ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
DIIF 0.876

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 612-616

Survey on NCCloud for Distributed Cloud Storage

[1] Venkat Sujan, [2]Dr.S.Prem Kumar
[1](M.Tech), CSE
[2]Professor & HOD, Department of computer science and engineering,
G.Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.

Abstract:- Late advances have offered ascend to the popularity and achievement of cloud registering. Cloud database empowers clients to remotely store their data and appreciate the on-interest fantastic cloud applications without the weight of neighborhood equipment and programming administration. It moves the application programming and databases to the unified expansive data focuses, where the data's administration and administrations may not be completely dependable. This one of a kind worldview realizes numerous new security challenges. To ensure outsourced data in cloud database against defilements, empowering integrity security, adaptation to internal failure and effective recuperation for cloud database gets to be basic. Consequently, we concentrate on the issue of remotely checking the integrity of recovering coded data against debasements under a genuine cloud database setting.

Keywords: cloud computing, remote data checking, security, integrity, regenerating code, FMSR-DIP

I.INTRODUCTION

Cloud computing characterized as "A large-scale distributed computing worldview that is driven by economies of scale, in which a pool of disconnected, virtualized, powerfully, adaptable, oversaw computing force, stockpiling, stages, and administrations are conveyed on interest to outer clients over theInternet." Cloud stockpiling offers an on-interest data outsourcing administration display, and is picking up prominence because of its versatility and low support cost. One significant utilization of cloud stockpiling is long haul archival, which speaks to a workload that is composed once and once in a while read. While the put away data is once in a while read, it stays important to guarantee its integrity for catastrophe recuperation or consistence with legitimate prerequisites. The ideas of integrity in cloud computing concerns are both data integrity and calculation integrity. Data integrity infers that data ought to be genuinely put away on cloud servers, and any infringement (e.g., data is lost, modified, or bargained) are to be identified. Calculation integrity infers the idea that projects areexecuted without being bended by malware, cloud suppliers, or different malevolent clients, and that any wrong computing will be identified. So it is attractive to empower cloud customers to confirm the integrity of their outsourced data in the cloud, on the off chance that their data has been inadvertently defiled or vindictively bargained by insider/untouchable Byzantine assaults. To meet the prerequisites of the gigantic volume of capacity, deletion codes have picked up a lot of consideration in cloud frameworks.

II. RELATED WORK

Security in Cloud Computing:

The popularity of Cloud Computing is mainly due to the fact that many enterprise applications and data are moving into cloud platforms; however, lack of security is the major barrier for cloud adoption [1]. According to a recent survey by International Data Corporation (IDC), 87.5% of the masses belonging to varied levels starting from IT executives to CEOs have said that security is the top most challenge to be dealt with in every cloud service. Many of the threats found in existing platforms. Out of them, the Security Threat is considered to be of High Risk. The major security aspect is Confidentiality, Integrity, Authentication, Authorization, Nonrepudiation and Availability which are further explained below: **Confidentiality** is the process of making sure that the data remains private, confidential and restrictedfrom unauthorized users [2]. Data encryption is one of the most popular options of security before pushing the Data into cloud. **Integrity** is the guarantee by which the data is protected from accidental or deliberate (malicious) modification. Hashing techniques, digital signa-

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 612-616

tures and message authentication codes are used to preserve data integrity [3]. Integrity problems are in big scale due to the multi-tenancy characteristic of cloud [4]. **Authentication** is the mechanism by which the systems may securely identify their users. Authorizationdetermines the level of access to system resources attributed to a particular authenticated user [5]. **Non-repudiation** is an extension to the identification and authentication service. It is used to ensure thatthe messages sent are properly received and acknowledgements are sent back to the sender. In other words, establishing a two way communication between a sender and a receiver. Availability ensures that an organization has its full set of computing resources available and usable at all times for its real users [8]. In this paper we will discuss about the integrity of data in cloud storage.

