Assuring Secured & Dependable Cloud Storage Services with Erasure Code Technique

Ravi Bhushan[1], Raj Singh yadav[2], M.Sriram[3]
1Student, Department of CSE, Bharath University, Chennai, *bhushanravi261@gmail.com*
2Student, Department of CSE, Bharath University, Chennai, *raajsyadav@gmail.com*
3Professor, Department of CSE, Bharath University, Chennai,*msr1sriram@gmail.com*

***Abstract*—** Cloud Computing Means a type of internet based Computing. Cloud Computing used a network of remote server hosted on the internet to store and manage data rather than a Local server or a own computer.Cloud storage that refers to online space that you can use to store data. It provides a secure way of remotely storing your important data.Cloud storage is gaining popularity due to its elasticity and low maintenance cost. In case data have been accidently corrupted or misplaced. In previously day they used Proof Of Retrivebility (POR) and Proof Of Data Possession (PDP) for repair the corrupted data and restore the unique data. But it is putting all data on the on its own server. MRPDP and HAIL method are used for regenerating code not reading and reconstructing the whole file. FMSR-DIP codes preserve fault tolerance and repair traffic saving. We are Implementing Erasure code to reconstruct the data. Erasure code is a method of data protection in which data is encrypted, splited and store up in the different server.

Keywords- *Remote data checking, secure and trusted storage system*

*****

I. INTRODUCTION

Cloud Computing Means a type of internet based Computing. Cloud Computing used a network of remote server hosted on the internet to store, manage and process data rather than a Local server or a Personal computer. Cloud storage that refers to online space that you can use to store data. It provides a secure way of remotely storing your important data. Cloud storage is gaining popularity due to its elasticity and low maintenance cost. If we detect corruption in our outsourced data (when a server crashes or is compromised)then we should repair the corrupted data and restore the original data. Proof Of Retrivebility (POR)[and Proof Of Data Possession (PDP) has been proposed to verify the integrity of a large file by spot checking[4]. Proof Of Retrivebility (POR)[4] and Proof Of Data Possession (PDP) for repair the corrupted data and restore the original data. But it is putting all data on the single server. MRPDP and HAIL method has to minimize repair traffic[2]. Hail is a remote file integrity checking protocol that offer effectiveness, protection, and modeling improvements [2][3].MRPDP and HAIL not reading and reconstructing the whole file during repair[2][1]. Functional Minimum-Storage Regenerating-Data Integrity Protection (FMSR-DIP) codes for allow clients to remotely verify the integrity of random subsets of long term archival data under multi server setting [1]. FMSR-DIP codes perform basic file operations Upload, Download, Check and restore[1] for 1. interpret data from the other servers, 2. rebuild the ruined data, and 3. Write the reconstruct data to a novel server. FMSR-DIP codes preserve fault tolerance and repair traffic saving[1].In this paper, we design and implement a erasure code for reading and reconstructing the data. in the erasure code Once the data was uploaded into the cloud server, the Cloud server will split the data into many parts and store all the data in the separate data servers. This techniques wasn't used in proposed system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in corresponding data server. We're also encrypting the data segments before storing into the data server. This encrypted data are converted into bytes and added parity bit process by the data owner in order to restrict TPA by accessing the original data.

The Cloud server generates the voucher number from the parity added encrypted data and compared with the signature provided to the TPA to verify the Data Integrity. We design Erasure code, which enable integrity protection, fault tolerance and security for cloud-storage.

We export tunable parameters form Erasure code, such that clients can make a trade-off between performance and security. we perform mathematical breakdown on the security of Erasure code for different parameter choice.

II. RELATED WORK

We briefly summarize the most recent and closely connected work. We consider the trouble of examination the integrity of fixed data, which is classic in long-term archival storage systems. This trouble is first measured under a single server.giving rise to the like design POR and PDP, respectively. A main restriction of the above schemes is that they are designed for a single-server. If the server is fully controlled, then the above schemes can only provide detection of corrupted data, but cannot recuperate the unique data. This lead to the propose of resourceful data checking schemes in a multi server. By striping unnecessary data across multiple

