Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 79 (2016) 175 – 181

Procedia Computer Science

7th International Conference on Communication, Computing and Virtualization 2016

Homomorphic Encryption for Security of Cloud Data

Mr. Manish M Potey[a]* , Dr C A Dhote[b], Mr Deepak H Sharma[c]

[a]*K J Somaiya College of Engineering,Vidyavuhar(East), Mumbai,400077*
[b]*Prof. Ram Meghe Institute of Technology & Research, Badnera, Amravati*
[c]*K J Somaiya College of Engineering,Vidyavuhar(East), Mumbai,400077*

Abstract

Cloud computing is a broad and diverse phenomenon. Users are allowed to store large amount of data on cloud storage for future use. The various security issues related to data security, privacy, confidentiality, integrity and authentication needs to be addressed. Most of the cloud service provider stores the data in plaintext 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. This paper focuses on storing data on the cloud in the encrypted format using fully homomorphic encryption. 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 users data is never stored in plaintext on public cloud.

Peer-review under responsibility of the Organizing Committee of ICCCV 2016

Keywords: Data security; Cloud computing; Fully Homomorphic encryption; AWS; DynamoDB; public cloud

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 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.

* Corresponding author. Tel.: +91-22-66449191 fax: +91-22-21025272.
E-mail address: manishpotey@somaiya.edu

Peer-review under responsibility of the Organizing Committee of ICCCV 2016
doi:10.1016/j.procs.2016.03.023

The security issues related to different type attacks related to several technologies needs to be addressed. Some security issues in cloud computing includes[6]:

• 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 refers to the entire process from data creation to destruction[8] is shown in Fig. 1. Care must be taken when the data needs to be destroyed.

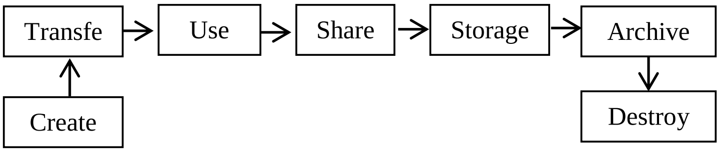

Fig.1. Data Life Cycle

• Third-Party Control- Cloud Service provider is managing the user data. Third party access may lead to leakage of sensitive information and trade secrets. There is also great threat to corporate spying. It should not also create a situation that user has to depend on a one such cloud service provider.

• Legal Issues and Privacy- User is unaware about where data is 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 by Pearson[9]

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.

Fully Homomorphic Encryption (FHE) 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

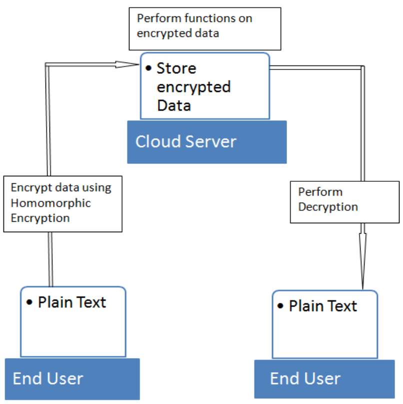

Fig. 2 FHE proposed system

The paper is categorised in five sections. Section 2 gives brief outline about various homomorphic encryption algorithms. Section 3 explains the proposed scheme for Fully Homomorphic Encryption (FHE). The core part of the paper, FHE over AWS Dynamo DB, 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 his 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

The flowchart in fig. 3 shows the proposed scheme to perform fully homomorphic algorithm. This scheme is simplified and efficient version of [18][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=706771865439661476141958620420656807042178113071709388236808179724600787770747 and
F=73039047329961611877474622320644292204439326844747783070676806904287578243639
Consider four bit number K' = 12 then compute
P0=1051958028511940305929320607565533427835574146640991376691781227504638919782237324865124737665581 and
P1=1087111923814632766481581241697004087337750199678917794234945876512572829144575038603913867044349
Perform Encryption and get
C=351538953026924605522606341314706595021760530379264175431646490079339093623377137387891293787689.
Decryption is performed and get back plain text N=9

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

Step 1: Create a DynamoDB instance on AWS
Step 2: Create Database Tables with proper schema
Step 3: Get the credentials from AWS and perform access controls
Step 4: 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.
Step 5: Follow the steps given in[23]AWS SDK.

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. The user is allowed to access only the rights that are given to him by the database owner.

