I J C T A, 9(17) 2016, pp. 8709-8716
© International Science Press

Two-tier Architecture to Improve Performance of Fully Homomorphic Encryption Security Mechanism

Manisha Rani[*] and Gagandeep[**]

ABSTRACT

Although Cloud Computing has performed lots of advancements in improving technology used in various fields, but still there exists many security threats to data and cloud resources which reduces the performance of cloud computing environment. Fully Homomorphic Encryption empowers one of the advanced solutions against cloud security threats. The basic approach of this scheme is to allow arbitrary mathematical computations over encrypted data without any decryption procedure before accessing data. Therefore, it achieves a prominent value in cloud computing by providing data confidentiality. This research paper focuses on the design of secure architecture which aggregates two different encryption mechanisms. At the bottom level, Fully Homomorphic Encryption technique has been implemented which performs homomorphism operations over ciphertext. At the top level, Identity Based Encryption technique has been implemented in which a session has been created for limited time period before executing the operations thereby, reducing the time complexity of Fully Homomorphic Encryption technique. The proposed architecture has been implemented using Hadoop MapReducer Java platform embedded with Eclipse IDE and the results obtained are better as compared to existing mechanisms of cloud security.

Keywords: Cloud Computing, Fully Homomorphic Encryption, levelled Fully Homomorphic Encryption, Additive Homomorphism, Multiplicative Homomorphism, Identity based Encryption

I. INTRODUCTION

Cloud Computing is a hybrid approach of distributed, grid computing and parallel computing which is composed of several hundreds and thousands of computing nodes deployed over cloud network. The nodes contain various computing resources like storage, software, infrastructure, applications etc. that can be accessed dynamically wherever and whenever required. The large amount of data can be stored and processed any time by integrating multiple resources over cloud network [15]. The clients can outsource their data over public network of cloud which can be shared among multiple cloud users in an efficient way. Therefore, Cloud Computing is being increasingly used in several applications like IT sectors, business, medical, military, finance etc. Ubiquitous access to data without knowing its physical location, data management, easy sharing among multiple users and so on are some advantages of outsourcing data to the cloud but it gives birth to various new security threats. Hence, it faces various data security issues such as data confidentiality, integrity and privacy among the important one. Different security attacks like replay attack, denial of service, fake data injection etc. are also associated with cloud [10]. Thus, cryptography seems to be the best possible solution to protect the information from malicious attackers and untrusted third parties. This technique involves enciphering of data in scrambled form which can be retained through decipherment operation.

Although cryptography is one of the promising solutions for data security, yet cloud users do not have direct control over their outsourced data which ceases users from trusting the service providers and hence,

[*] M.Tech Scholar, Department of Computer Science, Punjabi University, Patiala, Punjab, India, *E-mail: manishabhandari1993@gmail.com*

[**] Associate Professor, Department of Computer Science, Punjabi University, Patiala, Punjab, India, *E-mail: gdeep.pbi@gmail.com*

prevents them from acquiring the cloud environment. Thus, the data owners must have control over their data by encrypting the sensitive information over their sites themselves. Fully Homomorphic Encryption is a client side technique in which cloud users can perform encryption operations over their side themselves without any interference of cloud servers or service providers.

The rest of the paper is organized as follows: In Section II, basic concepts and ideas of Fully Homomorphic Encryption are overviewed. Section III comprises literature survey on related techniques of Fully Homomorphic Encryption technique. Section IV and V provides details of the proposed architecture along with implementation and results. Finally, the research paper has been concluded in Section VI along with future perspective of the work.

II. FULLY HOMOMORPHIC ENCRYPTION

Homomorphic Encryption [16] is a cryptographic mechanism which permits direct mathematical computations over ciphertext data itself without knowing actual data and gives ciphertext itself as an output. It performs few arithmetic operations over cipher text, either additive or multiplicative homomorphism operations. These further evolved to Partial Homomorphic Encryption scheme. RSA, Pallier and Elgamal techniques are some instances of Partial Homomorphic Encryption scheme. Additive Homomorphism performs addition operations over two or more encrypted data whereas Multiplicative Homomorphism technique performs multiplication operation over two or more cipher data. These homomorphic operations can be verified by decrypting the output of the operations which must match the results of corresponding operations performed on plaintext data.