 ISSN: 2321-8169
1476 - 1479

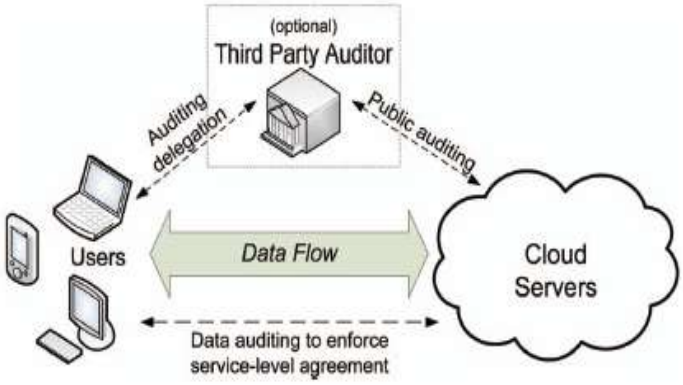
servers, the original files can still be recovered from a subset of servers even if some servers are down. Efficient data integrity checking has been proposed for different redundancy schemes, such as replication, and regenerate coding also. consider regenerating -coded storage space, there are key in difference with our effort. First, their drawing extend the single-server dense POR. RAID-like techniques used by disks and file system, other than at the cloud storage space level. We dispute that striping client data across many providers can allow customers to avoid retailer lock-in, decrease the rate of switch provider, and better bear source outages or failure.RACS can reduce the cost of switching storage vendors for a large organization such as the Internet Archive by seven-fold or more by varying erasure-code parameter. HAIL (High-Availability and Integrity Layer), a spread cryptographic system that permits a set of servers to prove to a client that a stored file is intact and retrievable. The most closely Related work to Functional Minimum-Storage Regenerating-Data Integrity Protection (FMSR-DIP) codes for allow clients to remotely verify the integrity of random subsets of long term archival data under multi server situation. FMSR-DIP codes execute basic file operation Upload, Download.

III. PROPOSED WORK

we design and implement a erasure code for reading and reconstructing the data. in the erasure code Once the data was uploaded into the cloud server, the Cloud server will split the data into many parts and store all the data in the separate data servers. This techniques wasn't used in proposed system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in corresponding data server. We're also encrypting the data segments before storing into the data server. This encrypted data are converted into bytes and added parity bit process by the data owner in order to restrict TPA by accessing the original data.

IV. EXPERMENTAL AND RESULT

In this paper we have used the various algorithm to repair and reconstructed the data on the cloud. We have used windows 8 operating system to accomplish the project, NETBEANS ide7.01 has been used as an included development environment. We use NETBEANS as frontage end and SQL server 2008 as rear end. The language used for coding is JAVA and the processor to execute the project should be minimum with Pentium Dual Core 2.00GHZ and hard disk of minimum 500GB.



V. MODULE DECRIPTION

5.1 Data Owner: Data Owner is the Person who is going to upload the data in the Cloud Server. To upload the data keen on the Cloud server, the Data Owner have be registered in the Cloud Server. Once the Data holder register in cloud server, the space will be allotted to the Data Owner.

5.2DATA SPLITTING AND ENCRYPTION**:** In this module, once the data was uploaded into the cloud server, the Cloud server will split the data into many parts and store all the data in the separate data servers. In techniques wasn't used in proposed system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in equivalent data server. We're also encrypt the data segment before storing into the data server.

5.3PARTY BIT ADDITION AND ERASURE CODE: Once the data are stored in the equivalent data servers and the keys are stored in the key servers. Then we're addition of parity bits to the data, so that the data will be changed. Also we're applying the Erasure Code by using the XOR operation, while XORing the block data, the data will be converted in binary data.

5.4TRUSTED PARTY AUDITOR**:** Once added the parity added bits, then the data will be given to the Trusted Parity auditor. The Trusted Parity Auditor will generate the signature using change and response method. The data will be audited in this module, if any changes occurs it will provide the intimation regarding the changes.

5.5 REPLICA SERVER:

We'll maintain the separate Replica Cloud server. If suppose the data in the data server was misplaced, then the Main Cloud server will make contact with the duplication Cloud server and get the data from the duplication Cloud Server. By using this notion, we can obtain the data if any data defeat occur.

