Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 7, July 2017, pg.194 – 202

Secure Parallel Processing on Encryption Cloud Data Using Fully Homomorphic Encryption

Prashant.K[1], Ms. Ranjana.N[2]

[1]Department of Studies in computer science and Engineering, VTU Belagavi, India

[2]Assistant Professor Department of Studies in computer science and Engineering, VTU Belagavi, India

[1] pkambar6@gmail.com, [2] ranjana@vtu.ac.in

Abstract- Cloud computing is not our system hard drive; we are using to store the huge amount of data and programs on cloud. Cloud also provides the access to stored data and programs through internet. Cloud computing provide the On-Demand services to user's, client's, organisations and etc. Client's stores the data on cloud in the encrypted form. Homomorphic encryption enables cloud computing to perform the operations on the encrypted cloud data. Fully-homomorphic encryption supports to perform multiple types of operations on encrypted cloud data. But performing the fully-homomorphic encryption (computations on encrypted cloud data) on a single node or in sequential process took the more processing time and memory than the performing the same operations on the plain text (unencrypted data). Parallel processing enables to perform operations on multiple nodes it will take lesser time to complete the applied operation than the sequential process. In this presenting work shows the parallel processing of fully-homomorphic encryption on encrypted cloud data. Here Gentry's encryption algorithm is used to perform FHE on multiple nodes. This parallel processing produces the better performance than the performing operation on sequential process. And also security is a major concern in cloud computing. Here also shows another work on Data Partitioning method is used to improve the security of client data on cloud. Client data will be divided into multiple parts of chunks with equal size and store on different server. And public is generated on client side; this public key is used to store and retrieve data from cloud storage.

1. INTRODUCTION

Cloud computing provides the cloud services to the users, clients, organizations, public and etc., as on the pay-asyou-go method. In cloud computing security_is the major concern. Commonly data encryption techniques are used by clients to secure the data on cloud_computing. Encryption techniques, they effectively secures the client data on public environment called cloud computing. Client can use encryption method on plaintext for security purpose while storing data on cloud, and client can use decryption method to get his own data from the cloud Generally, if client wants to apply some computational operations on his own encrypted data that data is stored on cloud storage. First he should retrieve the data by decrypting the cipher text (i.e., converting cipher text into plain text) from the cloud. After decryption he can apply the computing operations on that data, after applying the operations client can again encrypt the result and store on cloud. This decrypting the data and applying operations, again encrypting the result is an overhead procedure. So this long procedure is reduced by using homomorphic encryption method.

A. Homomorphic Encryption

Homomorphic encryption is a method, which enables to perform computation on encrypted cloud data that is homomorphic encryption method provides an ability to perform operations on the encrypted cloud data. That means perform the operations on the data (cipher text) that is encrypted and stored on cloud without decrypting it (without converting the cipher text into plain text). The result produced by the homomorphic encryption performed operations on encrypted data that is same as the result produced by the performing the same operations on unencrypted (plain text) data.

B. Data Partitioning_And_Encryption Technique

Now a day's cloud storage systems are used for storing the user's large volume of data. User's large data can be stored on cloud_storage and also users can share and download the data on cloud_storage. Data security is a main concern in cloud_storage. There are many techniques exist for the security in the cloud. Sometimes user's data may loss on cloud_storage. In this work presenting the data partitioning method for security purpose for user's data. This data partitioning method gives more security to user's data on cloud. In this user's data is first divided into multiple parts based on size (with equal size of chunks), after partitioning the user's data (text file) store them on the different cloud servers and also generates public key to store and retrieve the users data. This method gives more security to user's data. If attackers get any one part of file, it's impossible to get whole file data by attackers because the file data will divided and stored on different servers. Figure 1 shows the architecture diagram of data partitioning and storing them on different cloud servers