Mathematically, Additive and Multiplicative Homomorphism operations can be shown as follows:

$$\text{Encr}(x \# y) = \text{Encr}(x) \# \text{Encr}(y)$$

$$\text{Decr}(x \# y) = \text{Decr}(\text{Encr}(x) \# \text{Encr}(y))$$

Where Encr and Decr are encryption and decryption operation and # can be either additive or multiplicative homomorphism operation.

After conducting long experiments and analysis, Craig Gentry discovered Fully Homomorphic Encryption in 2009 after performing lots of modifications to simple homomorphic encryption using the concept of ideal lattices. Fully Homomorphic Encryption is basically based on construction of mathematical computational complexity programs or problems in which arbitrary number of homomorphism operations can be performed over ciphertext data itself results in encrypted outputs itself. In this scheme, data owner has full control over the data stored over the cloud network. It stores encrypted data over cloud servers and conceals the actual data from unauthorized users, cloud servers or trusted third parties. Hence, it is considered to be a client-side encryption mechanism. It enhances the security of data through protection of data privacy over public networks and hence, the trust of data owner towards the cloud environment. Fully Homomorphic Encryption scheme plays a big role in providing data confidentiality at Software-as-a-Service level.

Fully Homomorphic Encryption can be implemented in four phases named as Key Generation Phase, Encryption Phase, Decryption Phase and Evaluation Phase. First phase deals with the generation of either public key or private key for asymmetric encryption or symmetric encryption [12] respectively. The plaintext data can be encrypted in encryption phase using keys generated in prior phase. The main heart of this technique is evaluation phase in which different homomorphism operations can be evaluated over encrypted data. The outputs of operations can be decrypted using decryption phase. The basic steps of Fully Homomorphic Encryption Algorithm are as follows:

Key Generation: Outputs either pair of private and public key (PR, PU) or only secret key SK

Encryption: Encrypt plaintext message M using either public key or secret key

i.e., C=Enc (M, PU/SK)

Evaluation: Perform any number of additive and multiplicative operations over two or more encrypted data.

i.e., C_{add}=Add (C1, C2)

C_{mul}=Mul (C1, C2)

Decryption: Verification of homomorphism operations using private key generated in first phase.

M1+M2=Dec (C_{add}, PR)

M1*M2=Dec (C_{mul}, PR)

III. LITERATURE SURVEY

Many researchers have worked in the area of cryptography to enhance the security of outsourced data of clients. Different algorithms like RSA, Elgamal, Pallier and homomorphic encryption schemes have been used in security domain. In 2009, Gentry [1] brought Fully Homomorphic Encryption into light by doing various modifications in already existing homomorphic encryption schemes. It has been constructed using the concept of ideal lattices and permits arbitrary number of computations over ciphertext data itself. But high mathematical computations, large storage requirements and very high processing reduced the performance of scheme. After analyzing various pitfalls of Gentry's scheme, VanDijk [2][3] proposed Somewhat Homomorphic Encryption over the integers using the concept of modular arithmetic and reduced the complexity of technique to the order of 10. Further, Smart [4] performed refinements in Gentry's approach by obtaining better depth of decryption circuit using reduced ciphertext size and public key size. Brakerski [5] proposed the concept of standard Learning with Error (LWE) applicable over Fully Homomorphic Encryption thereby, to enhance the performance of scheme in terms of reduced ciphertext size. Brakerski [6] continued their work by implementing polynomial circuits over Fully Homomorphic Encryption based on Ring Learning With Error (RLWE) concept without utilizing Bootstrapping operation. Chen [7] discussed the principles and applications of Fully Homomorphic Encryption in order to prevent leakage of user's data. This paper constructed the cloud computing platform in different modules named as function module and control module etc. Kocabas [8] [9] analysed the feasibility assessment of cardiac health monitoring for long-term patients utilizing the Fully Homomorphic Encryption library named as HElib.

