RESEARCH ARTICLE OPEN ACCESS

Optimizing Fully Homomorphic Encryption Algorithm using Greedy Approach in Cloud Computing

Kirandeep Kaur [1], Jyotsna Sengupta [2]
Department of Computer Science
Punjabi University, and Patiala
Punjab - India

ABSTRACT
With the rapid growth of cloud computing, many users store their data and application on the cloud. However, the growth of cloud computing is slowed down by cloud security problem. Hence, cloud computing security becomes the current research focus. To secure the cloud data, encryption is used. Traditional encryption schemes cannot make cloud computing fully safe. Therefore, Fully homomorphic encryption is used to secure the data from exploitation during computation. In this paper, Fully Homomorphic Encryption (FHE) encryption scheme is optimized in which the time complexity and space used by encrypted data is reduced using Greedy approach. Greedy approach is applied to the input data that is tokenized. On this data, greedy approach will find out the max cost data that is to be encrypted first, but with a condition that it may utilize the full capacity of resources. The proposed approach that is Greedy based FHE provides better results as compared to the existing FHE encryption algorithm.

Keywords :— Cloud Computing, Cloud Security, Fully Homomorphic Encryption.

I. INTRODUCTION

Cloud computing is a delivery model of computing resources over the internet. It enables real time development, deployment and delivery of wide range of services and products where different services are delivered to an organization's computers and devices through the internet. In this computing, users are charged based on pay-per-use.

Even though cloud computing has become widely popular as a service model; adoption of its services is limited due to concerns about security of data. The solution to this problem is encryption. However, if the clients wish to perform computations on their encrypted information in the cloud, they require to share the secret key with cloud provider in order to decrypt it before performing the desired operations. Thus, the best solution is to use homomorphic encryption to deal with cloud computing security issues. It is because this method allows the client to perform operations on encrypted data without having to disclose the secret key required to decrypt the data.

In 2009, Craig Gentry [1] proposed the first fully homomorphic encryption scheme. A scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. In 2010, Smart and Vercauteren [2] presents a fully Homomorphic encryption scheme following the Gentry scheme with small key size and cipher text size. It defines a new algorithm named Rcrypt, which takes a dirty cipher text and convert it into new cipher text removing some errors. This scheme has smaller message expansion and smaller key size than Gentry encryption scheme. Dijk et al. [3] describes fully homomorphic encryption obtained from somewhat homomorphic encryption scheme with using Gentry's techniques to convert it into a fully homomorphic. In this scheme, somewhat encryption uses multiplication and additions over the integers. Zhao et al. [4] proposed an algorithm for overcoming the lack of security. In this paper, fully homomorphism encryption algorithm is described to ensure that the data stored on the cloud is secure. The proposed solution provides protection and is suitable for performing operations on the stored data effectively. Gupta and Sharma [5] proposed a fully homomorphic encryption scheme using symmetric keys. The operations are matrix based, that is mapping the operation on integer to matrix for processing the private data. Ahmed and Khandekar [6] proposed the application of a method to perform the operation without decrypting and provide the same result as computation performed on plain data. In this scheme, proxy re-encryption algorithm based on RSA and Paillier is used to prevent the cipher data from Chosen Cipher text Attack. This algorithm generates a random key cipher data. This system provides more security. Dongxi Liu [7] proposed a symmetric FHE that does not need any noise reduction method. This scheme allows large amount of noise in ciphertext and resulting cipher texts get decrypted correctly regardless of the noise in them. Yao et al. [8] proposed a protocol based on homomorphic encryption that enables the input of function can encrypted with different public keys. In new SMC protocol, two servers used for all the computations except for initial encryption and final decryption. Sha and Zhu [9] described an addition algorithm for modifying the existing RSA algorithm to obtain the characteristics of additions with characteristics of multiplications. It combines the Pascal's triangle theorem and RSA algorithm model to build a new cryptosystem that meets homomorphic computation of some operations on cipher texts. Jabbar and Najim [10] discussed the use of homomorphic encryption to encrypt the client's

data before storing in cloud storage. Homomorphic Encryption allows computation over encrypted data without decryption. Marwan et al. [11] proposed a technique based on homomorphic scheme to secure cloud database. Homomorphic cryptosystem proposed based on RSA and Palliler algorithms. This approach also guarantees data confidentiality and allows users to perform arithmetic operation over encrypted data. Potey et al. [12] focused on storing data on the cloud in the encrypted format using fully homomorphic encryption User's computations are performed on encrypted data in public cloud. In this scenario, user's data is never stored in plaintext on public cloud. This scheme is used for different medical and business purposes.