Data integrity proving schemes

Juels and Kaliski [8]. Proposed a scheme called Proof of Retrievability (POR). Proof of retrievability means verify the data stored by user at remote storage in the cloud is not modified by the cloud. POR for huge size of files named as sentinels. The main role of sentinels is cloud needs to access only a small portion of the file (F) instead of accessing entire file. Sravan and Saxena [7].Proposed a Schematic view of a proof of retrievability based on inserting random sentinels in the data file. **Provable Data Possession (PDP) Definition:** A PDP scheme checks that a file, which consists of a collection of n blocks, is retained by a remote cloud server. The data owner processes the information file to generate some metadata to store it locally. The file is then sent to the server, and the owner deletes the native copy of the file. The owner verifies the possession of file in using challenge response protocol. This technique is used by clients to check the integrity of the data and to periodically check their data that is stored on the cloud server. So this technique ensures server security to the client. **Native Method**: The main idea behind this algorithm is to compare the data. In this method client will compute the hash value for the file F and having key K (i.e. (K, F)) and subsequently it will send the file F to the server. Clients are having different collection of keys and hash values so it can check multiple check on the file F. Whenever client wants to check the file it release key K and sends it to the server, which is then asked to recomputed the hash value, based on F and K. Now server will reply back to the client with hash value for comparison. Limitation This method gives the strong proof that server is having the original file F.But this method has high overhead as every time hashing process is run over the entire file. It is having very high computation cost. **Proof of Retrivability (POR):** Juels and Kaliski [8]. Proposed a scheme called Proof of Retrievability. Proof of retrievability means Verify the data stored by user at remote storage in the cloud is not modified by the cloud. POR for huge size of files named as sentinels. The main role of sentinels is cloud needs to access only a small portion of the file (F) instead of. In this scheme data are divided into number of block as shown in figure 2. This technique uses theauditing protocol when solving the problem of integrity.

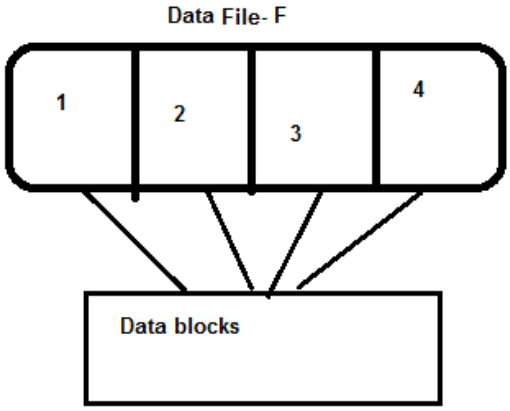

Fig1. A datafiles with 4 blocks.

III. SYSTEM STUDY

There are various study performed to check the integrity of data, which are typical in long-term archival storage systems. This problem is first considered by Juels et al. [8]. And Ateniese et al. [9]., giving rise to the similar notions proof of retrievability [8]. (POR) and proof of data possession (PDP) [9]., respectively, which are proposed to verify the integrity of a large file by spot-checking only a fraction of the file via various cryptographic primitives. The basic POR scheme [8]. Embeds a set of pseudorandom blocks into an encrypted file stored on the server, and the client can check if the server keeps the pseudorandom blocks later on. Error correcting codes are also included in the stored file to allow recovery of a small amount of errors within a file. How-

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 612-616

ever, the number of checks that the client can issue is limited by the number of the embedded random blocks. On the other hand, PDP [9]. Allows the client to keep a small amount of metadata. The client can then challenge the server against a set of random file blocks to see if the server returns the proofs that match the metadata on the client side. These both schemes are single server storage scheme. So in these methods whole the data are stored on a single server in which single-point-failure [11]. And vendor-lock-ins [10]. Problems are arises. To overcome these problems one possible solution is to stripe data across multiple servers. Thus, to repair a failed server, we can (i) read data from other surviving servers, (ii) reconstruct the corrupted data of the failed server, and (iii) write the reconstructed data to a new server. MR-PDP [13]. And HAIL [12]. Extend integrity checks to a multi-server setting using replication and erasure coding, respectively. In erasure coding based system (e.g. Reed Solomon Code) requires less storage overhead compare to Replication based system [14]. For the same faulttolerance level.

A. Replication Based system Ensuring reliability requires the introduction of redundancy. The simplest form of redundancy is replication, which is adopted in many practical storage systems. In which k identical copies of each data object are kept at each instant by system members. Figure 2 shows an example of replication based system.

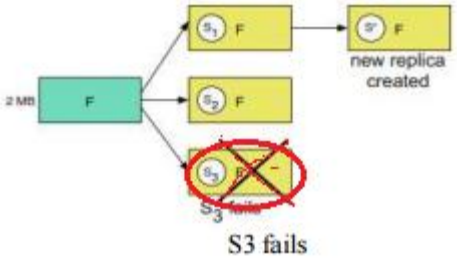

Fig. 2 Replication based distributed system

Simple replication offers one avenue to higherassurance data archiving. Only single copy of file is required to repair any node. For example if any node fails then simply copy the replica of that file from healthy node and store it on new node. But it requires often unnecessarily and unsustainably high expense. The storage cost for replication based system is very high.

