RESEARCHERID
THOMSON REUTERS

[Kalaskar, 8(2): February 2021] ISSN 2348 – 8034
DOI: https://doi.org/10.29121/gjesr.v8.i02.2021.1 Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

RETRIEVAL MODEL FOR CLOUD VIRTUAL ENVIRONMENT USING DISTRIBUTED FILE SHARING

Dr. Keshao D. Kalaskar*[1], Ms. Shipra Yadav[2] & Dr. Pankaj Dhumane[3]
[*1]Associate Professor, Dr. Ambedkar College, Chandrapur, (MH), India
[2]Research Scholar, IICC, RTM Nagpur University Nagpur, (MH), India
[3]Assistant Professor, Sardar Patel College, Chandrapur, (MH), India

ABSTRACT

Cloud based services are mostly used for data storage. Several problem occur pertaining to data storage and to cope up with those issues security is being addressed. Our Aim this paper is to provide the architecture for user splitting data and the solutions in retrieving different data chunks. Stored on different Cloud storage with the help of this not only the single served stress is reduced but security and storage are efficiently used . Data would be stored and retrieved in slices hence the chance of data fraud is dismissing. File processing would be faster as split part would be fetched from different cloud enabling parallel processing.

Keywords: *File sharing system, Cloud security, data sharing.*

I. INTRODUCTION

Cloud services is being used my Cloud computing is being adapted by many organizations due to its dynamic scalability and virtualized resources [1]. Cloud computing, particularly cloud storage, has been proven in multiple real-world cases to simplify information technology (IT) operations helping companies to save millions of dollars. The era of big data would not be possible without the emergence of highly scalable cloud storage [2]. Cloud storage provides ease to users in a variety of ways like providing faster accessibility, rapid deployment, accessibility, and data security, backup and recovery. Due to the extensive use of cloud, problems pertaining to it are becoming noticeable [3]. The load on the servers is getting heavier and the system must look for more advisable methods to minimize it. For that purpose, distributed file systems and cloud storage come handy and address many of the problems discussed above. In these, a dataset is divided into small chunks and these chunks are then stored over different servers to minimize the load on any single server, hence increasing efficiency and security [4]. Goals that have been achieved by data distribution include fault tolerance, ability to store bigger data and managing them efficiently, append operations on file allowance, and reliable communication among different machines. However, attention is still needed on the domain of retrieval of distributed data in their complete original form. This project focuses on data, stored onto different servers, distribution and retrieval. This can be achieved by the help of an information retrieval method which will facilitate us by maintaining the order of data slices for the same file, so while fetching they can be arranged into the complete original file. Several schemes have been proposed in context of file splitting and sharing, but the proposed scheme focuses on the following points.

- To design a hybrid multi-cloud model to improve cloud efficiency and provide a secure environment.
- A data distribution technique and splitter is used to encrypt and split data into chunks.
- To develop an efficient and intelligent method for the retrieval of split file chunks into its original form.

II. RELATED WORK

Security as a service (SaaS) provides the opportunity to divide data into chunks for security enhancement. When they are divided, each chunk is then encrypted and stored onto different databases [5]. But this technique has a serious issue which is the linked list based distribution of the data. The tail of each node of the linked list contains information of the next node. So, if one node is hacked, it will give information of all the other nodes.

[Kalaskar, 8(2): February 2021] ISSN 2348 – 8034
DOI: https://doi.org/10.29121/gjesr.v8.i02.2021.1 Impact Factor- 5.070