This paper addresses the security of user data in cloud. The main aim of this paper is to introduce the concepts of Homomorphic encryption and how to optimize the fully homomorphic encryption discussed by Potey et al. [12] in their paper.

The rest of the paper is organized as follow: Section II discuss about cloud computing concepts and its services and deployment models. Section III describes about cloud security issues. Section IV discuss encryption in cloud and provide details about how homomorphic encryption secure the user data in cloud. Section V describe the design of optimized fully homomorphic encryption algorithm. Results are shown in Section VI. Finally our conclusions are drawn in section VII.

II. CLOUD COMPUTING

Cloud computing defines the combination of logical entities like data, software which are accessible over internet. Client data is stored in the banks of servers spread across the globe. If an organizations uses a cloud computing, it does not need to spend money to buy hardware or software licenses. Therefore, it drastically reduce the cost and management of owning and operating computers and networks [13].

A. Cloud Computing Models

1) Services Models: A service model determines what kind of computer resources and services are offered to consumers [13].

- *Software as a Service* refers to software that is accessed via a web browser.
- *Platform as a service* provides development environment as a service.
- *Infrastructure as a service* offers the computing resources like storage and processing as a service.

2) Deployment Models: Cloud services can be deployed in following four ways [13].

- *Public cloud* is available to all public users who can subscribe the required services.
- *Private cloud* owned and managed by the organization or the designated service provider.
- *Hybrid cloud* combines two or more clouds. This model provides flexibility to organizations.
- *Community cloud* is shared among several organizations that have common requirements or concerns and works together to complete their objectives.

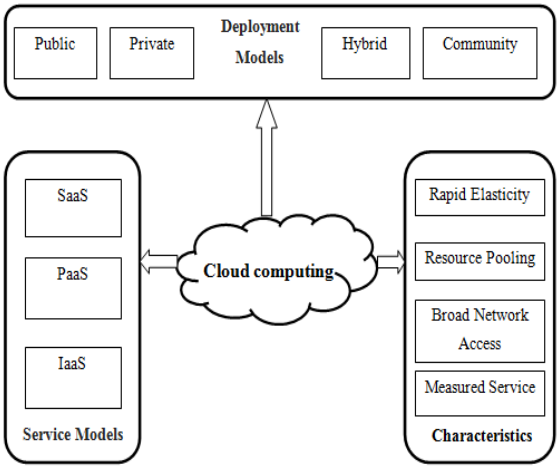

Fig. 1 Cloud Computing Overview

B. Characteristics of Cloud Computing

- *Rapid Elasticity:* Additional resources can be easily provisioned or released as per the demand.
- *Resource Polling:* Cloud resources like storage, processing and virtual machines can be shared by the users to serve multiple consumers using a multi-tenant model [14].
- *Broad Network Access:* Cloud Services are provided over the network with secure protocol so that it can be accessed from various client machines [14].
- *Measured Device:* The cloud system uses a metering capability to maintain a transparent record of resource usage.

III. CLOUD SECURITY

Cloud Security is an evolving sub-domain of computer security, network security and information security [15]. Data location is critical factor in cloud security. The users have no control and no knowledge about what could happen to their data. This is a great concern in cases when users store precious and their personal information in a cloud. Some of security aspect with data in cloud are described below:

Data Confidentiality ensures that data contents are not made available or disclosed to illegal users.

Data Access Controllability means that a data owner can perform the selective restriction of access to his data stored in cloud [15].

Data Availability gets affected by the denial of attack and network deficiency. To overcome this issue, Fault tolerance and load balancing techniques are used [15].

Data Integrity demands maintaining and assuring the accuracy and completeness of data.

IV. ENCRYPTION IN CLOUD

Cloud encryption is the transformation of a cloud service customer's data into cipher text. Cloud service provider offers cloud encryption schemes to encrypt data before storing on cloud. The cloud encryption capabilities of the service provider need to match the level of sensitivity of the data being hosted.

A. Homomorphic Encryption

Homomorphic Encryption scheme allow computation with encrypted data. One can perform additions or multiplication over two encrypted numbers and the other could decrypt the results without of being able to find the values of individual numbers. Homomorphic Encryption originated from the concept of privacy homomorphism introduced by the Rivet et al [16]. In their paper, they discussed about performing operations on encrypted data. In 2009, Craig Gentry introduced the first Fully Homomorphic Encryption. In 2010, M. Dijk, C Gentry et al discussed the second fully homomorphic encryption scheme.

1) *Homomorphic Encryption types:* There are two types of homomorphic encryption schemes.

