Design and Implementation of Secure File Storage using Distributed Cloud Mechanism

Manoj V. Bramhe[1] & Dr. Milind V. Sarode[2] & Dr. Meenakshi S. Arya[3]
[1]Phd Research Scholar, Department of Computer Science & Engineering G.H. Raisoni College of Engineering, Nagpur.
[2]Professor and Head, Department of Computer Engineering , Government Polytechnic, Yeotmal.
[3]Associate Prof. & Head, Department of Computer. Science & Engineering, G.H. Raisoni College of Engineering, Nagpur.

Received: January 09, 2019 Accepted: February 09, 2019

ABSTRACT: *Cloud computing and the cloud based services like storage and application are the most demanding resources over the internet from few years. With better network or internet connectivity these become very easy tools for remotely managing the information and the files. As the system introduced it comes along with multiple issues and threats also. Even after having better network security mechanism, data protection algorithm, information privacy system service provider can't ensure full proof solution for information security. Let's ignore the security about the network transfer and other part; finally file resides at particular storage with company ownership which may be called storage service provider and the company employee with higher authorities having always access to customer's files. Here the proposed solution deals with the malicious user attack security and prevent file access by utilizing the multiple cloud environment for storing files.*

Key Words: *Cloud Computing, DFS, API*

I. INTRODUCTION

Cloud Computing is popular model for distributed data storage due to its various advantages like on demand network access with minimal cost as per user's usage but it faces various security challenges as whole data is available to cloud service provider. Many solutions were proposed for maintaining security and privacy in cloud like encryption, replication, VM isolation, etc [3] but most of which have focus on single cloud environment where whole information is stored on single public cloud which creates security concerns as system administrator has full access to data hence malicious administrator can use data for malevolent purpose, vendor-lock-in problem has loss of full information in case service provider stops his services, loss of data integrity due to modification of data by unauthorized persons [7].

All these issues with single cloud provider can be removed by moving to multi-cloud services wheredata is distributed partially among various clouds hence no entity will ever get complete set of data a time. Distributed file system (DFS) are used for managing files from multiple hosts.

Our proposed solution is to provide secured and reliable storage service for multi cloud environment. It will fragment user files into multiple parts at user application layer which will be uploaded securely after applying encryption in multiple clouds. Only authorized user or owner of the files can download data with proper user authentication .

II.RELATED WORK

Many researchers have proposed various solutions to cloud storage but most of them are related to single cloud provider where even though cost of storage is less but has security concerns and storage failure issues due to vendor -lock-in problems. Some solutions pertain to multi-cloud storage system as mentioned in [2], [6],[8],[9] where data is fragmented into multiple chunks and stored on existing public / private cloud infrastructure. Authors in [4] proposed four type of architecture for maintaining security of data in cloud. Most of the papers like [11], [14], [16] have used first architecture of replication of application but our system uses combination of those architectures by not only replicating data but dividing it into multiple parts. Tahoe-LAFS scheme was discussed in [7] where open source distributed file system Tahoe is used instead of other DFS for reliability purpose. System uses encryption for confidentiality purpose and achieves recovery of data using secret sharing scheme. In [8] cloud storage service model for

inter and intra cloud is proposed at Iaas level. Different data chunks of file are stored in various VMs of single or multi cloud. User can store and retrieve data from multiple cloud .System uses user authentication followed by file splitting / file retrieval by cloud manager interface and then it will be handover to multiple clouds/users. System supports both inter and intra cloud operations but security is not enough as encryption methodology is not used. Some multi-cloud storage approaches as discussed in [16] and [17] are similar to our system but they had worked on integrity, availability and vendor lock-in problems and not paid attention to all the security parameters like confidentiality . Our proposed system handles all major parameters of security of maintaining confidentiality, availability and integrity of data.

III. PROPOSED SYSTEM

Looking toward the limitations, described in the literature review, it is necessary to have a novel universal procedure to overcome those limitations. As almost every big organization having web identity and runs on hosting or cloud environment, the organizational information becomes important aspect. Even after cloud provides industry a wide range of security and system free from malicious software virus attacks, still the data is not safe from malicious users having administrative rights. One having super user rights can access the data from cloud storage with wrong intensions. After going through different solutions to reduce malicious user attack we found keeping the data out of user reach will make data more secure than by any other way, but we can't forget the fact we have to keep the data somewhere. Hence the proposed system is designed by keeping these scenarios in consideration where we will distribute and secure the data at different location in order to hide original data directly from user. Here system will take smart decision on the basis of user request and split the data after successful encryption in to different blocks and store it on the multiple cloud storage, now malicious user can access the data but of no use. Another point of consideration is the safety of data in case of critical condition. The proposed system is less with capability to re calculate the data from rest of the storage if any one of the storage gets failure.