IV. OUR CONTRIBUTION

We propose two layer secure architecture has been implemented using Hadoop MapReducer Java platform embedded with Eclipse IDE. It is an aggregation of two different security mechanism: Identity-Based Encryption has been employed after implementation of Levelled Fully Homomorphic Encryption scheme. Fully Homomorphic Encryption based on DGHV scheme has been implemented at the bottom level of architecture coupled with identity-Based Encryption at the next level of security architecture.

(A) Levelled Fully Homomorphic Encryption

Basically, Levelled Fully Homomorphic Encryption computes arbitrary number of additive operations but limited number of multiplicative operations having depth of at most l so as to reduce the noise component added by multiplicative operation in encrypted data. The levelled Fully Homomorphic Encryption can be implemented using DGHV scheme in which both encryption and decryption is performed using symmetric key only [14]. The Algorithm for same is as follows:

Level_FHE (Message m, Add A, Multiply Mu)

1. Generate Private key p using any positive odd number. The encryption parameters q and r are chosen where large positive integer is chosen as q and r is selected randomly and encrypted using PK.

2. Plaintext Message m can be encrypted as follows:

$$C = m + p*q +2\ r$$

3. Homomorphism operations over two ciphertexts can be performed arbitrarily over evaluation circuit CK as below:

   Additive: $A= C_1+C_2$

   Multiplicative: $Mu=C1+C_2$

   Where $C_1 =m_1+p*q_1+2r_1$

   $C_2=m_2+p*q_2+2r_2$

4. The above operations can be decrypted to get original value using following mathematical equation:

   $D=(C \bmod p) \bmod 2$

(B) Identity Based Encryption

Identity-Based Encryption (IBE) [21] is a cryptographic technique in which either public, private key pair or only secret key is generated for each user based on their identity. If any user wants to access cloud data, he/she must have to verify his/her identity before it. If his/her identity is authenticated, he/she will have permission to access otherwise permission will be denied which protects the sensitive information from malicious users. Thus, the encrypted data stored over cloud can be accessed by authenticated users only. Digital Signatures, session based encryption and application based encryption etc. are kinds of Identity-Based Encryption. In this Paper, a kind of Identity based encryption i.e., Session Based Encryption is implemented over top of Levelled Fully Homomorphic Encryption.

Thus, Identity Based Encryption is implemented by creating a small session of limited time period whenever an authenticated user login. User must have to perform encryption and evaluation process within the session. It also creates a new private key (or session key) for each login and destroys after session expires. It generates different private key based on integer random number each time when user logins to session Therefore, it gives different ciphertext data of same message for each session. It enhances data confidentiality of Fully Homomorphic Encryption to the most. It reduces the execution time of performing homomorphism operations over encrypted data based on size of input message. That is, large input size message will be executed slower than small size messages. It also facilitates user authentication along with data confidentiality to Fully Homomorphic Encryption scheme at SaaS level of cloud model. Both Fully Homomorphic Encryption and Identity Based Encryption can be implemented step by step using following algorithm:

```
LEVEL_IBE (username, password, Add, Mul)
    1. Login Username and password
    2. Check authenticity of user:
       If username and password are valid, then go to step 3
       Else generate error “access denied” and go to step 10.
       End If
    3. Begin user Session for some time (say, 1 min)
    4. Generate Private/Session Key (PR) using random numbers.
    5. Perform Homomorphism operations using private key by selecting flag value either 0 or 1 as follows:
    6. If enter 0, Perform Additive Homomorphism:
       Write: $C_{add}$ (C1+C2, PR)
    End If
7.  If enter 1, check depth of multiplicative operation:
    If depth<d, then Perform multiplicative homomorphism:
    Write: $C_{mul}$ (C1*C2, PR)
    Else go to step 7
    End If
    End If
8.  Update server database over cloud.
9.  Logout server page.
10. User session expires, go to step 1.
11. Exit.
```