- *Partially homomorphic encryption:* Partially homomorphic schemes only support one type of homomorphic operation. RSA and El Gamal provides support to only multiplication operation. Pascal Pallilier introduced the Pallilier cryptosystem, which support the additive homomorphism [16].
- *Fully homomorphic encryption:* Fully homomorphic support both the operations i.e additions and multiplications. Craig Gentry firstly proposed this type of encryption.

V. DESIGN OF GFHE ALGORITHM

Optimized algorithm named as greedy based fully homomorphic encryption (GFHE). In this, greedy approach combines with fully homomorphic encryption. Using the greedy approach, the blocks of text get sorted by the greedy method so utilize the full memory utilization. Using this property of the greedy approach the time and space utilization reduced. (J, K) forms a private key and (P0, P1) presents public key.

To convert a string into integer following equation is to be used

S [0]*31^ (n-1) + S [1]*31^ (n-1) +....+ S [n-1].....(1)

A. Flow Chart of GFHE Algorithm

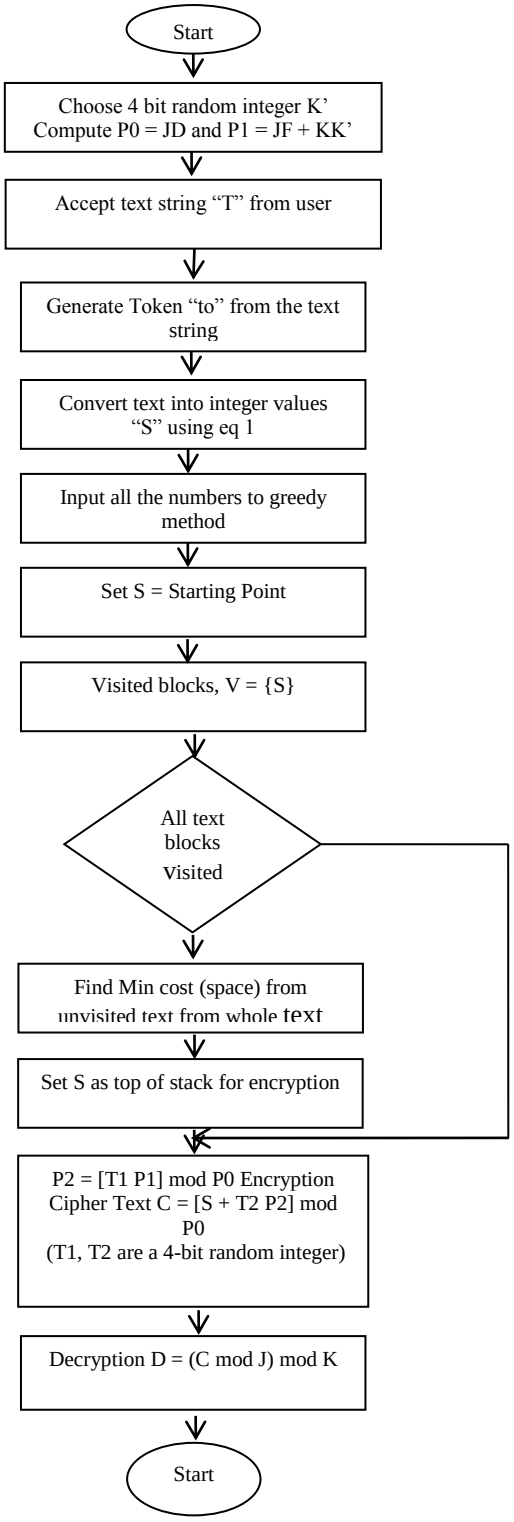

Fig. 2 Flow Chart

VI. RESULTS

The result of an algorithm is totally based on these parameters. However, the comparison between the algorithms can be convenient by using different parameters. This thesis uses parameters which are reflecting the performances and are as follows:

Encryption Time: The time used to generate a cipher text from a plaintext, is known as encryption time.

TABLE I

ENCRYPTION TIME

FILE SIZE	ENCRYPTION (ns)	
	FHE	GFHE
10 KB	92503905	69299243
20 KB	185845371	140502077
30 KB	370320532	279702323
40 KB	458119567	359806729
50 KB	654295471	475599707

Comparison between these algorithms is shown in the form of following graphs.

