International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

Privacy Enabled Data Integrity System for Cloud Storage

Dinesh Hiregange[1], Dr. Chandramohan D[2]

[1] *M. Tech, Dept. of Computer Science and Information Technology, Dayananda Sagar University, Bengaluru, India*
[2]*Assistant Professor, Dept. of Computer Science and Engineering, Dayananda Sagar University, Bengaluru, India*

Abstract - *Cloud storage provides an on-demand data outsourcing service model, and is acquiring popularity due to its elasticity and low maintenance cost. However, security concerns arise when data storage is outsourced to third-party cloud storage providers. It is necessary to provide security for the user to verify the integrity of their outsourced data in the cloud, in case user data has been modified accidentally or by any attackers. In this paper, we introduce the privacy enabled data integrity system for regenerating coding cloud storage that have upload, download and repair operation and can achieve 50% fault tolerance. This project is very easy to implement and accomplish careful repair by XORing the chunks. Our system reduces the repair traffic and regaining from multiple storage server failures in a cloud storage location. To deliver fault-tolerance for cloud storage we have to divide data and send it to multiple cloud servers. If cloud storage server suffers from a permanent failure and losses all its data, we need to repair the lost data using other surviving cloud server. We have used FMSR methodology that aims at providing fault tolerance and reduces the storage capacity when storing files using multiple cloud storage server. FMSR uses proxy server that interconnects multiple cloud storage server which helps to achieve cost effective repair for permanent multiple cloud server failure. We have introduced AES encryption technique to encrypt the data in the cloud and MD5 hashing technique for data integrity verification.*

Key Words: Cloud storage, regenerating codes, privacy, Data integrity, network coding, fault tolerant system

1.INTRODUCTION

Cloud computing is an innovative proficiency in the field of information and expertise. It provides so many things in terms of "As-A-Service" basis. Cloud Computing is the basically visualization of computing as a utility, where customers can tenuously store up their data into the cloud so as to benefit from the on-request services from a collective pool of configurable computing assets.
Cloud storage provides an on-demand data outsourcing service model, and is acquiring popularity due to its elasticity and low maintenance cost. However, security concerns arise when data storage is outsourced to third-party cloud storage providers. It is necessary to provide security for the user to verify the integrity of their outsourced data in the cloud, in case user data has been modified accidentally or by any attackers. In this work, we concentrate on Privacy Enabled Data Integrity System for Regenerating-coding-Based Cloud Storage. To enabled privacy, our system uses AES algorithm which encrypt the data before uploading on to the cloud. Our system also uses MD5 algorithm to generate hash key which is used for data integrity checking.

A single-cloud storage provider may get problem of single point of failure and vendor lock in. A feasible solution is to stripe data across multiple cloud vendors. However, if cloud storage suffers from a permanent failure then the data on a disastrous cloud will become permanently inaccessible. In order to protect our valuable data against such failures, it is required to maintain data redundancy and fault tolerance. A repair process regains the failed data from existing surviving nodes over the network and rebuilds the lost data in a new node. The system uses semi-trusted proxy which is used to achieve fault-tolerant storage over multiple cloud storage providers, which is referred as FMSR method. Our FMSR code implementation preserves double fault-tolerance and has the same storage cost as in old-style erasure coding schemes based on RAID-6 codes, but uses fewer repair traffic when recovering a single-cloud failure and somewhat more for recovering a multi cloud failure.