B. Erasure Coding Based (Reed Solomon Code) system As a generalization of replication, erasure coding offers better storage efficiency. For instance, we can divide a file of size M into k pieces (to be called fragments), each of size M/k, encode them into n encoded fragments (of the same size) using an (n, k) maximum distance separable (MDS) code, and store them at n nodes. Then, the original file can be recovered from any set of k coded fragments. This performance is optimal in terms of the redundancy– reliability trade-off because k pieces, each of size M/k, provide the minimum data for recovering the file, which is of size M. Example of (4, 2) Erasure coding based system is shown in figure 2. In which repair traffic of the system if M which is same as our file size.

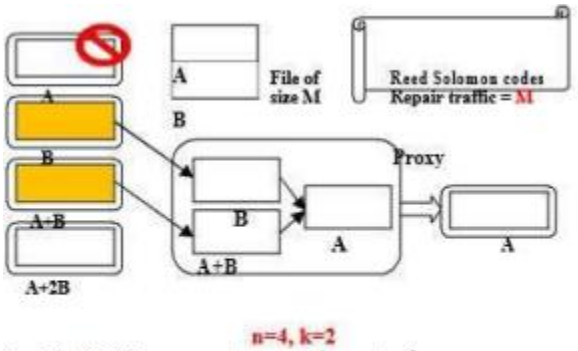

Fig. 3 Reed Solomon Erasure Coding

C. Regenerating Coding Based system For an erasure coded system, a common practice to repair from a single node failure is for a new node to reconstruct the whole encoded data object to generate just one encoded block. This is clearly an inefficient way of regeneration, since the network bandwidth is often a critical resource. This has motivated the development of family of codes, referred as regenerating codes, designed to carry out the regeneration efficiently. Regenerating codes [15]. Have been proposed to minimize this repair traffic (i.e., the amount of data being read from surviving servers). In essence, they achieve this by not reading and reconstructing the whole file during repair as in traditional erasure codes, but instead reading a set of chunks smaller than the original file from other surviving servers and reconstructing only the lost (or corrupted) data chunks. Regenerating codes are constructed systematically such that the source symbols are stored in k nodes called as data nodes, the remaining n-k nodes are called as parity nodes which contain symbols obtained through suitable encoding operation. Such systematic codes, where the data nodes are regenerated exactly but only functionally

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 612-616

equivalent form of parity nodes are regenerated. Example of (4, 2) Regenerating code is shown in figure 3. In which repair traffic is 0.75M which is less than the Reed Solomon Erasure coding based system.

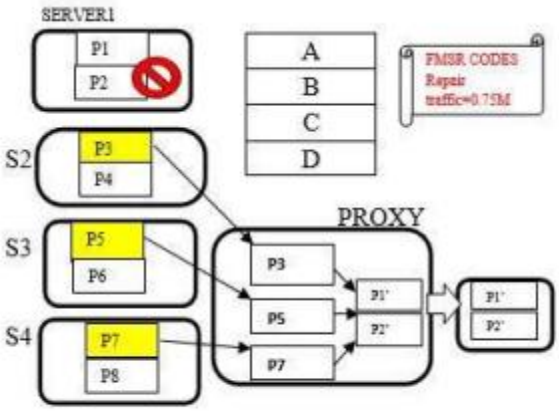

Fig. 4 Regenerating Code based system

IV.PROBLEMS IN EXISTING SYSTEM

As it is noted in the different techniques of checking integrity of cloud storage data in analysis part there are some drawbacks in the existing system, which are like not secure against byzantine mobile adversary. Mobile Byzantine means that the adversary compromises a subset of servers in different time epochs (i.e., mobile) and exhibits arbitrary behaviours on the data stored in the compromised servers (i.e., Byzantine). To ensure file availability, we assume that the adversary can compromise and corrupt data in at most n−k out of the n servers in any epoch, subject to the (n, k)- MDS fault tolerance requirement. At the end of each epoch, the client can ask for randomly chosen parts of remotely stored data and run a probabilistic checking protocol to verify the data integrity. Servers under the control of the adversary may or may not correctly return data requested by the client. If corruption is detected, then the client may trigger the repair phase to repair corrupted data. Instead of performing whole-file checking, which incurs a substantial transfer overhead; it is only feasible for the client to randomly sample data for integrity checking. The adversary may corrupt a small portion of data within the Error-correcting capability in each epoch, but the level of Corruption can render the errors unrecoverable after several Epochs if they are not spotted early. This leads to creeping Corruption [12]. Thus, it is necessary that the client can quickly spot the corrupted data without accessing the whole File.