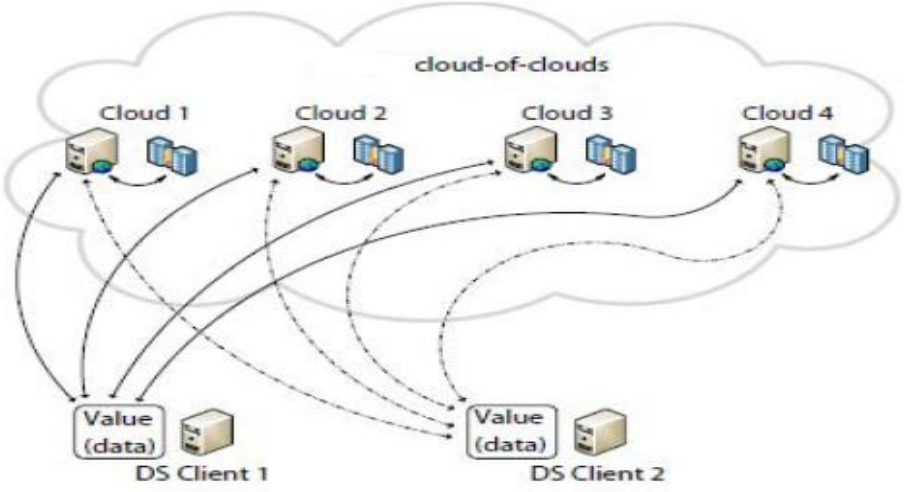


Fig. 1.4 Data Partition Architecture Diagram

2. RELATED WORK

In this section, we review related work on data partitioning and secure parallel processing on encrypted cloud data using homomorphic encryption.

In paper [1] shows fully homomorphic encryption scheme enables to operate multiple types of operation on encrypted data. "Ryan Hayward, Chia-Chu Chiang" [1], presents a work processing the data on multiple nodes by parallel processing the encrypted data using fully homomorphic encryption. In this work, they used the gentry's algorithm to perform FHE. The parallel processing will decrease the time taken to perform the applied operations on encrypted data in a cloud environment. The fully homomorphic encryption is performed on the multiple nodes to reduce the processing time. This work is done on a private cloud using gentry's algorithm.

In paper [2] shows user stores their data on cloud and they want to protect those data from the third party attacker or unauthorised users. So users need security to their data that is stored on cloud. On cloud storage, security is a one of the major issue. There are several encryption methods are exist, used for secure the user's data that is stored cloud. Some methods are like Full Disk Encryption and Fully Homomorphic Encryption. Samjot Kaur, Vikas Wasson presents a work on homomorphic encryption and they used the Diffie Hellman algorithm for symmetric key agreement. Diffie Hellman algorithm is a key exchange algorithm. When two authorised parties want to communicate each other, this algorithm create session key between them. And it also creates "HMAC" for the user's data integrity and "One Time Password" for more security.

In paper [3] cloud computing provides the on-demand services to the users of cloud. Users are charged as per payper-use model. In this paper, Mbarek Marwan, Ali Kartit and Hassan Ouahmane present a work based on the homomorphic encryption technique to secure the client data. And they also show performing the arithmetic operations (addition and multiplication) on encrypted data. RSA algorithm is used to processes the multiplication computation on encrypted cloud data because RSA is a multiplicative homomorphic encryption. Paillier encryption is used to apply homomorphic addition operation on encrypted data.

In [4] this paper presents a medical application. They used the homomorphic encryption technique to allow computation on encrypted cloud data without decrypting the cipher text. And also they describe about homomorphic encryption roles on encrypted data; it will provide privacy data sharing and confidentiality of data on cloud environment. In this they show partial homomorphic algorithms to perform arithmetic operations on encrypted cloud data. This proposed medical operation is used to process the sensitive patient's data that is stored on cloud.

3. DESIGN AND IMPLEMENTATION:

A. Architecture of Parallel homomorphic Encryption: The basic working principle of the system is shown in figure 2. The architecture comprises three entities: Client, Computation Dispatcher, and Computation Servers. Each entity is briefly described below.

Client: The client is one who wants to encrypt the file, upload on different cloud server and apply the operations on encrypted data.

Computing Dispatcher: It provides the services to store and manage client data. Computing dispatcher receives data from the client, divide the data and store on different computation server. Here it takes parallel processing of encrypted data.