2. PROPOSED METHODOLOGY

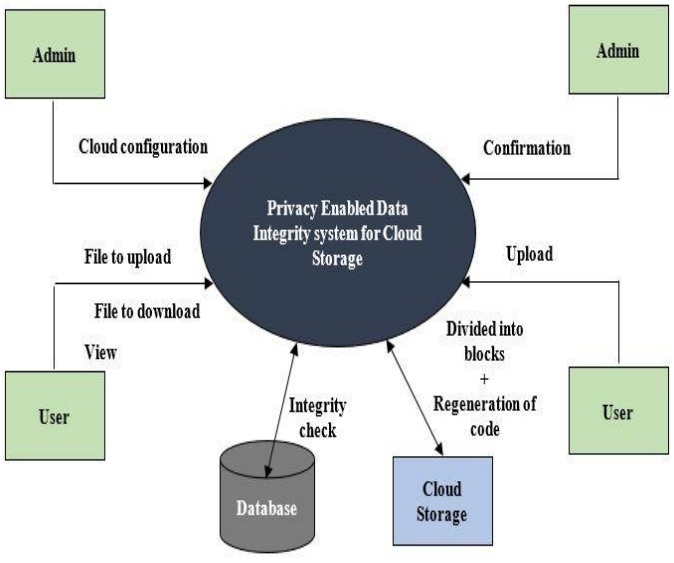

Fig -1: Architecture of the proposed methodology

In system model think about 3 kinds of entity: the cloud service providers, the admin and cloud users.

The Admin will have control over users, he has access over cloud storage server configuration. If he finds any dishonest activities of users then he has the power to delete the user. User has the facility to upload and download the file from the cloud. The database is used to store Admin data, User data, User transactions, MAC address of each file block for integrity test and store cloud storage details. When user upload the file, the file will divide into 4 native blocks and we create another 4 blocks which is in linear combination with native 4 blocks. So, there are totally 8 blocks. We store 2 blocks in each of the cloud storage

2.1 File Encryption Process

User has to login to upload the File, when user wants to upload data file to the cloud storage he has to select the file from his storage. When file is uploading to the cloud, we are generating the digital signature of the file and keep a copy of the digital signature in the user's database storage. The File content will be encrypted with users secrete key using the AES Algorithm. The Encrypted file will be send to the cloud storage by connecting through the file transfer protocol (ftp). once the connection is establishing with the cloud storage, encrypted file will be transferred to cloud storage.

Fig -2: Encrypted file stored on cloud

2.2 Data Integrity Check using Hashing Technique

When user is uploading the File to cloud, first the File will be read in byte stream for generating the MD5 key using Hashing Technique. The MD5 Key will be contains 16 bits and MD5 key will be generating based on the content of the uploading File. This MD5 key will be stored in user's database Server.

2.3 File Splitting Process

When User is uploading the file to cloud, File will be Transfer to server and the server will divide the file into four equal blocks Name the blocks as A, B, C & D and Create another four blocks as below

- A (XOR) C
- B (XOR) D
- A (XOR) D
- (B (XOR) D) (XOR) C

Now there are eight blocks, generate the MAC for all the eight blocks and Store the MACs in database.

2.4 Cloud Storage Server Process

To demonstrate the regenerating code based in cloud storage, we are keeping the four-cloud storage server. In each cloud storage two block of file will be stored. Thus all 8 blocks will be stored in the four different cloud storage servers as shown in below table 1

Cloud 1	Cloud 2	Cloud 3	Cloud 4
A	C	A⊕C	A⊕D
B	D	B⊕D	(B⊕D)⊕C

Table 1: Storage of files in different cloud servers

2.4 Regenerating code – Based Process

When the User wants to download the file from the cloud. First it will check for the status of each cloud like Checking for cloud storage 1 & 2 Status, if it is active

- Download all the four blocks
- Generate the MAC
- Retrieve the MAC from table
- Compare the MAC
- Display the Result
- If result is PASS then merge the blocks and form a File
- Download the File to the local system

If it is not Active, it will do the regenerating code for retrieving the file from the active cloud storage server as shown in table 2. Each Block division in the all cloud like (T-Cloud Active Status, F – Cloud Failure)

Cloud 1	Cloud 2	Cloud 3	Cloud 4	A	B	C	D
T	F	T	F	A	B	(A⊕C)⊕A	(B⊕D)⊕B
F	T	T	F	(A⊕C)⊕C	(B⊕D)⊕D	C	D
T	F	F	T	A	B	(B⊕D)⊕((B⊕D)⊕C)	(A⊕D)⊕A
F	F	T	T	(A⊕C)⊕C	(B⊕D)⊕D	(B⊕D)⊕((B⊕D)⊕C)	(A⊕D)⊕A

Table 2: FMSR Codes.