Security is provided by isolating the process of decryption and encryption to a third party, so the data is encrypted using a secured co-processor [5]. Cloud systems have provided ease to users in many ways like storing data, running applications, data recovery, and flexibility. Cloud also includes risk of data integrity, network dependency and centralization. This is the main reason many why big companies restrain from using clouds as primary storage. CSA, ENISA and NIST published general security guidance and recommendations, for cloud usage to provide some level of protection ranging from physical security to network/system/application security [6]. The idea of distributed computing turns is gaining popularity over the last few years. Data storage is a critical and important research field in distributed computing. In [4], authors presented the idea of distributed computing and distributed storage and the design of distributed storage right off the bat. Authors in [8] developed a dynamic load balancing algorithm to balance the load across the storage nodes during the expansion of private cloud storage. Authors in [9] addressed encryption challenges by the use of tanked searchable symmetric encryption. Authors in [7] implemented data integrity protocols to detect data corruption. Authors in [10] discussed cloud computing and its service models, cloud security issues, challenges, and analyzed various solutions with TTPA and studied their benefits in terms of data integrity, access control mechanism, and data confidentiality. Authors in [11-14] provided a KNN classification method, a privacy preserving protocol which accesses data in database using encryption and solved input record query of data mining.

III. RESEARCH METHODOLOGY

A. Data Splitting and Encryption Techniques

Cryptography is a secure communication technique used when there is a presence of malicious third-parties—known as adversaries. Encryption which is a main module of cryptography uses an algorithm and a key to transform an input known as plaintext into an encrypted output called ciphertext. Algorithm will always decrypt the cipher text/block into plaintext if the same key is provided. The encryption algorithm is considered secure because if any of the encrypted data (cipher block) is lost, information from this block cannot be retrieved until the encryption key is given.

1) Data Encryption

- **Symmetric encryption**: This is the simplest kind of encryption that involves only one secret key to cipher and decipher information. Symmetrical encryption is an old and known technique. The secret key that can either be a number, a word, or a string of random letters. It is blended with the plain text of a message to change the content in a way . The Sender and the Recipient should know the secret key to encrypt and Decrypt the message.
- **Asymmetric encryption**: It uses two keys one to encrypt the data other to Decrypt these interdependent keys are generated together. One is label as public key and is distributed freely. The other is labeled as Private Key and must be Kept hidden.
- **Hash** : It is a mathematical function that takes input data of a arbitrary length. And then generates. Fixed length hash based on the input. It is easily calculated but it is very difficult to generate the original data if the hash value is not known.

2) Data Splitting:

Cryptography splitting is an algorithm that splits the input/data into a number of chunks. Splitting is done at bit level. A secure key is used to control splitting and splitting is done randomly.

3) Data Security Algorithm

Blowfish, AES, RC4, DES, RC5, and RC6 are used for encryption to enhance security. The most widely used algorithms are AES-128, AES-192, and AES-256. The technique that we are using is the AES (advanced encryption standard)-256 which has the advantages of low memory cost, high speed, same key used to encrypt and decrypt, and the cipher and plain blocks are of the same size

[Kalaskar, 8(2): February 2021] ISSN 2348 – 8034
DOI: https://doi.org/10.29121/gjesr.v8.i02.2021.1 Impact Factor- 5.070

4) Cryptographic Data Splitting Algorithm Pseudo code

START
Get filePath
CALL: doEncryption
DECLARE counter: 1
DECALRE sizeOfFile: 1024 * 1024
DECLARE buffer: ARRAY[sizeOfFile] DECLARE filename: getFileName() DECLARE bytesAmount: 0
DECLARE bufferInputStream: bis
DO UNTIL bytesAmount=bis.READ(buffer)> 0
DECLARE filePartName: (filename + counter) DECLARE File newfile: File(filePartName)
DECLARE fileOutputStream: out(newfile)
out.write(buffer, 0, bytesAmount)
WHILE LOOP END
STOP

5) Merging of File Chunks Algorithm Pseudo Code

START
Get FileChunks CALL: doDecryption DECLARE File: into
DECLARE FileOutputStream: fos(into)
DECLARE BufferOutputStream: mergingStream(fos);
REPEAT FOR File f: FileChunks Files.copy(f.toPath(), mergingStream) END FOR LOOP
STOP

6) Distribution of File in to Chunks

The uploaded will be first encrypted and then broken down in to a number of chuncks according to number of public cloud available.