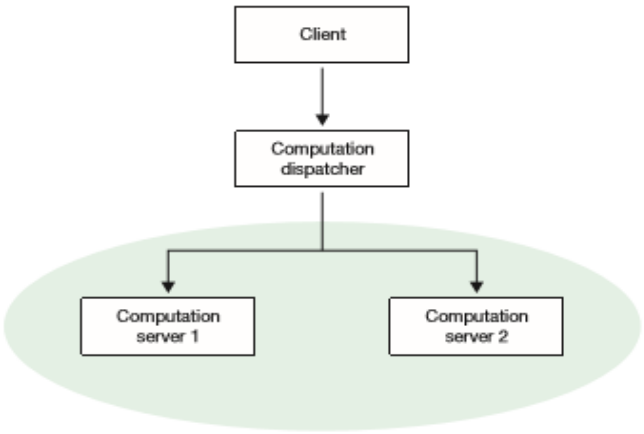


Fig. 1. Client-server model.

Computation Servers: Each computation severs perform the applied operations on the client's encrypted data in parallel. And return the result back to the computation dispatcher.

Gentry's algorithm is used in this work. In this encryption scheme Gentry's method uses the bootstrapping procedure to reduce noise in the process of fully-homomorphic encryption. Gentry's encryption scheme shows that, it will took few seconds to perform two 8-bit integers subtraction, addition and comparison arithmetic operations. And this algorithm also shows, it took few minutes to perform multiplication operation on two 8-bit integers and for division operation it took hours

Parallel processing process the operations encrypted cloud data on multiple nodes using fully-homomorphic encryption scheme, it reduces the processing time. "Ryan Hayward, ChiaChu Chiang [1]" (2013a, 2013b) presents the work on parallel processing of fully-homomorphic encryption in private cloud using OpenStack. Here I am using java programming language and stand alone function.

A client-Server model is shown in below diagram Fig. 2.1; it shows the parallel processing of fully-homomorphic encryption.

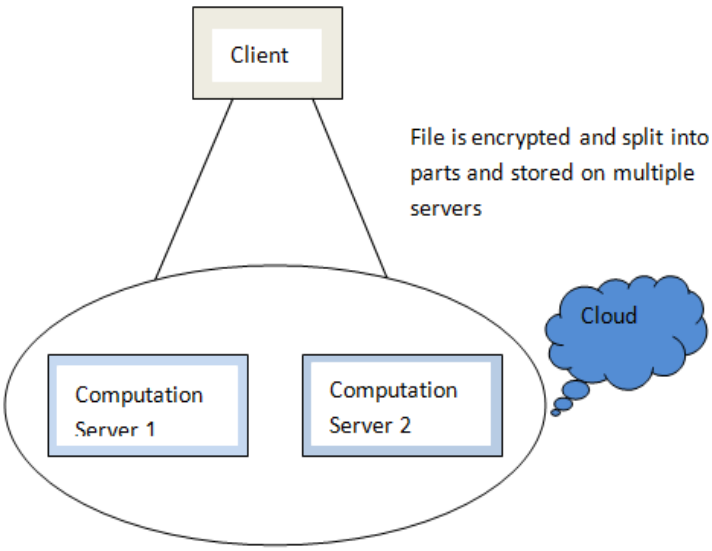


Fig.2.1.Client-ServerModel.

Here shows that the client inputs the set of data (in the form of integers) and those integers are encrypted and split into multiple parts (for integers it will split into pair wise). After splitting inputted data stored on multiple servers. Each computation server performs the applied operations.

C. Data Partitioning To Improve the Security of Cloud Storage:

Here another concept presenting the work on data partitioning and storing on multiple nodes. To improve the security of cloud data storage using the data partition method. Here client inputs the data, encrypt the data and divided into multiple parts based on the size of user data. Divided data will store on different cloud servers. This process is shown in below block diagram Fig. 2.2. In this figure shows block diagram of data partition method. First client encrypts the data and generate a public key to store and retrieve the client data to (from) the different cloud servers. Next divided inputted data into multiple parts with equal size and also based on number of servers available to store the client data. And next store the divided data on different servers. When client want that data back from the cloud, client should enter file name and public key to get the original data. Finally the decrypted result gives back to the client. Partitioning method take an important role in this process. It divides (splits) the client data into equal size of multiple chunks to store on different cloud servers. And it also gives easy access to authorised user when that data need.