3. RESULTS

The fault tolerant implementation of FMSR codes as shown in Figure 3. Here we have a tendency to divide the file into

four native chunks, and construct eight distinct code chunks shaped by completely different linear combination of the native chunks. every native chunk has constant size M/4 as native chunks. Any 2 nodes are used to recover the original four native chunks. If Server 1 and server 4 are down, the proxy collects one code chunks from each surviving node, thus it downloads two code chunks is size M=4 every. Then the proxy generates 2 combinations of the 2 code chunks and put it in the new node.

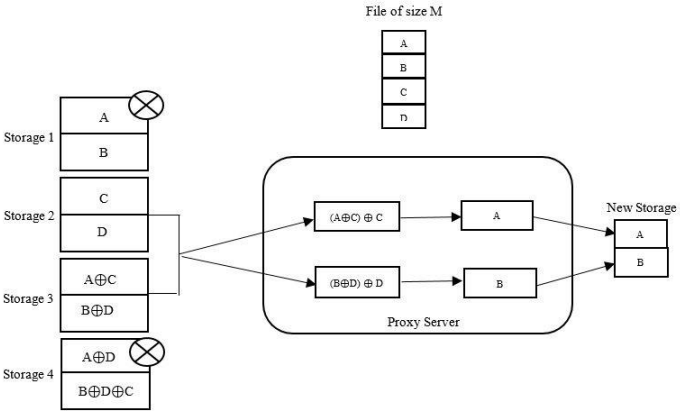

Fig -3: Re-generation of code using FMSR

4. CONCLUSION

In this paper, we introduce privacy enabled data integrity system for re-generating-coding-based cloud storage, where the data in the cloud are encrypted and ensures the data coming from the cloud are not modified. We also proposed multi-node failure recovery using network coding(FMSR) method. This paper not only provides fault tolerance in storage, but also allows cost-effective repair when a cloud permanently fails. We used a proxy server which uses the XOR implementation to regenerate the lost data using other surviving nodes. In cloud storage, requirement is to provide security to data stored on cloud. Our system uses AES algorithm which helps to encrypt the file on cloud. Therefore, data will be secured on cloud storage. The cloud user can check integrity of their data stored on cloud server using Hash technique. Our system uses MD5 algorithm which helps to create a key based on the file content. If any file is modified in the cloud then user get the message that the file is corrupted.

REFERENCES

[1] M. Armbrust et al., "Above the clouds: A Berkeley view of cloud computing", Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009

[2] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," Proc. ACM First ACM Symp. Cloud Computing (SoCC10), 2016.

[3] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS09), 2009

[4] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in Proc. ACM Workshop Cloud Comput. Secur. Workshop, 2010, pp. 31–42.

[5] Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage," in IEEE Trans. on information forensics and security, vol.. 10,no. 7, July 2015

[6] NCCloud:" A network coding based storage system in a cloud of c clouds", Henry C. H. Chen, Yuchong Hu, Patrick P. C. Lee, and Yang Tang, Jan 2014.

[7] K. Bowers, A. Juels, and A. Oprea. HAIL: A High-Availability and Integrity Layer for Cloud Storage. In Proc. of ACM CCS, 2009

[8] H. C. H. Chen and P. P. C. Lee, "Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 407–416.

BIOGRAPHIES

Dinesh Hiregange
M.Tech, Computer science and Information Technology
Dayananda Sagar University, Bengaluru, India

Dr. Chandramohan D
Assistant Professor, Department of Computer science and Engineering, Dayananda Sagar University, Bengaluru, India