V.CONCLUSIONS

Though Cloud computing offers great potential to improve productivity and reduces costs. It also imposes many new security risks which are related to cloud storage. As cloud is mainly used for the storage of the data, data integrity is the main issue of the client side because after uploading data to the server, client will lost the control of the data. There are so many techniques available in the literature, out of which we have analyze Provable Data Possession (PDP) and Proof of retrievability (POR), This paper facilitate the client in getting a proof of integrity of the data which He/She wishes to store in the cloud storage servers with bare minimum costs and efforts. The scheme used in this paper reduces the computational and storage overhead ofthe client as well as to minimize the computational overhead of the cloud storage server. This also minimized the size of the proof of data Integrity so as to reduce the network bandwidth consumption. Seeing thepopularity of outsourcing archival storage to the cloud, it is desirable to enable clients to verify the integrity of their data in the cloud. We study design of data integrity protection (DIP) scheme for functional minimum storage regenerating (FMSR) codes under a multi-server setting. This DIP scheme preserves the fault tolerance and repair traffic saving properties of FMSR. And also it allows clients to remotely verify the integrity of random subsets of long term

REFERENCES

1. Bansidhar Joshi, A. Santhana Vijayan, Bineet Kumar-Joshi,"Securing Cloud computing Environment against DDoS

2. Ramgovind S, Eloff MM and Smith E, "The Management of Security in Cloud Computing", IEEE, 2010.

3. Wentao Liu, "Research on Cloud Computing Security Problem and Strategy", IEEE, 2012, pp.1216-1219.

4. Nitin Singh Chauhan and Ashutosh Saxena. "Energy Analysis of Security for Cloud Application.

5. Rohit Bhadauria, Rituparna Chaki, Nabendu Chaki and Sugata Sanyal, "A Survey on Security Issues in Cloud Computing", IEEE 2010.

6. Xiaojun Yu and Qiaoyan Wen "A view about cloud data security from data life cycle", IEEE 2010.

7. Saxena, Sravan Kumar and Ashutosh,"Data Integrity Proofs in Cloud Storages", IEEE 2011.

8. A. Juels and B. Kaliski. PORs: Proofs of retrievability for large files. In Proc. ACM CCS, pages 584–597, 2007.
9. 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. on Information and System Security, May 2011.
10. H.Abu-Libdeh, L. rincehouse, and H. Weatherspoon. RACS: A Case for Cloud Storage Diversity. In Proc. of ACM SoCC, 2010.
11. 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. Communications of the ACM, 53(4):50– 58, 2010.
12. K. Bowers, A. Juels, and A. Oprea. HAIL: A High-Availability and Integrity Layer for Cloud Storage. In Proc. of ACM CCS, 2009.
13. R. Li, J. Lin, and P. Lee. CORE: Augmenting Regenerating-Coding-Based Recovery for Single and Concurrent Failures in Distributed Storage Systems. arXiv, preprint arXiv:1302.3344, 2013.
14. I. Reed and G. Solomon. Polynomial Codes over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics, 8(2):300–304, 1960.
15. J. Breckling, Ed., the Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61. Y. Hu, P. P. C. Lee, and K. W. Shum. Analysis and Construction of Functional Regenerating Codes with Uncoded Repair for Distributed Storage Systems. In Proc. of IEEE INFOCOM, Apr 2013.
16. N Cao, S Yu, Z Yang, W Lou, YT Hou. LT codes-based secure and reliable cloud storage service- INFOCOM, 2012 Proceeding IEEE, 2012.
17. Y. Hu, H. Chen, P. Lee,and Y. Tang. NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds. In Proc. of USENIX FAST, 2012.
18. Y. Deswarte and J. Quisquater, "How to Trust Files Stored on Untrusted Servers,"2002.
19. F. Sebe, A. Martinez-Balleste, Y. Deswarte, J. Domingo-Ferrer, and J.-J. Quisquater. "Time- bounded remote file integrity checking." Technical Report 04429, LAAS, July 2004.
20. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Proceedings of the 16thACM conference on Computer and communications security - CCS '09, p. 213, 2009.
21. Q. Wang, S. Member, C. Wang, and K. Ren, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," pp. 1–13.-2009.
22. S. Ni-Na and Z. Hai-Yan, "On Providing Integrity for Dynamic Data Based on the Third-party Verifier in Cloud Computing," 2011 First International Conference on Instrumentation Measurement,Computer Communication and Control, pp. 521–524, Oct. 2011..