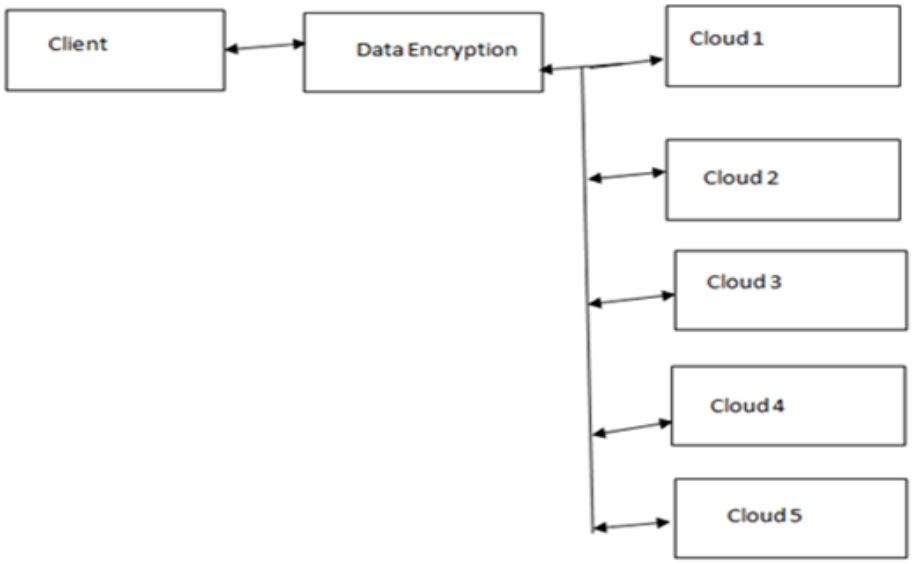


Fig. 2.2. Block Diagram of Data Partitioning Method

D. Implementation Modules:

a) Files Split

b) Storing to the cloud

c) Homomorphic

d) Retrieving the file

e) Integers based

a) Files Split:

Here first client can inputs the data by browsing the file, which is available in the client system. And also generates the public key. Secret Splitting is done in this process, where secret information between the two or more individuals. The inputted data will be split and stored on different servers; it yields more security to the

client data. All the individuals should be agreed and shared secrete data to merge the individual parts to get the original data. Fig.2.1. Shows the splitting data and storing on different servers

b) Storing to the cloud: After partitioning the client, each part is store on different servers. Each server on cloud contains a part of file.

c) Homomorphic Technology: Homomorphic encryption method enables to process the operations on the encrypted cloud data. After processing the operation, then decrypted result will be same as the result produced by the applying same operation on the plain text data (unencrypted data).

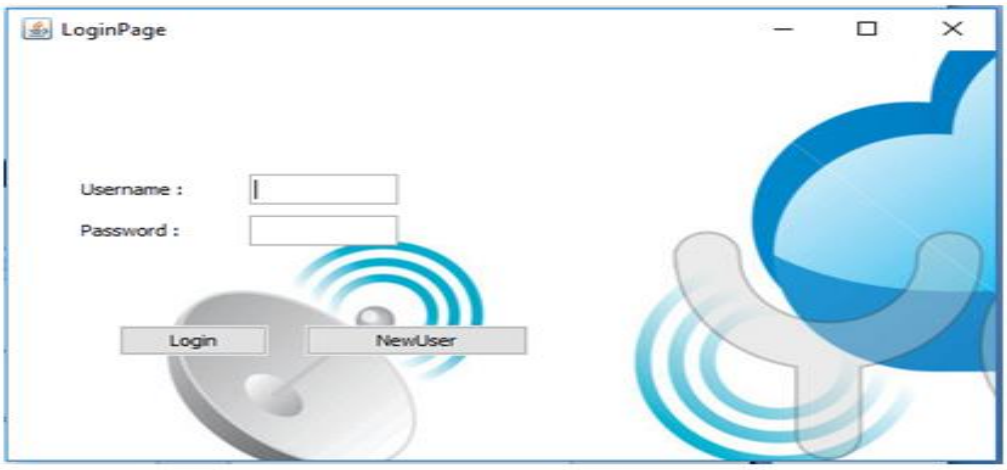
d) Retrieving the file: To retrieve the file from different servers, authorised client should send a particular file name and public key to be fetched from the different servers. Then servers received the file name from the authorised client and match the file name with the files that are available in the storage of servers. And then matched file content and will send to the client.