Step 6: Exit Java Code

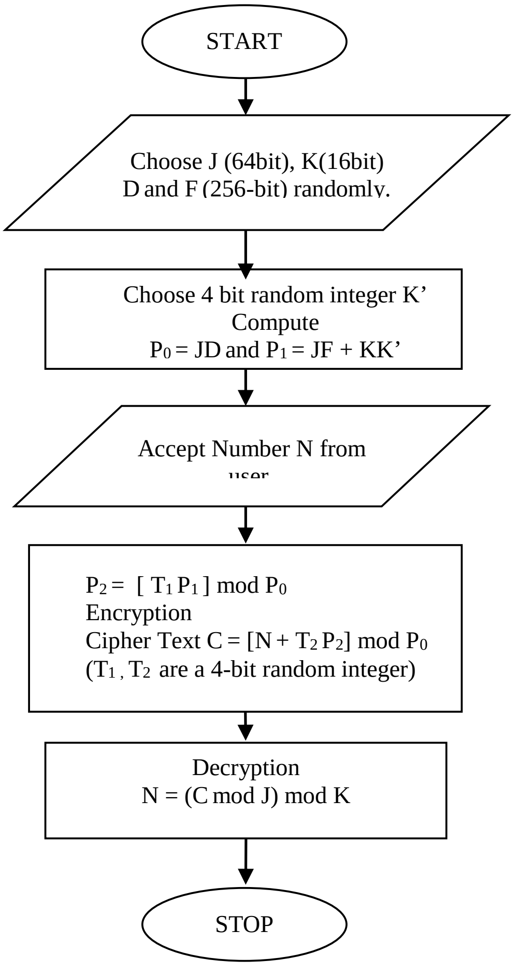

Fig. 3: Flowchart of Fully Homomorphic Encryption Scheme

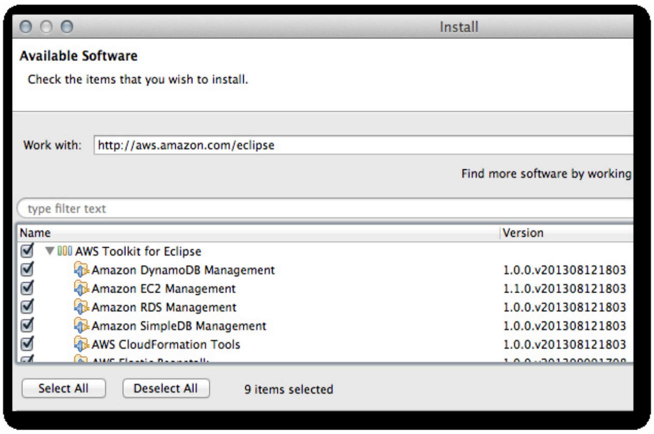

Fig.4 Checklist for installation of Java SDK on Eclipse Kepler Version

4.1 Results

Two tables are created on DynamoDB. Balance is stored using homomorphic encryption scheme. User is allowed to perform addition and subtraction on this encrypted balance. User is allowed to check balance in plaintext. The sample data on DynamoDB database is shown in fig. 5

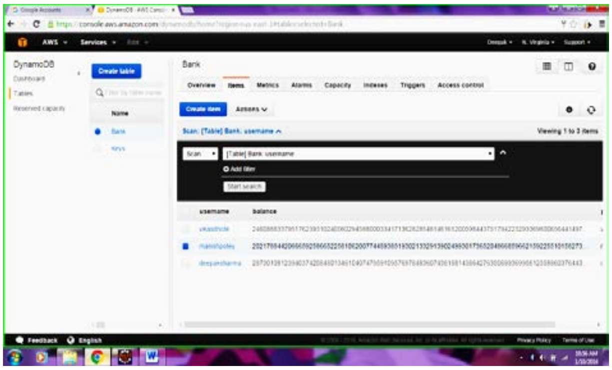

Fig. 5: Database on DynamoDB

The code is executed at client and performed sample operations. It is shown in Fig. 6.

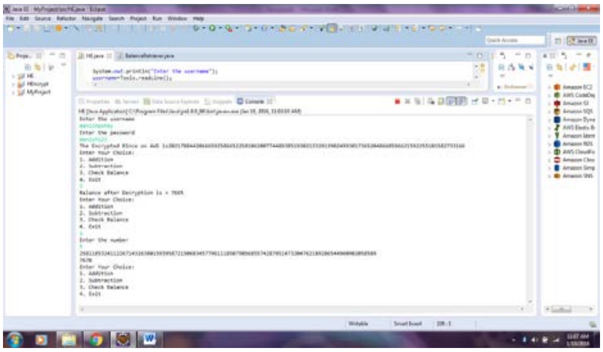

Fig. 6 Executions at Client

After execution of this code at client side the data is updated on AWS DynamoDB table. It is shown in fig. 7.

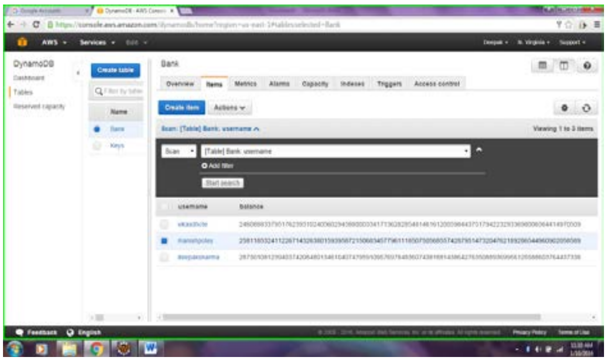

Fig. 7: Data in DynamoDB table after performing addition in balance

4.2 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 as in no stage data is exposed in plain 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. 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 nonspecialists." 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 ciphertext 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. Stehlé, Damien, and Ron Steinfeld. "Faster fully homomorphic encryption." Advances in Cryptology-ASIACRYPT 2010. Springer Berlin Heidelberg, 2010. 377-394.

21.Amazon Web Services DynamoDB, Available: Amazon Web Services Online, http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

22.Amazon Web Services DynamoDB, Available: Amazon Web Services Online, https://github.com/amazonwebservices

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

24. Gentry, Craig. A fully homomorphic encryption scheme. Diss. Stanford University, 2009.