Figure 1 describes the proposed system layered architecture along with applicable methods and the DFS structures. Overall system designed to have independent development platform to design and develop multiple cloud based file storage mechanism, where application will only responsible for utilizing the DFS system and it's API set to access the file storage by providing proper credentials to the connect to multiple cloud storage. DFS methods or API set are designed for three main modules like, A) Encryption and decryption module B) Construction and deconstruction module C) File transfer module.

A) Encryption and decryption module: this module is mainly designed to utilize the encryption and decryption methods in order to encrypt the files or the file part to protect it from unauthorized read operations and later on decrypt the file to allow user to read the file as it was originally. This module utilizes the Advance Encryption Standard (AES) algorithm to perform the encryption and decryption task.

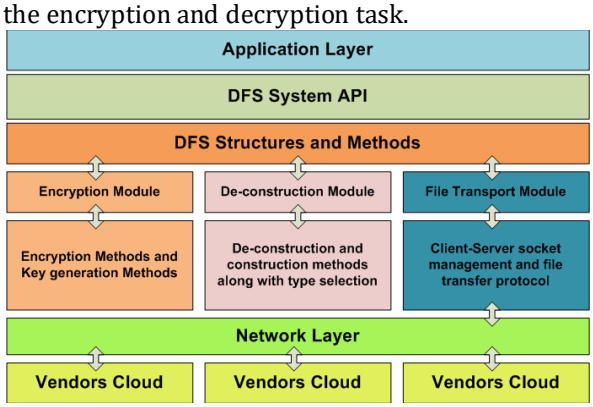

Figure 1: Proposed System Architecture

B)Construction and deconstruction modules: As the proposed system describes its approach to protect the file by dividing the file in multiple parts and store it at different locations or cloud storage. This module is used for fragmentation of original file into multiple parts as displayed in figure 2 . First step system does is to separate $100/3^{th}$ part of the file and saves this 1/3rd part as 1st part in local server. Next is to again split remaining 66.67% part in to 2 equal parts and store it in two different files in public clouds, This will work out in following ways,

Step 1: Collect the first part of the file in temp buffer,

- *Temp_Buff_1=MemStream.Take(100/3)*

Step 2: Write the Temp bufferin first part of the file,

- *FL.write(File_Chunk_A,Temp_Buff_1)*

Step 3: Collect remaining part of file for further separation,

- *Temp_Buff_2=MemStream.Skip(100/3)*

Step 4: Collect the second part of file for further separation,

- *Temp_Buff_1=MemStream.Take(100/2)*

Step 5: Write the Temp buffer in second part of the file,

- *FL.write(File_Chunk_B,Temp_Buff_1)*

Step 6: Collect remaining part of file as a last or 3rd part of the file in buffer,

- *Temp_Buff_1=MemStream.Skip(100/2)*

Step 7: Write the Temp buffer in third part of the file,

- *FL.write(File_Chunk_C,Temp_Buff_1)*

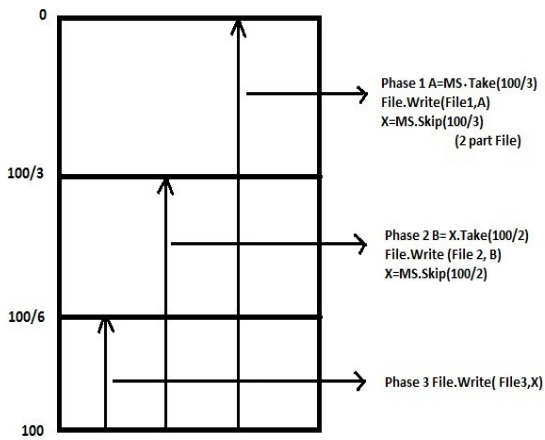

Figure 2:File Split Technique

Above mentioned techniques get performed in simple steps yet gives accuracy and stable performance almost every time and this allow system easy merging of file by simply copying all the parts sequentially in separate file in original state.

C)File transfer Module: This module is mainly deal with the transporting file over TCP/IP network using different protocols or services like FTP, Azure, and AWS. Even after this is the main based of the system.