e) Integers based: Here first client can input the eight integers (8-bit). And those integers are split and store on different servers. Each computation server performs the applied operation on the stored data. Here shows an example of addition operation on encrypted cloud data and fig.2.3. shows the process of this operation. First client inputs the 8-bit eight integers and addition of these integers was taken by dividing the 8-integers into 4-pairs and addition of each pair on different nodes. Then resulting will be 4-integers, again these 4-integers are splitting into 2-pairs and finding the sum of each pair. And so on. This addition operation on multiple nodes shows the parallel processing on different servers. This parallel processing operation decreases the processing time. The vector product is done by first applying the pair wise product, and then resulting integers are summed.

E. Results and discussion Home page:

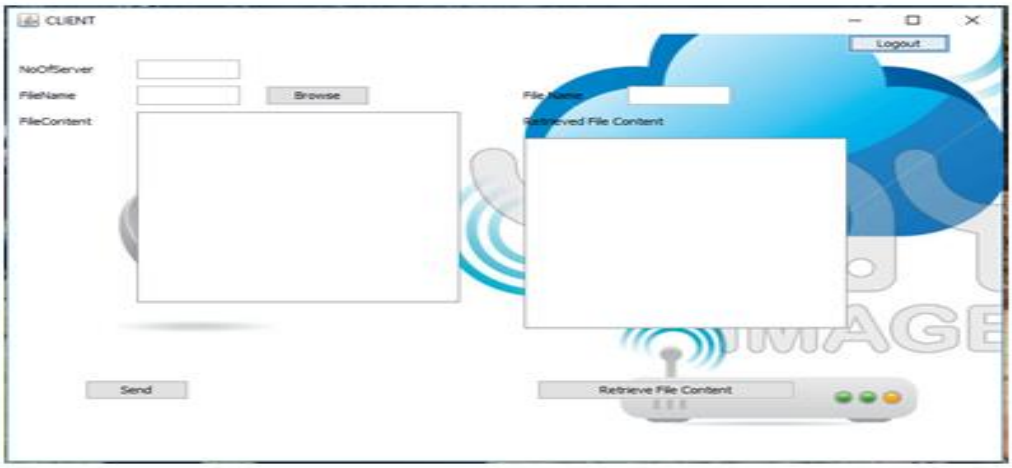
The user needs to register with his personnel information. If account is already registered than user can go to login page. New user should register first by using his personnel information such as name, password, email id, etc.

Login Page: After registration client can log in through using username and password. This is the authentication process; an authorised user can only be log in using registered username and password. Client login page is shown below.



