DATA INTEGRITY PROTECTION AND SECURITY IN CLOUD BASED STORAGE

K.BRINDHA[1]
PG Scholar
Computer science and Engineering
Dr. N.G.P Institute of Engineering
Coimbatore, Tamil Nadu, India
brindha8892@gmail.com

P.ANITHA[2]
Assistant Professor
Computer Science and Engineering
Dr. N. G. P. Institute of Technology,
Coimbatore, Tamil Nadu, India
anitha@drngpit.ac.in

C.VINOTHINI[3]
Assistant Professor
Computer Science and Engineering
Dr. N. G. P. Institute of Technology,
Coimbatore, Tamil Nadu, India
vinothini@drngpit.ac.in

Abstract— **Cloud storage is widely used by the people and organization, were they buy or lease storage capacity to store large amount of data. On-demand data outsourcing is offered in cloud storage, but it is critical to protect against data corruption and provide data integrity and availability. Security problem arises when data storage is outsourced to third party cloud storage. We implement a schema called Data Integrity Protection (DIP) were data are striped and they are encrypted and sent across multiple servers. This provides both integrity and security for data which are stored for long term. Client can verify the integrity of random subsets of outsourced data against corruption and availability of data at low cost. Remote modification of data can be done without downloading the file. We make a trade-off between performance and security which is very important for maintaining efficient cloud storage.**

Keywords- ***remote data checking, data striping, secure cloud storage, integrity protection***

I. INTRODUCTION

Cloud computing provides the capabilities of storing and processing the data in third party data center. The concept of cloud computing is coverage infrastructure and shared services. Security is one of the main problem faced in cloud storage. So protecting the data against corruption and enhancing fault tolerance is essential in cloud storage. Data loss, data modification, loss of control and lack of trust should be prevented for efficient cloud storage.

Cloud storage is one where the digital data are stored in a logical pool and keeps the data available, accessible, reliability all the time. It provides on-demand data outsourcing services with low maintenance cost. Here we no need to purchase, manage and maintain expensive hardware.

Cloud storage provides long-term archival where the data are read and written rarely. The data which is stored should be the same when it is retrieved. So integrity checking of the data is done to avoid deleted and modified data before retrieval. So, remote data integrity checking can be done to make this process more efficient.

In cloud storage there are huge amount of data that has to be stored and retrieved. So two concepts that are introduced are Proof of Retrievability (POR) [16] and Proof of data Possession (PDP) [3] . This is proposed to spot check the huge amount of data. POR [16] and PDP [3] are proposed only in single server.MR-PDP[10] and HAIL[4] is proposed to check the integrity for multiple server using replication coding.

Regenerating codes [11] is introduced to minimize repair traffic. Minimizing the repair traffic was done by reading the whole file and reconstructing it. But this was a very difficult process to follow. To make this simple a set of chunks smaller than the original file is taken from another surviving server and reconstructs only the lost data chunks. In simple words when any unreliable nodes are found, a new node should be created.

Regenerating codes allow a new node to communicate functions of the stored data from the surviving nodes. It reduces the repair bandwidth. There is a fundamental tradeoff between storage and repair bandwidth. It can be achieved at any point in this optimal tradeoff. We can enable integrity check with regenerating codes by preserving repair traffic by HAIL [4] concept. It protects data on per file basis and distributes across different servers. To repair lost data in a failed server one has to access the whole file. This does not support the regenerating code [11] concepts. Thus we need to design integrity protection with regenerating code.

Thus Data Integrity Protection (DIP) for regenerating codes was introduced. We also implement Functional Minimum-Storage Regenerating (FMSR) codes and finally FMSR-DIP codes is constructed. FMSR-DIP codes are to randomly check the data in a multiserver settings. This is constructed for efficient data integrity and security simultaneously. It targets on long-term archives.

Due to security issues in cloud storage regeneration codes are not helpful. So the data which are stored in server are in encrypted form. When there is any loss of data the server backup will automatically heal the lost data. Only thin cloud interface [27] is only assumed where only read/write functionalities are supported. The contributions of FMSR-DIP are it provides integrity protection, fault tolerance and efficient recovery for cloud based storage.