IV. IMPLEMENTATION

Main idea behind the system is to implement complete system as API or SDK set so that it can be reused in multiple applications hence final implementation will come in the form of independent reusable packages like *.DLL in windows architecture. Every sub class or sub package will have different methods as per defined in DFS structures and methods.

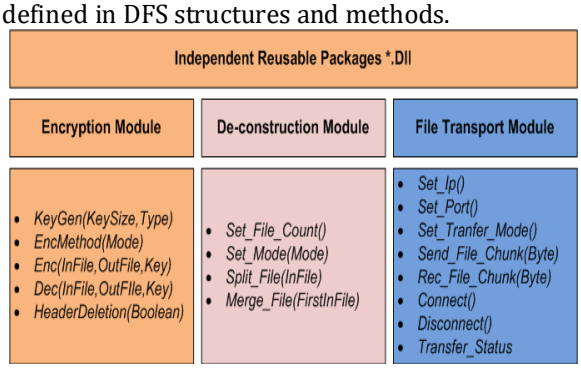

Figure 3: System API Set Model

All the modules defined are independent of another module as every module has its unique functionality and no integration required from other modules but system can't deny the fact that these needs to construct in proper sequence to get desired result. Below example explain process in detail,

K=KeyGen(Size,Type);
FL=Enc(OriginalFile,K);
P[]=Split(FL,3);

Where, P[0]: First part P[1]: Second Part P[2]: Third part

Finally,

Pt=Connect(IP_1);
SendFile(P[0],Pt);
Pt=Connect(IP_2);
SendFile(P[2],Pt);
Pt=Connect(IP_3);
SendFile(P[3],Pt);

This API sets are designed to allow user configure the connection parameters of multiple cloud file storage using FTP protocol mainly.

V. EXPERIMENTAL RESULTS

As the system deals with multiple level and system to execute the defined process calculations for execution time defers time to time and file to file. First file is uploaded to application server hence resources like CPU, Hard disk and network, comes in to the picture. Every resource has its own response time and instruction processing capacity. With the help of figure 4, system describes its multiple levels of system process time or the load.

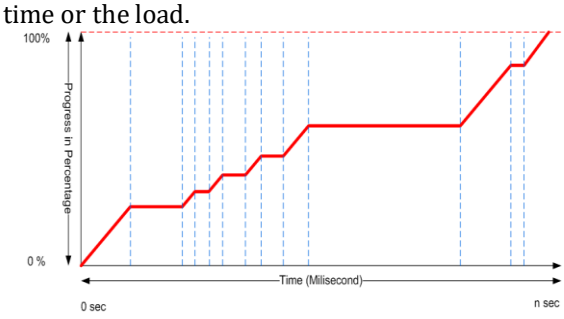

Figure 4: Time Progression Graph

Level 1: Connection time; in order to upload the file over network server first system need to establish the network communications with the server. Connection time differs over local network and internet speed and even in internet scenario connection state like sleep or active take another few seconds to wake up the network connection.

Level2: Handshake time; once the network communication is ready, every application protocols have its own overheads which may include authentication using credentials and this takes another few seconds to start actual data transfer methods.

Level 3: Transfer time; this is the main time required for actual file or data transfer takes place in the network. It may vary and totally depend on the size of the file currently queued for transfer.

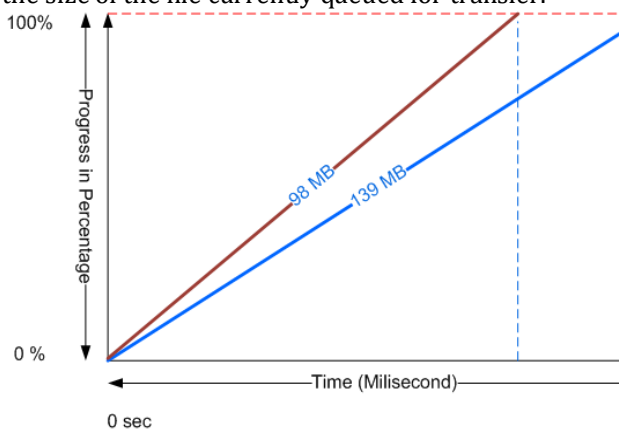

Figure 5: Transfer Time Analyses

Level 4: Network load; even if the transfer start over the network, it is a continuous process executes till the end of the file transfer, this may depends on the network load by other applications running in the network over same interface.