7) Placement of Chunks on different Clouds

These encrypted chunks will be stored on different public clouds and the server ID of each chunk on which it is stored is fetched

8) Maintaining Log File

A log file will be maintained having information about the number of registered users and uploaded files, along with the number of chunks and the cloud reference on which is stored for each chunk. Figure 1 shows the system flow of the proposed technique. Figure 2 shows the architecture of the proposed technique.

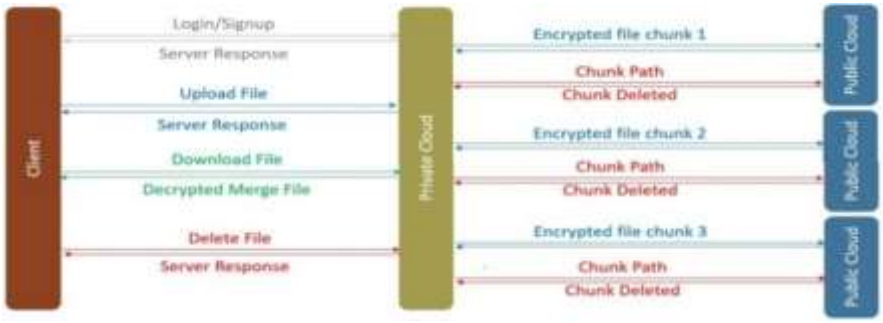

Fig. 1. System flow diagram

[Kalaskar, 8(2): February 2021] ISSN 2348 – 8034
DOI: https://doi.org/10.29121/gjesr.v8.i02.2021.1 Impact Factor- 5.070

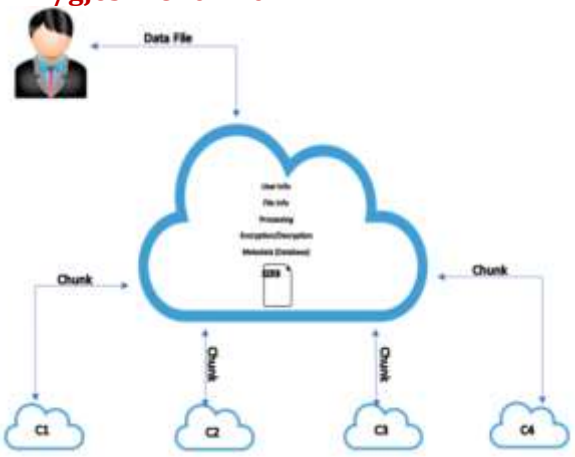

Fig. 2. Proposed model architecture

IV. RESULTS AND DISCUSSION

In this section, the proposed model is compared with the plain file system. The comparison is based on time. The total time taken by the proposed model is the accumulated sum of receiving, encryption, splitting and transferring times.

A. Proposed File Distribution Model

Table I shows the total time taken by different modules (receiving, encryption, splitting and transferring) when users upload files of different size. The smallest file (3.73MB) takes a total processing time of 17499ms whereas the largest file (25.3MB) takes52638ms. However, total time also depends on bandwidth, CPU, and memory.

Table 1 : file processing time of fdm

File Size (MB)	Receiving time (ms)	Encryption time (ms)	Splitting time(ms)	Transferring time(ms)	Total time(ms)
3.73	12013	527	15	5471	17499
18	63003	103	58	13925	76986
17.8	59720	118	54	20457	80231
25.3	41749	594	67	10822	52638
16.2	28409	97	91	8891	37391

B. Plain File Distribution System

Table II shows the total time taken by the plain file distribution system (PFS) when users upload files of different sizes. Total time is inversely proportional to the bandwidth.