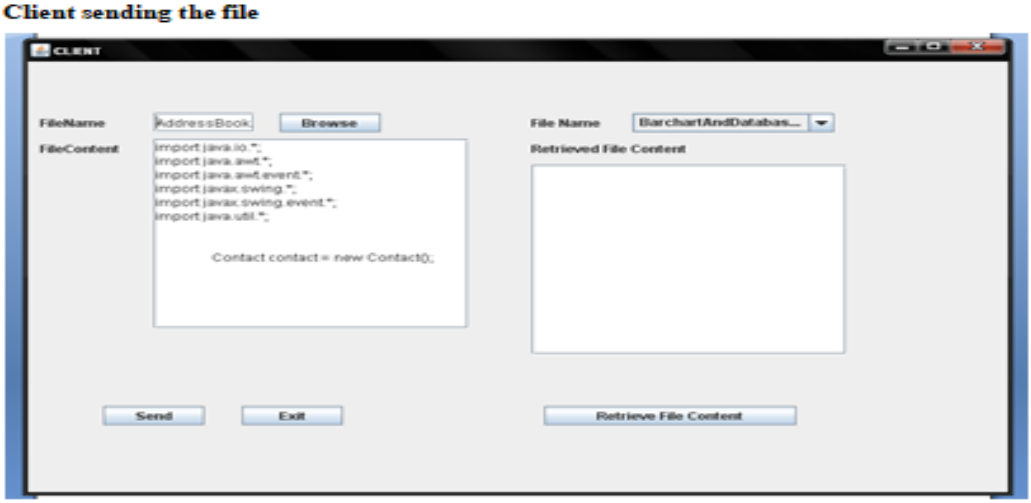
Client Dashboard

Below shows the client dashboard, this will be displayed after authentication of the client using username and password. Here client can upload the text file and also retrieve the uploaded his own file. Here generates public key for security purpose to the client data. While retrieving the client data, client should enter the file name as well as public key.



Client sending the file

Here shows the client browse and upload the text file and also generates the public key. After that file will divided and send the divided file to different servers.

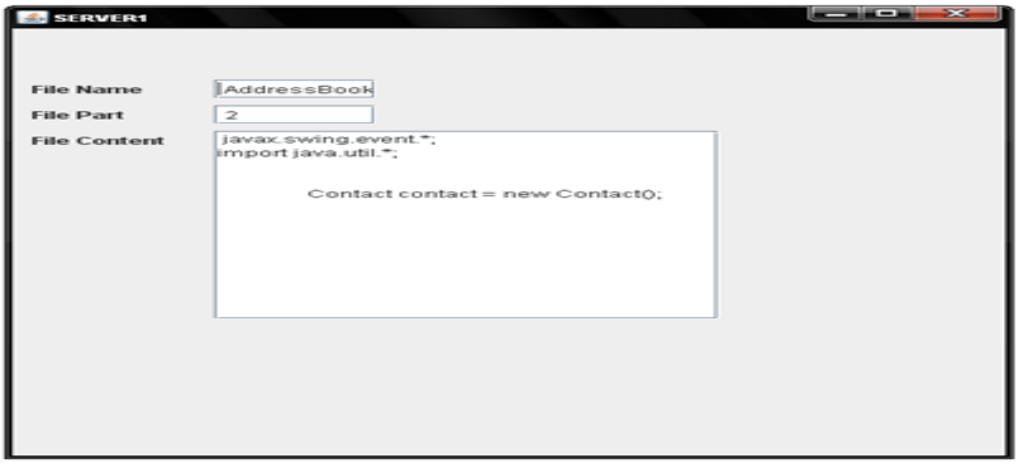


Server receiving part of a file

Here shows two servers receiving the file part from the client.

Another Server receiving other part of a file

Here shows another part of file storing on another server, the servers are created based on the client entered in the number of server box.

