillier [15]. Fontaine & Galand has presented a survey of homomorphic encryption schemes in [16]. Gentry from IBM have proposed fully homomorphic encryption in [5]the thesis and paper [17].

Many researchers proposed the variants of Gentry's model with some improvement. Homomorphic encryption on smaller size cipher text is proposed by [18] Smart and Vercauteren. The arithmetic operations over integers are proposed by Dijk, Gentry, Halevi, and Vaikuntanathan [19]. Faster improvement to Gentry's model is proposed by Stehle and Steinfield [20]. Y Govinda Ramaiah [4] has proposed "Efficient Public Key Homomorphic Encryption over Integer Plaintexts"

3. FULLY HOMOMORPHIC ENCRYPTION SCHEME

Algorithm

1. Choose J (64bit), K(16bit) D and F (256-bit) randomly
2. Choose 4 bit random integer K' Compute P0 = JD and P1 = JF + KK'
3. Accept Number N from user
4. P2 = [T1 P1] mod P0
5. Perform Encryption Cipher Text C = [N + T2 P2] mod P0 (T1 , T2 are a 4-bit random integer)
6. Decryption N = (C mod J) mod K

The above algorithm shows the proposed scheme to perform fully homomorphic algorithm. This scheme is simplified and efficient version of [4][19] applied in AWS public cloud for security of users data. (J, K) represent a secret Key and (P0, P1) forms a public key. Number N to be encrypted is accepted as user input.

For Example-

Input is given as J=14883982794894487223, K=43321 and number to be encrypted N=9

Then D and F are calculated as

D=70677186543966147614195862042065680704217811307 17093882368081797246007877074 7 and

F=73039047329961611877474622320644292204439326844 74778307067680690428757824363 9

Consider four bit number K' = 12 then compute

P0=10519580285119403059293206075655334278355741466 40991376691781227504638919782237324865124737665581 and

P1=10871119238146327664815812416970040873377501996 78917794234945876512572829144575038603913867044349

Perform Encryption and get

C=35153895302692460552260634131470659502176053037 926417543164649007933909362337713738789129378768 9.

Decryption is performed and get back plain text N=9

On the similar line we can be able to implement partial homomorphic encryption. It is also proposed to reduce the size of cypher text. The algorithm for the same is under experimentation and validation using mathematical approach is under consideration. Intelligence to this algorithm is given by providing automatic selection of choice for homomorphic encryption as per the data or application need. Such experimentation is in process. The next section explains about the implementation of fully homomorphic encryption.

4. IMPLEMENTATION

The user can connect to the AWS DynamoDB service through the Eclipse IDE for Java EE Developers. This allows the user to login based on his credentials and then the user can perform operations on their data based on requirements. Once user is done with all the tasks, it can opt to exit the system.

The following are the steps performed for the implementation

- Create a DynamoDB instance on AWS
- Create Database Tables with proper schema
- Get the credentials from AWS and perform access controls
- Install Eclipse Kepler version and Java SDK on it (Fig. 4)

After the installation of AWS SDK on Eclipse framework the user is available with all the needed packages.

- Follow the steps given in [21]AWS SDK.
- Exit Java Code

The Java Code that is built to interact with the DynamoDB. It runs on this eclipse platform. All the interaction needed that is Data manipulations such as Addition, Subtraction, or check balance in the data base is performed using this platform. The user logs in to the system using the interface provided and then performs functionalities provided.

4.1 Modules Involved

1. Client Machine- The requests is made to access the data from the cloud server through the client machine.
2. Login- The login component helps the user to login to the system without correct username and password which are validated by the server before the user gets access to the system.
3. Key Selection- This component selects the key based on the user that has logged into the system for encryption and decryption of data.
4. Query- Once the user has logged in, he will select the operation to be performed.
5. Computations - Based on the operation selected, computations are performed and passed on to the encryption component to be stored.
6. Encrypt and store- This component encrypts the data given by the user, or the data which has been computed by the system and thereby stores/ updates the value in the cloud database.
7. Retrieve and decrypt- This component retrieves the data required by the user from the cloud database and thus presents it to the user on the client machine.
8. AWS Cloud (DynamoDB)- This is the cloud database where all the data is stored and is accessed by the login module for the verification of details of the user, by the Key selection component to retrieve the keys stored in the database, by the encryption component to store data in encrypted format, by the decryption component to retrieve the data and decrypt it and also by computation to perform operations on the user data as per requirement or the query fired.

5. CONCLUSION AND FUTURE WORK

Homomorphic Encryption will bring a new dimension to cloud storage. It provides confidentiality to the data because at no stage data is exposed in plain text.The proposed intelligent homomorphic encryption scheme can be used for various applications such as online auctioning, medical purposes and business purposes.

Low cipher text size homomorphic encryption algorithm will save storage space and it will useful for efficient data processing. This scheme strengthens the confidentiality goal of security.

6. 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. A fully homomorphic encryption scheme.Diss. Stanford University, 2009.

[6] Potey, Manish M., C. A. Dhote, and Deepak H. Sharma. "Cloud Computing-Understanding Risk, Threats, Vulnerability and Controls: A Survey." International Journal of Computer Applications 67.3 (2013).

[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] Rivest, Ronald L., Adi Shamir, and Len Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21.2 (1978): 120-126.

[12] A. C. Yao. Protocols for secure computations (extended abstract). In 23rd Annual Symposium on Foundations of Computer Science (FOCS '82), pages 160-164.IEEE, 1982.

[13] Goldwasser, Shafi, and Silvio Micali. "Probabilistic encryption." Journal of computer and system sciences 28.2 (1984): 270-299.

[14] ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms."Advances in cryptology. Springer Berlin Heidelberg, 1985..

[15] Paillier, Pascal. "Public-key cryptosystems based on composite degree residuosity classes." Advances in cryptology—EUROCRYPT'99. Springer Berlin Heidelberg, 1999..

[16] Fontaine, Caroline, and Fabien Galand. "A survey of homomorphic encryption for non-specialists." EURASIP Journal on Information Security 2007 (2007): 15.

[17] Gentry, Craig. "Fully homomorphic encryption using ideal lattices."STOC.Vol. 9. 2009.

[18] Smart, Nigel P., and Frederik Vercauteren. "Fully homomorphic encryption with relatively small key and cipher text sizes." Public Key Cryptography–PKC 2010.Springer Berlin Heidelberg, 2010.420-443.

[19] Van Dijk, Marten, et al. "Fully homomorphic encryption over the integers." Advances in cryptology–EUROCRYPT 2010.Springer Berlin Heidelberg, 2010.24-43.

[20] Stehle, Damien, and Ron Steinfeld. "Faster fully homomorphic encryption."Advances in Cryptology-ASIACRYPT 2010.Springer Berlin Heidelberg, 2010.377-394.

[21] AWS Toolkit For Eclipse, http://docs.amazonaws.cn/en_us/AWSToolkitEclipse/latest/GettingStartedGuide/aws-tke-gsg.pdf?