The paper has the following sections. Section 2 describes about the related works. Section 3 describes about the system model for the proposed system. Section 4 describes the experimental results and section 5 describes conclusion and future work of the paper.

II. RELATED WORKS

We consider the problem of integrity security of static data, which is in long term archival storage systems. The major limitation found in POR [16] and PDP [3] is that they are designed only for single server setting. It detects only the corrupted data and cannot recover the original data. Efficient data integrity checking is has been proposed for different redundancy schemes, such as replication [10], erasure coding [4], and regenerating coding [6]. Compact Proof of Retrievability [25] is a data storage center prove to the verifier that all the data stored are the client’s data only. Here both efficient and provably secure storage system provided. Cumulus [27] is a efficient filesystem backup system which is specifically designed for thin cloud. It provides only least-common-denominator which is get and put of complete files.

A proxy-based storage system for fault-tolerant multiple-cloud storage called NC Cloud is introduced, which achieves cost-effective repair for a permanent single-cloud failure.

Functional Minimum-Storage Regenerating (FMSR) codes are introduced which maintain the same fault tolerance and data redundancy as in traditional erasure codes.

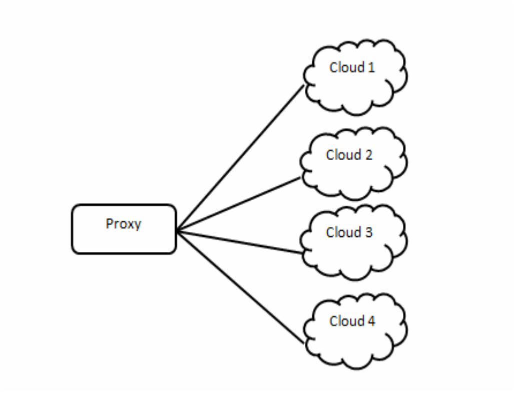

Figure 1. Normal operation

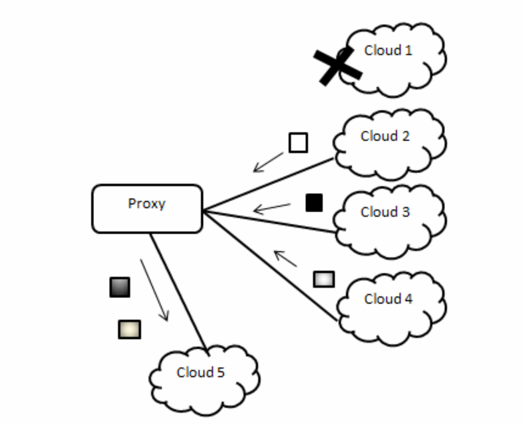

Figure 2. Repair operation

This is a Proxy-based design for multiple-cloud storage. Fig 1 is the normal operation, and Fig 2 is repair operation. When the cloud node 1 fails the proxy regenerates the data to new cloud 5.

The work mainly focuses on how to maintain both integrity and security of private data in a multiserver settings.

III. SYSTEM MODEL

A cloud based storage system is proposed which is more efficient compared to other storage system. Here the data which has to be stored is striped and the striped data is encrypted using AES algorithm. Those encrypted data are divided into chunks and are sent across multiple servers.

Security is maintained efficiently because the encrypted data are stored in different servers. It is difficult to know in which pattern the data are striped and encrypted sent to different servers. DIP is nothing but it checks that the data which is stored and retrieved are the same. Here remote data integrity checking is done. Data loss, data modification by third party are avoided. The user can store the data and retrieve the data quickly and efficiently. All this process is executed in thin cloud.

The file is uploaded and that file is stripped. That represents the FMSR code chunks. That file is then encrypted and that is the FMSR-DIP code chunks. Those chunks is then sent to a storage interface and are distributed to multiple different servers. Fig 3 gives the System Architecture of the above process.