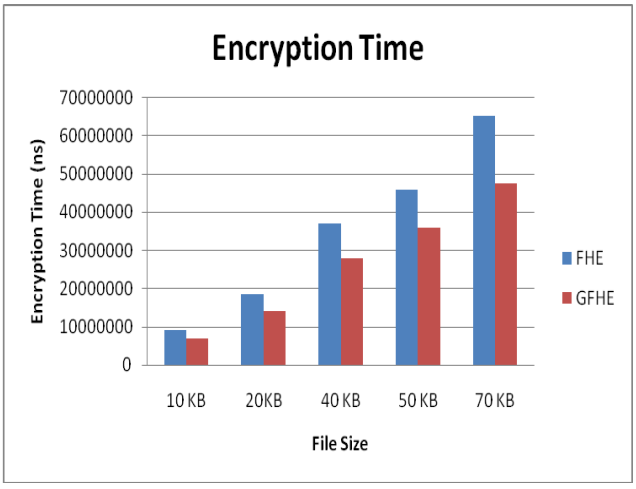

Fig. 3 Encryption time / File size

From the fig. 3 it is clear that the Greedy based Homomorphic Encryption algorithm perform encryption operations in less time as compared to Homomorphic Encryption because greedy algorithm always tends to do more operations in a particular defined resource set.

TABLE II

DECRYPTION TIME

FILE SIZE	DECRYPTION (ns)	
	FHE	GFHE
10 KB	99851365	43103282
20 KB	187306817	86802770
30 KB	351155769	173007138
40 KB	459532653	217506729
50 KB	603764791	316599707

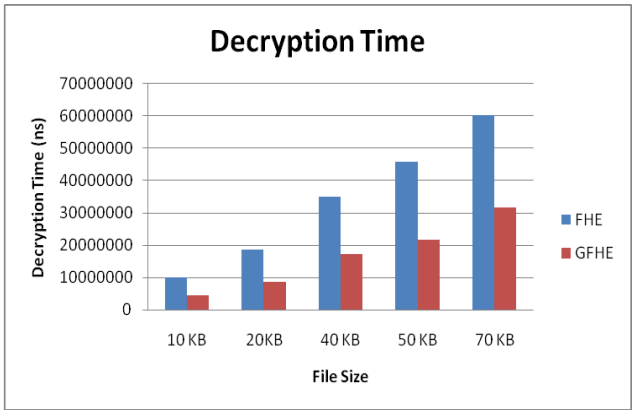

Fig. 4 Decryption time / File size

Fig. 4 shows that the Greedy based Homomorphic Encryption algorithm takes less time to decrypt the cipher data as compared to fully homomorphic encryption algorithm.

Space Utilization: Space utilization of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.

TABLE III

SPACE UTILIZATION

FILE SIZE	Space Utilization (bits)	
	FHE	GFHE
10 KB	91296	76080
20 KB	189292.6	157744
30 KB	388723.2	323936
40 KB	485155.9	404296
50 KB	678700.8	565584

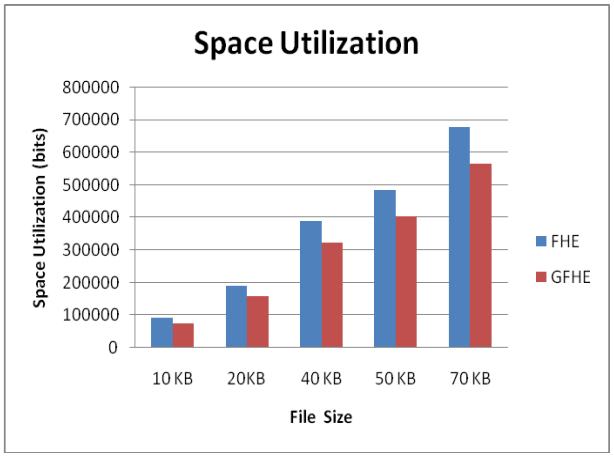

Fig. 5 Space Utilization / File Size

Fig. 5 depicts that the Greedy based Homomorphic Encryption perform operations takes less space as compared to Homomorphic Encryption because greedy algorithm always tends to do more operations in a particular defined resource set.

Encryption Overheads: Encryption overheads defined as the number of extra bits required to perform some kind of operations.