Table II file processing time of pfs

File size (MB)	Receiving time(ms)	Total time (ms)
3.73	12013	12013
18	63003	63003
17.8	59720	59720
25.3	41749	41749
16.2	28409	28409

[Kalaskar, 8(2): February 2021] ISSN 2348 – 8034
DOI: https://doi.org/10.29121/gjesr.v8.i02.2021.1 Impact Factor- 5.070

A detailed comparison between the proposed and the plain file system is shown in Figure 3. The proposed system looks a bit complex and took more processing time as compared to the PFS. It is understood that computational complexity can be compromised for data security purposes. The proposed system is more secure than the PFS.

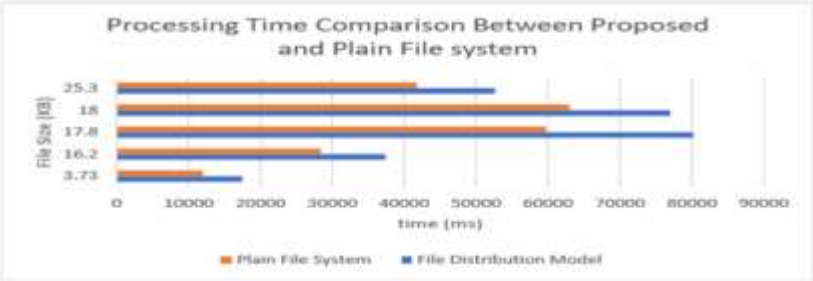

Fig 3 Processing time comparission between Proposed and Plain File System

C. Result Analysis

From the above results, it is concluded that the total time taken by the proposed distributed file system is increased by only a few seconds. This time is utilized in taking security measures (i.e. encryption, splitting) and in efficient storage utilization. To achieve high security, one must compromise on time. Table III shows the time consumption difference between the proposed and the plan file system. The proposed distributed file system is technically compared with Google file system (GFS) and the SaaS model. Table IV shows the comparisons. Table V shows the comparison of the proposed distributed file system with the SaaS model.

Table III Result and Analysis

Total time of the processed file System(ms)	Total time of Plain file system(ms)	Difference(ms)
17499	12013	29512
76986	63003	1398
80231	59720	20511
52638	41749	10889
37391	28409	8982

Table IV Comparison of proposed model with GFS

GFS	PROPOSED MODEL
Consist of single master server, multiple chuck server and clients	Consists of private cloud along with multiple public clouds
Master manages all the system data and file directory structure	Manages all operations including signup and login requests, file distribution encryption ,maintaining log file and deploying chunks on to public servers
A clients interact with the master server for metadata only and interact with chunk server directly for all other data	A client interact with the private server only. A client is not required to communicate directly with the public server because the private server will be responsible for carrying out the client's request
Divide file in to fixed size chunks of 64MB each	There is no restriction on the chunk size. The file distribution is based on the number of public clouds available.
Each chunk is replicated on three public chunks server	There is no replication of chunks ensuring data integrity

Table V Comparission of proposed with saas

Saas	Proposed Model
Has three major parts (header , tail and encrypted data)	Uses only Encrypted data.
Header contains all important details including encryption techniques, total number of chunks, total size of chunks chunk number and a unique user id.	Log file (database) will maintain all the important details
Tail will contain reference to the next chunk	Next chunk can be identified from the log file
A third party does encryption and decryption. User can select between multiple encryption techniques	Private cloud is responsible fir encryption and decryption. User has no choice over the encryption techniques.
Log file contain the record of the user data and help to identify the first chunk	Log file will contain the total number of chunks along with server IDs
Log file will be maintained in master database and all the data will be dumped in to random databases.	Log file will be maintained in the master cloud and all the data chunks will be saved in to slave clouds
Enhance the security of the encrypted data by distributing the data with in the cloud	Master and slave clouds. User can directly interact with the master cloud only. Communication takes place between master and slave clouds.