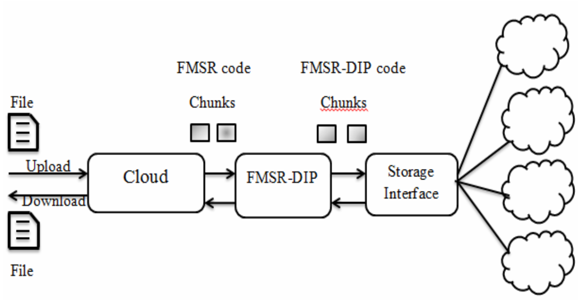

Figure 3. System Architecture

We use thin cloud storage where each server only provides a basic interface for clients to read and write stored files. Now a days cloud storage providers provide a interface called RESTful interface which has only PUT and GET commands. PUT is used to write the file and GET allows to read the file. DIP uses PUT and GET commands to interact with servers.

We also use a Adversarial Error-Correcting Code (AECC) [5],[9] to protect the chunks against corruption. In conventional Error-Correcting Code (ECC) the large encrypted file is broken into smaller strips and then ECC is applied. But AECC uses Pseudorandom Functions (PRFs) [13][14] which makes infeasible for target to corrupt any particular striped data. FMSR codes and AECC provides fault tolerance. The main difference between FMSR codes and AECC is FMSR codes is applied for striped data which is stored in server and AECC is applied for single chunk stored in a server. Both are used to improve the integrity and security.

IV. Experimental Result

File which are stored and retrieved are performed in private cloud to enhance security. The operations performed are Data Upload, Data Modification, Data Retrieval. In the Data Upload process the data are striped and encrypted using AES algorithm. Those data are divided into chunks and are stored in different servers. Only the encrypted data are stored so this maintains the integrity of the data. The secret keys can be securely stored by the client without being reveled to the servers.

In Data Retrieval process the name of the file is given and the file is retrieved. Data Modification is a process where at the time of retrieval if the file has to be modified the modification is done and then the file is uploaded. Remote modification of text file can be done.

If the data are corrupted that can be recovered by AECC. The AECC first checks which part of the data is corrupted. The it checks the backup server and heals the encrypted data lost in server. So automatic healing of data loss and corruption is provided to increase the availability of data in cloud storage.

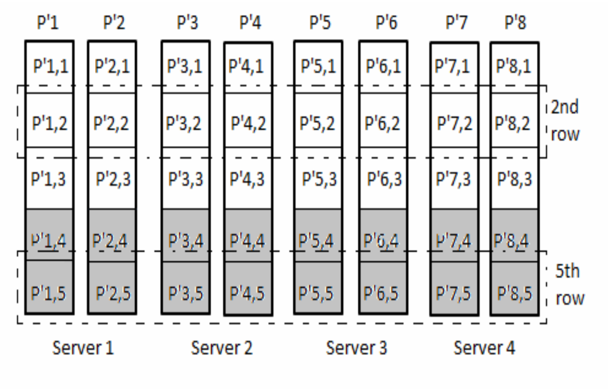

Figure 4. Server Setup.

The encrypted data are stored in P'i,1 , P'i,2 and P'i,3. This is how the data are divided and are stored as chunks. P'i,4 and P'i,5 corrosponds to the AECC parities of the chunks. When the data are lost or corrupted the the AECC goes and checks with the backup data and recover the data.

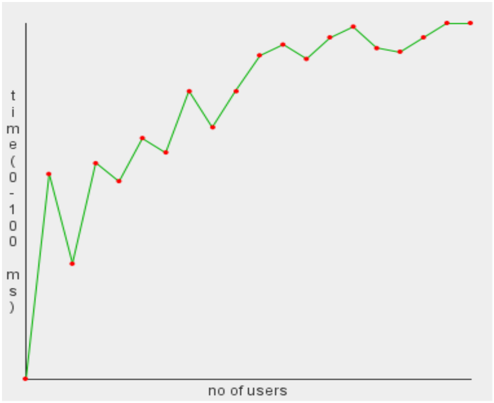

Figure 5. Time taken to upload and retrive the data

When the number of users increases then the time taken to upload and retrive the data efficiency also increases.