Encryption overhead (bits) = encrypted bits - plaintext bits

TABLE IV

COMPUTATIONAL OVERHEADS

FILE SIZE	Computation overheads (bits)	
	FHE	GFHE
10 KB	81786	66570
20 KB	169574	138026
30 KB	348231.8	283444
40 KB	434618	353759
50 KB	608002.8	494886

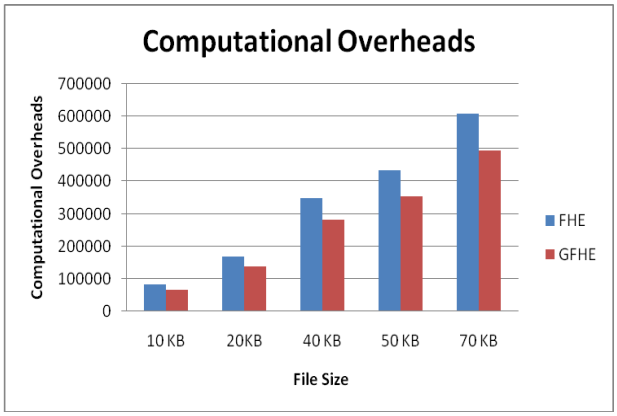

Fig. 6 Computational Overheads / File Size

From the fig 6 it is clear that the computational overheads are more in case of Homomorphic encryption because this encryption has not any support of optimization so it will not tend to fully utilize the resource vector that is given.

VII. CONCLUSION

In the proposed work, fully homomorphic encryption algorithm is improved with greedy approach. Using the greedy approach, the blocks of text are sorted by the greedy method to utilize the full memory utilization. Using this property of the greedy approach the time and space utilization is reduced also. The comparative study of the Fully Homomorphic Encryption and Greedy based Fully Homomorphic Encryption is done on the basis Encryption time, Decryption time, space utilization, encryption overheads and throughput. From the result part, it shows that that proposed approach is far better than that of existing Homomorphic encryption.

In future, this algorithm can be improved by reducing the space complexity with time. This algorithm can be used for encryption of multimedia data.

ACKNOWLEDGEMENT

The fruition of any research relies on collaboration, coordination and consolidated endeavours of information. I am thankful to Dr. Jyotsna Sengupta (Associate Professor, Punjabi University, Patiala).

REFERENCES

[1] C. Gentry, "Fully Homomorphic encryption using Ideal Lattices," *InProc of STOC*, pp. 169-178, 2009.

[2] N.P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," *public Key Cryptography-PKC Springer Berlin Heidelberg*, vol. 6056, pp. 420-443, 2010.

[3] M.V. Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, "Fully homomorphic over the integers", *In Proc. of Eurocrypt*, vol. 6110, pp. 24-43, Jan 2010.

[4] F. Zhao, C. Li and C.F. Liu, "A cloud computing security solution based on fully homomorphic encryption," pp. 485-488, 2014.

[5] C. P. Gupta and I. Sharma, "A Fully Homomorphic Encryption scheme with Symmetric Keys with Application to Private Data Processing in Clouds," *International Journal of Communication Networks and Distributed Systems*, vol. 14, pp. 379-39, Oct 2009.

[6] I. Ahmad and A. Khandekar,"Homomorphic encryption applied to cloud computing," *International journal of Information and computer and technology*, vol. 4, pp. 1519-1530, 2014.

[7] D. Liu, "Practically Fully Homomorphic Encryption without Noise Reduction," *Technical report, IACR Cryptol. ePrint Arch.*, vol. 12, pp. 305-309, 2015.

[8] Y. Yao, J. Wei, J. Liu and Ru Zhang, "Efficiently secure multiparty computation based on homomorphic encryption," *Cloud Computing and Intelligence Systems, IEEE*, pp. 343-349, 2016.

[9] P. Sha and Z. Zhu, "The modification of RSA algorithm to adapt fully homomorphic encryption algorithm in cloud computing," *Cloud Computing and Intelligence Systems, IEEE*, pp. 388-392, 2016.

[10] I. Jabbar and S. Najim, "Using Fully Homomorphic Encryption to Secure Cloud Computing," *Internet of Things and Cloud Computing*, vol. 4, pp. 13-18, 2016.

[11] M. Marwan, A. Kartit, and H. Ouahmane,"Applying homomorphic encryption for securing cloud database," *Information Science and Technology, IEEE*, pp. 658-664, 2016.

[12] M. M. Potey, C. A. Dhote, and D. H. Sharma, "Homomorphic Encryption for Security of Cloud Data," *7th International Conference on Communication, Computing and Virtualization*, vol. 79, pp. 175-181, 2016.

[13] Y. Jadeja and K. Modi, "Cloud Computing Concepts, Architecture and Characteristics," *International Conference on Computing, Electronics and Electrical Technologies*, pp. 877-880, 2012.

[14] P. Singh and A. Jain, "Survey Paper on Cloud Computing," *International Journal of Innovations 'in Engineering and Technology*, vol. 3, pp.84-89, 2014.

[15] K. Munir and S. Palaniappan, "Security threats/attacks present in cloud environment," *IJCSNS*, vol. 12, no. 12, pp.107-110, 2012.

[16] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Communication of the ACM*, vol.21, pp.120-126, 1978.