Figure 5 shows transfer time analysis for transferring the data over the network. Different time periods are required for transferring the load depending upon various network parameters discussed in above section. Thus our total time required to process the data is summation of connection time, handshake time, transfer time and network load time. All the above parameters affect the processing of data in distributed environment.

VI. CONCLUSION

We have implemented secured storage system using symmetric key algorithms. Our experimental results shows that many network parameters like connection time, Handshake Time and network load affects processing of files in cloud environment. Proposed system deals with splitting of file into multiple chunks and storing parts in multiple cloudswhich solves the file protection from malicious user but again the major problem with the system resources requirement is too high and dependence of different cloud storage service provider come across. In near future work we proposed a system to deploy malicious user attack protection over single cloud storage system.Byremoving the file ownership information mechanism system can prevent storage from malicious user also.

VII. REFERENCES

1. Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", IEEE Communications Surveys & Tutorials, March 2012
2. MohammedA.AlZain, Eric Pardede, Ben Soh, James A. Thom, "Cloud Computing Security: From Single to Multi-Clouds", IEEE 45th Hawaii International Conference on System Sciences, 2012
3. Singhal M., Chandrasekhar S., TingjianGe., Sandhu R., Krishnan R., Gail-JoonAhn., Bertino E., "Collaboration in Multicloud Computing Environments: Framework and Security Issues", IEEE computer society journal, Vol. 46, Issue 2, pp. 76-84, Feb 2013
4. Bohli J., Gruschka N., Jensen M., Lo Iacono L., Marnau N, "Security and Privacy Enhancing Multi-Cloud Architectures," IEEE Transaction on Dependable and secure computing, VolPP, Issue 99, 2013
5. Tran Doan Thanh, Subaji Mohan, EunmiChoil, SangBum Kim, Pilsung Kim "A Taxonomy and Survey on Distributed File Systems," IEEE Fourth International Conference on Networked Computing and Advanced Information Management, 2008
6. Su Chen, Yi Chen, Hai Jiang, Laurence T Yang, Kuan-Ching Li, " A secure distributed file system based on revised Blakely's secret sharing scheme," 11th IEEE international conference on trust, security and privacy in computing and communications, 2012
7. Fan-Hsun Tseng, Chi-Yuan Chen, Li-Der Chou, Han-Chieh Chao, "Implement a reliable and secure cloud distributed file system," IEEE international symposium on intelligent signal processing and communication systems, November 2012
8. ShushantShrivastava, Vikas Gupta, Rajesh Yadav, Krishna Kant, "Enhanced Distributed storage on the cloud," IEEE 3rd international conference on computer and Communication technology, 2012
9. KhengKok Mar, "Secured virtual diffused file system for the cloud," 6th International IEEE conference on internet technology and secured transactions, UAE, December 2011
10. RajkumarBuyya, Introduction to the IEEE Transactions on Cloud Computing, IEEE Transactions on Cloud Computing, Vol. No. 1, January –June 2013
11. NirnayGhosh, SoumyaGhosh, Sajal Das, "SelCSP: A framework to facilitate selection of cloud service providers," IEEE Transactions on Cloud Computing, Vol. 3, No. 1, January-March 2015
12. Chien-An Chen, Myounggyu Won, RaduStoleru, GeofferyXie, "Energy-Efficient fault-tolerant

data storage and processing in mobile cloud," IEEE Transactions on Cloud Computing, Vol. 3, No. 1, January 2014

13. Quanlu Zhang, Shenglong Li, Zhenhua Li, Yuanjian Xing, Zhi Yang, Yafei Dai, " CHARM: A Cost-efficient multi cloud data hosting scheme with high availability," IEEE Transactions on Cloud Computing, Vol. 3, Issue 3, July-September 2015
14. AlyssonBessani Miguel Correia Bruno Quaresma Fernando Andre Paulo Sousa, " DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds", ACM Transaction on Storage, Vol. 9,No. 4, Article 12. November 2013
15. Sancha Pereira, Andre Alves, Nuno Santos, Ricardo Chaves , "Storekeeper: A Security-Enhanced Cloud Storage Aggregation Service", IEEE 35th Symposium on Reliable Distributed Systems, 2016
16. Hussam Abu-Libdeh, Lonnie Princehouse, Hakim Weatherspoon, " RACS: A Case for Cloud Storage Diversity", International conference for Internet technology and Secured Transaction, December 2012
17. Kevin D. Bowers, Ari Juels, AlinaOprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage", 16th ACM conference on Computer and communications security, November 2009.