The proposed model has some characteristics which differentiate it from the previous models/techniques. These are:

- User cannot directly communicate with slave clouds, which ensures data security.
- The proposed model does not give privilege of encryption technique to user which ensures reliability.
- The proposed model contains all information about data chunks and their order in secured log Files
- The proposed model doesn't involve any third party for encryption which is more trustworthy where SaaS model depends on third parties for encryption techniques.
- The Proposed Model is a Hybrid model consisting of multiple cloud in which one is the master (Private) and other are slaves(public)

V. CONCLUSION

The distributed file system is a very efficient way of portioning data and storing them to multiple clouds. In that way the load on a single cloud is reduced and the performance is increased. It also helps in creating a secure way for file retrieval and processing, hence managing and protecting user's data. The main concern of the project is the efficient distribution of data and their retrieval in correct order.

REFERENCES

1. *R. M. Babu, "Secure Storage and Retrieval of Big Data on Distributed Cloud Environment", International Journal of Applied Engineering Research, Vol. 11, No. 10, pp. 7063-7071, 2016*
2. *J. Broberg, S. Venugopal, R. Buyya, "Market-oriented grids and utility computing: the state-of-the-art and future directions", Journal of Grid Computing, Vol. 6, No. 3, pp. 255–276, 2008*
3. *J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, R. Bayya, "Ensuring Security and Privacy Preservation for Cloud Data Services", ACM Computing, Surveyss, Vol 49, No 1, Article No 13, 2016.*
4. *K. Liu, L. J. Dong, "Research on Cloud Data Storage Technology and Its Architecture Implementation ", Procedia Engineering, Vol 29, PP- 133-137, 2012*

5. *C. P. Ram, G. Sreenivaasan, "Security as a Service (SasS): Securing user data by coprocessing and distributing the data" Trendz" in informationscience and computing, Chennai India December17-19,2010.*
6. *T. Sivashakthi, D. N. Prabakaran, "A Survey on Storage Techniques in Cloud Computing International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 12, pp. 125-128, 2013*
7. *V. Ukey, N. Mishra, "Dataset Segmentation for Cloud Computing and Securing Data Using ECC", International Journal of Computer and Information Technologies, Vol. 5, No. 3, pp. 4210-4213, 2014*
8. *K. Gabhale, N. Jadyal, A. More, V. Bhalekar, V. V. Dakhode, "Data Partitioning Technique to Improve Cloud Data Storage Security", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 3, pp. 24-26, 2016*
9. *N. M. Anuradha, G. A. Patil, "Secure and Efficient Data Retrieval in Cloud Computing", International Journal of Engineering Research & Technology, Vol. 4, No. 4, pp. 2278-0181, 2015*
10. *P. H. Atulkumar, P. A. Atulkumar, "A Review of Addressing Storage Correctness in Cloud Computing with Trusted Third-Party Auditor", International Journal of Research in Engineering & Advanced Technology, Vol. 1, No. 2, pp. 1-4, 2013*
11. *V. Goutham, P. A.. Reddy, K. Sunitha, "K-Nearest Neighbor Classification on Data Confidentiality and privacy of user 's input Queries", International Journal of Advanced Research in Computer and Communication Engineering, Vol 5,No 7, PP 704-708,2016*
12. *Vasam Varshini, B. Sridhara Murty, " Efficient Data Storage Mechanism in Cloud Computing " , Journal of Global Journal of Engineering Science and Researches, Vol Varshini 5(6), June 2018, pp.16-21*
13. *Bhavya Shree MN, K.S. Anand Kumar " Accessing Data in Cloud Platform using identity and providing Strong Security Auditing Based on IBDO Scheme ", Journal of Gobal Journal of Engineering Science and Researches, Vol (ICRTCET-2018) , PP 54-61.*
14. *Puthviraj.B. Pawar & Hitendra Chavan , "Data Depulication of Protected Data in Cloud Computing ", Journal of Global Journal of Engineering Science and Researches , Vol (NCRAES-20), PP -145-151.*