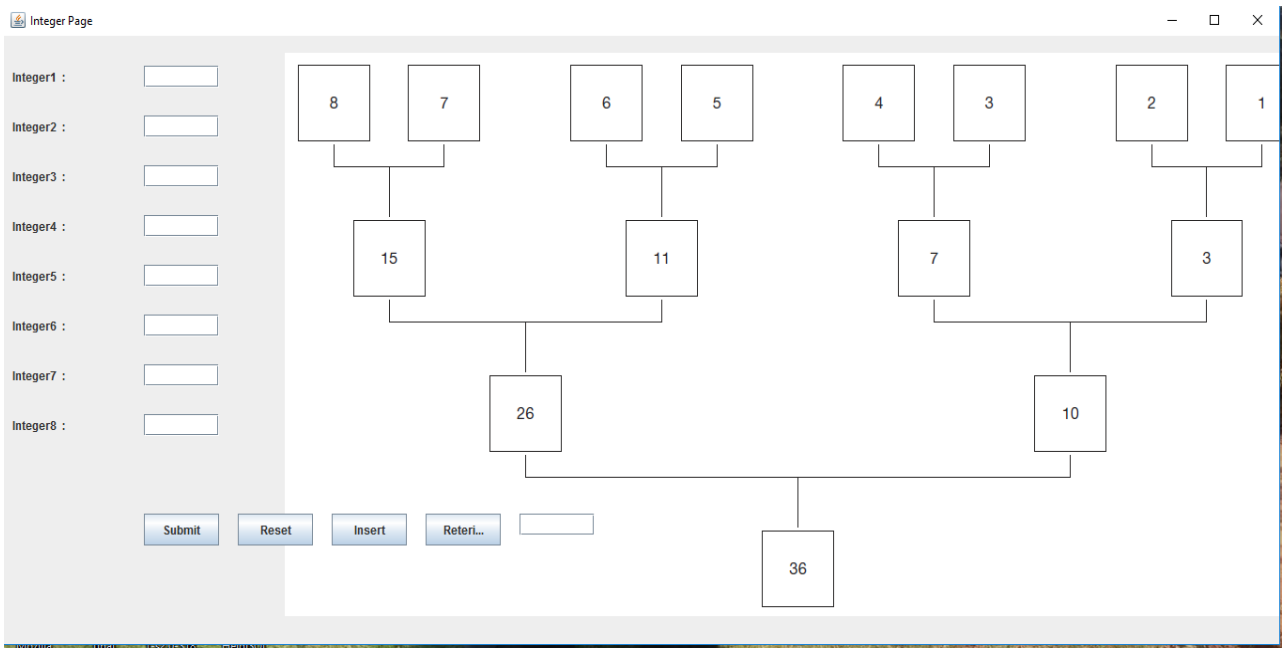


Retrieving the file from servers

When client want his data back from server, he should enter the file name of the uploaded file and public key of that file. This entered file name will match the files are stored on different servers and merge the content from different server. And display the content of that file to the client.

Addition Of integers

Here shows addition of integers using fully homomorphic encryption on multiple nodes. First authorised client can login through using username and password. Here client can enable to enter 8-bit eight integers. After client entering the eight integers he can click the insert button to store on different cloud servers. Here those inputted integers are split into 4pairs to perform the addition operation and also data dependency graph will be created to perform the addition of encrypted integers. Each child node is depends on the output of the parent node.



4. CONCLUSION

Cloud computing is used to store up and process the huge data and programs of cloud users. Homomorphic encryption provides data privacy and data confidentiality. Fully homomorphic encryption supports to execute various types of operations on encrypted cloud data. In this work presents the secure parallel processing on encrypted cloud data using FHE. Parallel processing means performs the operations on multiple nodes. This parallel processing produces the better performance than the computing the same operations in sequential process. The final result shows the improvement in the processing time means parallel processing of FHE decreases the processing time of performing operation on the encrypted cloud data. Data Partitioning method provides more security to the client data on cloud. This process generates public key; it is used to store and retrieve the data from cloud storage. Client data is dividing into multiple chunks with equal size of chunks. And each part is stored on different servers. While retrieving the data from cloud, authorised user should enter the public key. This method shows the data confidentiality.

REFERENCES

[1] Ryan Hayward, Chia-Chu Chiang , "Parallelizing fully homomorphic encryption for cloud environment", ScienceDirect 2015 Journal of Applied Research and Technology 13 (2015) 245-252.

[2] Samjot Kaur, Vikas Wasson, "Enhancement in Homomorphic Encryption Scheme for Cloud Data Security", IEEE 2015 9th International Conference.

[3] Mbarek Marwan, *Ali Kartit and Hassan Ouahmane, "Applying Homomorphic Encryption For Securing Cloud Database", 2016 IEEE.

[4] Yasmina BENSITEL , Rahal ROMADI, "Secure data storage in the cloud with homomorphic encryption", 2016 IEEE.

[5] Mr. Manish M Potey, Dr C A Dhote , Mr Deepak H Sharma,"Homomorphic Encryption for Security of Cloud Data", ScienceDirect 7th International Conference on Communication, Computing and Virtualization 2016.

[6] Monique Ogburn , Claude Turner, Pushkar Dahal, "Homomorphic Encryption", ScienceDirect .

[7] Santosh Kumar Majhi, Sunil Kumar Dhal, "A Study on Security Vulnerability on Cloud Platforms", ScienceDirect International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA.

[8] Farhad Farokhi, Iman Shames, Nathan Batterham, " Secure and Private Cloud-Based Control Using Semi-Homomorphic Encryption", ScienceDirect IFAC-PapersOnLine 49-22 (2016) 163–168.