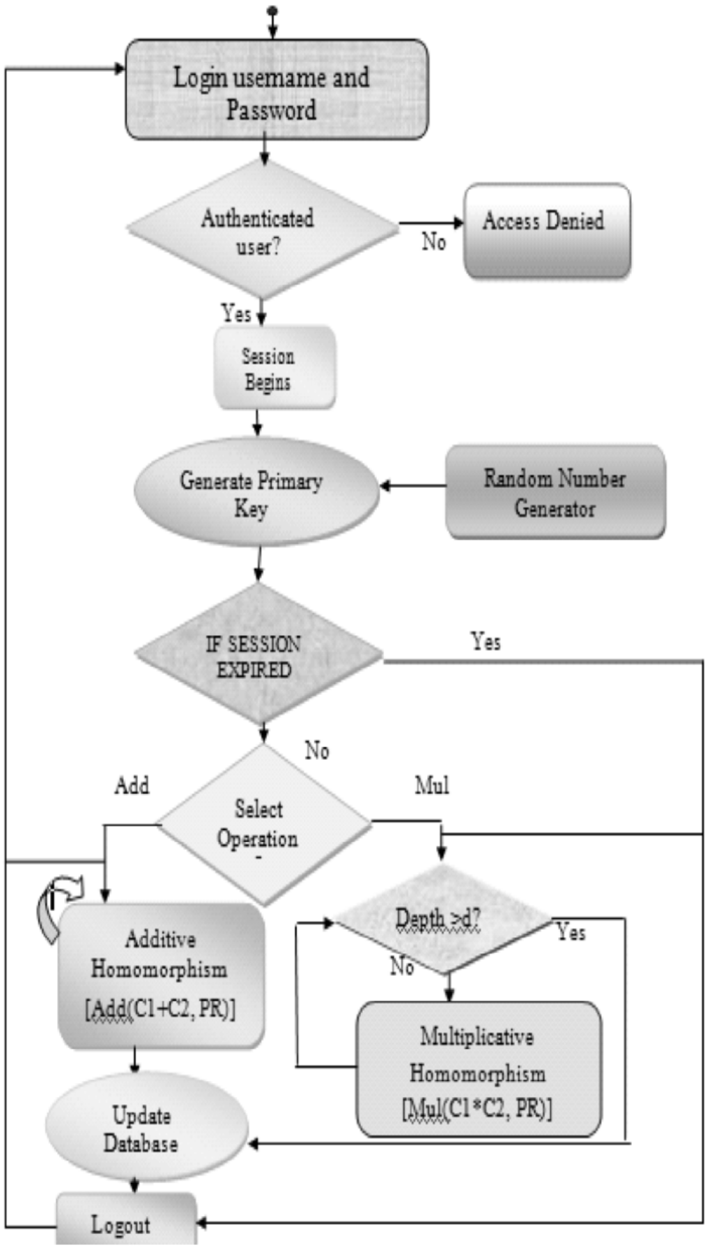

Figure 1: Working of Two-tier architecture

V. IMPLEMENTATION AND RESULTS

Levelled Fully Homomorphic Encryption based on Identity Based Encryption has been implemented using Hadoop MapReducer software embedded with Eclipse IDE. The execution time of the proposed architecture has been compared with DGHV scheme for same input and the results obtained are shown in Table 1.

Table 1
Comparison of Execution Time

Encryption	*Execution Time (in ms)*
DGHV scheme	21.93
Proposed Scheme	18.92

The execution time and level of security have been compared based on applied input.

Execution Time: The execution time of an algorithm is compared with respect to size of input data. According to our scheme, large input sized data is executed fast due to reducing nature of Hadoop for Big Data. Thus, the execution time of system decreases as the size of input data increases.

Level of Security: Whereas Level of Security defines the strength of security which can be measured as maximum time that can be taken to break the encryption process. It mainly depends on static and dynamic nature of private key. It is evaluated using product of input size , its execution time and either static or dynamic key value.

LOS=N*T*P_i iÎ{0,1}

where N is size of input

T is corresponding execution time

P_0 is size of static private key

P_1 is size of dynamic private key

As per the Proposed architecture, level of security increases as size of input data increases. It has been observed that in case of dynamic private keys, value of private key also changes every time when encryption is performed. Thus, level of security increases as compared to static private keys. Table 2 represents level of security in terms of size and execution time for both static and dynamic keys.