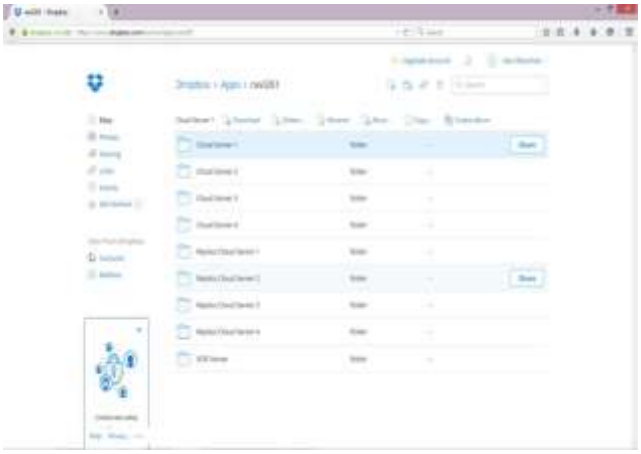
VI. IMPLEMENTATION AND WORKING:

 ISSN: 2321-8169
1476 - 1479

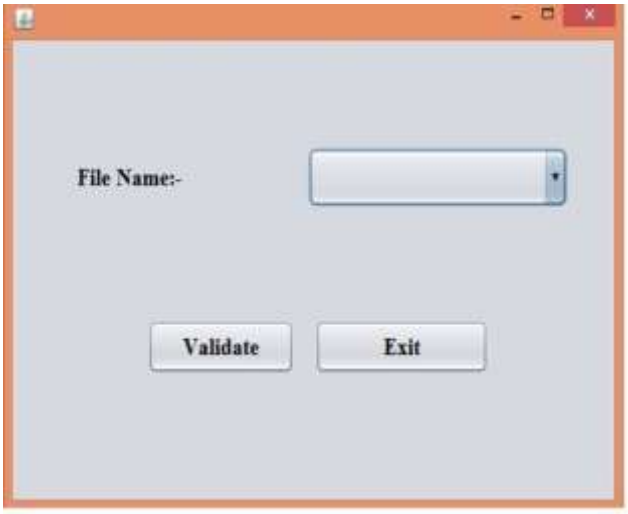
Here, in Fig(1) the user has to first register the account.

In Fig(2) the registered user has to login in there account.

Fig(3) the registered user will upload the file to the cloud server.



Fig(4) here the uploaded files are splited and stored into the different cloud server.



Fig(5) The files automatically gets validated before download.

Fig(6) Here the registered user downloads the files from the cloud server.

VII. CONCULISON

The trouble of data protection in cloud data storage, which is basically a spread storage system, is discussed in this work. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage facility for user, we recommend an efficient and elastic spread scheme with explicit dynamic data maintain, with block update, remove, and attach. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilize the homomorphism sign with spread authentication of erasure coded data, our plan achieves the incorporation of storage correctness indemnity and data fault localization, i.e., whenever data corruption has been detect through the storage accuracy authentication across the spread servers, we can almost assurance the immediate detection of the mischievous server(s). Allowing for the time, calculation resources, and even the connected online weight of user, we also provide the addition of the proposed main scheme to support intermediary audit, where users can securely hand over the reliability examination tasks to intermediary auditor and be undisturbed to use the cloud storage services. Through full safety and broad experiment results, we show that our system is extremely efficient and flexible to complicated failure, spiteful data adaption assault, and even server collude attack.

REFERENCES

[1] Henry C.H. Chen and Patrick P.C. Lee "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage: Theory and Implementation", VOL. 25, NO. 2, FEBRUARY 2014

[2] K. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.

[3] R. Curtmola, O. Khan, and R. Burns, "Robust Remote Data Checking," Proc. ACM Fourth Int'l Workshop Storage Security and Survivability (StorageSS '08), 2008

[4] K. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Proc. ACM Workshop Cloud Computing Security (CCSW '09), 2009.

[5] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," Proc. First ACM Symp.Cloud Computing (SoCC '10), 2010.

[6] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp 50-58, 2010.

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote Data Checking Using Provable Data Possession," ACM Trans. Information and System Security, vol. 14, article 12, May 2011.

[8] L. Chen, "NIST Special Publication 800-108," Recommendation for Key Derivation Using Pseudorandom Functions (Revised), http:// csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf, Oct. 2009.

[9] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop Cloud Computing Security (CCSW '10), 2010.