V. Conclusion and Future Work

In this work, we have discussed how security, integrity and availability of data is maintained highly and efficiently in cloud storage. To achieve this we have designed FMSR-DIP codes which shows a trade between performance and security. Data loss and data corruption is managed using AECC, adding fault tolerance to the cloud based storage. The stripping and encryption of the data increases the security by 3% compared to the existing system. We have implemented only for text documents.

In future work, security of data is can be improved by introducing shuffling algorithm. So the data could be stripped and those data are encrypted. Then the encrypted data are shuffled in a particular pattern and are divided into chunks. The chunks are stored across multiple servers. This increase the security of private and confidential data. We can implement this method for video files where each individual frame is taken and are shuffled and are stored in multiple server.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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).

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

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

[7] H.C.H. Chen and P.P.C. Lee, "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage," Proc. IEEE 31st Symp. Reliable Distributed Systems (SRDS '12), 2012.

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

[10] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. IEEE 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), 2008.

[11] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Sys- tems," IEEE Trans. Information Theory, vol. 56, no. 9, 4539-4551, Sept. 2010.

[12] D. Ford, F. Labelle, F.I. Popovici, M. Stokel, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in Globally Distributed Storage Systems," Proc. Ninth USENIX Symp. Operating Systems Design and Implementation (OSDI '10), Oct. 2010.

[13] O. Goldreich, Foundations of Cryptography: Basic Tools. Cambridge Univ. Press, 2001.

[14] O. Goldreich, Foundations of Cryptography: Basic Applications. Cambridge Univ. Press, 2004.

[15] Y. Hu, H. Chen, P. Lee, and Y. Tang, "NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds," Proc. 10th USENIX Conf. File and Storage Technologies (FAST '12), 2012.

[16] A. Juels and B. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), 2007.

[17] H. Krawczyk, "Cryptographic Extraction and Key Derivation: The HKDF Scheme," Proc. 30th Ann. Conf. Advances in Cryptology (CRYPTO '10), 2010.

[18] E. Naone, "Are We Safeguarding Social Data?" http:// www.technologyreview.com/blog/editors/22924/, Feb. 2009.

[19] J.S. Plank, "A Tutorial on Reed-Solomon Coding for Fault- Tolerance in RAID-Like Systems," Software - Practice & Experience, vol. 27, no. 9, pp. 995-1012, Sept. 1997.

[20] M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," J. ACM, vol. 36, no. 2, pp. 335- 348, Apr. 1989.

[21] I. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," J. Soc. Industrial and Applied Math., vol. 8, no. 2, pp. 300- 304, 1960.

[22] B. Schroeder, S. Damouras, and P. Gill, "Understanding Latent Sector Errors and How to Protect against Them," Proc. USENIX Conf. File and Storage Technologies (FAST '10), Feb. 2010.

[23] B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proc. Fifth USENIX Conf. File and Storage Technologies (FAST '07), Feb. 2007.

[24] T. Schwarz and E. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Sto- rage," Proc. IEEE 26th Int'l Conf. Distributed Computing Systems, (ICDCS '06), 2006.

[25] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), 2008.

[26] "TechCrunch," Online Backup Company Carbonite Loses Customers' Data, Blames and Sues Suppliers, http://techcrunch.com/2009/03/ 23/online-backup-company-carbonite-loses-customers-data- blames-and-sues-suppliers/, Mar. 2009.

[27] M. Vrable, S. Savage, and G. Voelker, "Cumulus: Filesystem Backup to the Cloud," Proc. USENIX Conf. File and Storage Technologies (FAST), 2009.

[28] "Watson Hall Ltd," UK Data Retention Requirements, https:// www.watsonhall.com/resources/downloads/paper-uk-data- retention-requirements.pdf, 2009.

[29] A. Wildani, T.J.E. Schwarz, E.L. Miller, and D.D. Long, "Protecting Against Rare Event Failures in Archival Systems," Proc. IEEE Int'l Symp. Modeling, Analysis and Simulation Computer and Telecomm. Systems (MASCOTS '09), 2009.