Table 2
Level of Security

Size of Input (N in bytes)	*Execution Time (T in ms)*	*Level of Security (N*T*P)*	
		Static (P=7)	*Dynamic(P varies)*
26	32.85	5978.7	9395.1
41	42.01	12056.87	13786.64
56	35.17	13789.64	17725.68
73	38.95	19903.45	36963.55
174	34.01	41424.18	47341.92

VI. CONCLUSION

Cloud Security is a big issue in the developmental progress of cloud computing where Fully Homomorphic Encryption provides a promising way of solving the problems of cloud security. Yet, high computational complexity of Fully Homomorphic Encryption in solving large number of multiplicative operations still remains a challenging task. This paper explores two-tier architecture to improve the security of SaaS level

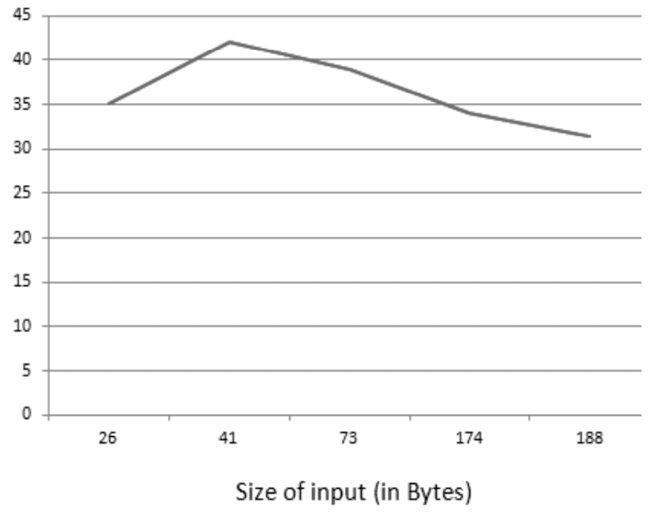

Figure 2: Execution Time vs. Input Size

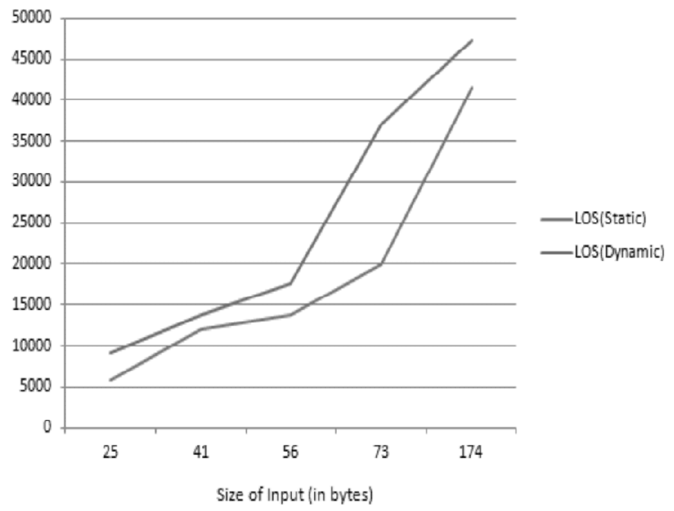

Figure 3: Level of Security (Static vs Dynamic)

using Fully Homomorphic encryption mechanism along with identity-based encryption technique in order to provide user authentication along with data confidentiality. Moreover, generation of dynamic private key boosts the security in cloud. Still, more options need to be exploited in the area of cloud security.

REFERENCES

[1] C. Gentry, "Fully Homomorphic Encryption using Ideal Lattices", Proc. 41st Annual Association for Computing Machinery Symposium on Theory of Computing, New York, USA, pp. 169-178, 2009.

[2] V.M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers", 29th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, *Springer Berlin Heidelberg Publishers*, French Riviera, pp. 24-43, 2010.

[3] J.S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys", Advances in Cryptology- *Lecture Notes in Computer Science*, pp. 487-504, 2011.

[4] N.P. Smart, and F. Vercauteren, "Fully Homomorphic Encryption with Relatively small key and ciphertext sizes", 13th Int. Conf. on Practice and Theory in Public Key Cryptography, France, pp. 420-443, 2010.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully Homomorphic Encryption without Bootstrapping", Proc. 3rd Innovations in Theoretical Computer Science Conference, New York, USA, pp. 309-325, 2012.

[6] Z. Brakerski, and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE", Proc. IEEE 52nd Annual Symposium on Foundations of Computer Science, Washington, USA, pp. 97-106, 2011.

[7] B. Chen, and Na. Zhao, "Fully Homomorphic Encryption Application in Cloud Computing", 11th Int. Comp. Conf. on Wavelet Active Media Technology and Information Processing, Chengdu, pp. 471-474, 2014.

[8] O. Kocabas, and T. Soyata, "Utilizing Homomorphic Encryption to Implement Secure and Private Medical Cloud Computing", Proc. IEEE 8th Int. Conf. on Cloud Computing, New York, pp. 540-547, 2015.

[9] J.P. Couderc, O. Kocabas, T. Soyata, M. Aktas, J. Xia, and M. Huang, "Assessment of Cloud-based Health Monitoring using Homomorphic Encryption", Proc. IEEE 31st Int. Conf. on Computer Design, Asheville, North Carolina, pp. 443-446, 2013.

[10] A. Blilat, A. Bouayad, M.N. El Houda, and M. El Ghazi, "Cloud Computing: Security Challenges", Int. Colloquium on Information Science and Technology, Fez, Morocco, pp. 26-31, 2014.

[11] C. Gentry, and S. Halevi, "Implementing Gentry's Fully Homomorphic Encryption Scheme", Proc. 30th Annual International Conference on Theory and applications of cryptographic techniques, Berlin, pp. 129-148, 2011.

[12] C.P. Gupta, and I. Sharma, "A Fully Homomorphic Encryption Scheme with Symmetric Keys with Application to Private Data Processing in Clouds", 4th International Conference on the Networks of Future, Pohang, pp. 1-4, 2013.

[13] N. Aggarwal, C.P. Gupta, and I. Sharma, "Fully Homomorphic Symmetric Scheme without Bootstrapping", Int. Conf. on Cloud Computing and Internet of Things, China, pp. 14-17, 2014.

[14] F.C. Liu, and F. Zhao, C. Li, "A Cloud Computing Security Solution based on Fully Homomorphic Encryption", 16th Int. Conf. on Advanced Communication Technology, pp. 485-488, 2014.

[15] I. Foster, and K. Chard, J.S. Tuecke, "Efficient and Secure Transfer, Synchronization and Sharing of Big Data", IEEE Journal on Cloud Computing, pp. 46-55, 2014.

[16] M.F.B. Zolkipli, and A.N. Jaber, "Use of Cryptography in Cloud Computing", Proc. IEEE Int. Conf. on Control System, Computing and Engineering, pp. 179-184, 2013.

[17] M. Yang, J.L. Han, C.L. Wang, and S.S. Xu, "The Implementation and Application of Fully Homomorphic Encryption Scheme", 2nd Int. Conf. on Instrumentation and Measurement, Computer, Communication and Control, pp. 714-717, 2012.

[18] S. Dara, "Cryptography Challenges for Computational Privacy in Public Clouds", IEEE Int. Conf. on Cloud Computing in Emerging Markets, Bangalore, pp. 1-5, 2013.

[19] S. Picek, and D. Hrestak, "Homomorphic Encryption in the Cloud", 37th Int. Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, pp. 1400-1404, 2014.

[20] S. Vagdevi, and R. Kangavalli, "A Mixed Homomorphic Encryption Scheme for Secure Data Storage in Cloud", Int. Advance Computing Conf., Bangalore, pp. 1062-1066, 2015.

[21] V.S. Agme, and A.C. Lomte, "Cloud Data Storage Security Enhancement using Identity Based Encryption", Int. Journal of Application or Innovation in Engineering & Management, pp. 340-345, April 2014.