A Review on Remote Data Checking Using Code Regeneration in Cloud Storage

D.Sumathi;

Department of Computer Science and Engineering, Malla Reddy Engineering College

[1]sumathi.research28@gmail.com

***Abstract*—** Regenerating codes attract many researchers due to the low repair bandwidth. Several existing remote checking schemes provide only private auditing, and this feature makes the data owner be available all the time and must be active enough for auditing and as well as repairing. This feature seems to be impractical. A review of various existing schemes has been done in order to analyze the benefits of various verification tasks. It is observed that a public auditing scheme for the regenerating code based cloud storage is required and hence this task could be either done by the third party auditor or by the person who might act as a proxy. A secure novel public verifiable authenticator is responsible for the generation of keys and the keys are used for further regeneration process.

***Keywords*—** Regenerating Codes, Proof of Retrievability, Provable Data Possession, Remote Data Checking, Proxy

I. INTRODUCTION

Cloud storage – a new paradigm of hosting various services based on the demand of the users provide several benefits such as data access irrespective of location, reduction in capital expenditure that has been done on hardware, software and personal maintenances, etc. Storage availability and data protection is intrinsic to object storage architecture, so depending on the application, the additional technology, and effort and cost to add availability and protection can be eliminated [1]. The responsibility of a service provider is to monitor the storage maintenance tasks, and if there is any need for additional storage, it has to be monitored and purchased. A web service interface has been installed by the service providers in such a way that the consumers could access the resources immediately.

Cloud storage can be used for copying virtual machine images from the cloud to on-premises locations or to import a virtual machine image from an on-premises location to the cloud image library. In addition, cloud storage can be used to move virtual machine images between user accounts or between data centers. This computing model raises the alarm for the individuals or enterprises since there might be chance of security threat for the data that has been uploaded by the user. Data owners totally lost the control over the data that has been outsourced and thus the availability, data integrity and correctness becomes the challenging task. Several internal/external antagonist might delete or corrupt the data that has been uploaded by the user and on the other hand, chances are there for the service providers to act in such a way that the data loss or corruption could be hidden from the users since the reputation of the providers might be damaged. Hence, a periodical verification of the outsourced data has to be done such that the data maintained must be in a proper manner. Many mechanisms such as PDP (Provable Data Possession) model and POR (Proof of Retrievability) model were proposed by Ateniese et al., Juels & Kaliski. Several redundancy schemes such as erasure codes, replication, and regenerating codes have been used for checking the data integrity.

Data authenticity verification is a crucial issue that has to be addressed. The issue occurs in peer-to-peer storage systems, network file systems, long-term archives, web-service object stores, and database systems. Such systems prevent storage servers from misrepresenting or modifying data by providing authenticity checks when accessing data. Data verification falls into two categories namely provable data possession (PDP) and Proofs of Retrievability (POR).

In PDP, the verification process is done without retrieving the data from the server. This process is made possible by checking the blocks that are chosen at random and thus this model produces probabilistic proofs of possession. Due to this testing, the cost of I/O gets reduced. A stable amount of metadata has to be maintained by the client. This challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. The two provably-secure PDP schemes those are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data.

II. PORS: PROOFS OF RETRIEVABILITY FOR LARGE FILES

The delegations of computing services are based on the demand. Due to the centralization of hardware and software facilities, the energy, computing-system complexity and labor costs could be mitigated. Increasingly, users employ software and data that reside thousands of miles away on machines that they do not own. Grid computing, the harnessing of disparate machines into a unified computing platform, has played a role in scientific computing for some years. Correspondingly, software as a service (SaaS) provides architectures in terms of terminal/mainframe computing seems to be a milestone in the internet-technology strategies of major companies. A POR scheme has to generate a short and clear proof for the data verification. It has to report on the reliable data transfer. The main features of POR protocols are the storage requirements of the verifier, count of the memory accesses for the proverb and communication costs. This scheme facilitates the feature of providing an archive or a back up service so that a proof will be generated and issued to the user so that the user i.e. verifier could start the process of transmitting the data reliably that must be enough for the user to recover the file F. Through this scheme, large file also could be handled with the cryptographic proof of knowledge (POK). In a POR, there is no need for the prover or the verifier to have the knowledge of file that is to be verified. This scheme is an important mechanism for semi-trusted online archives. Those archives that have stored must be verified for its data integrity and correctness. The main objective of POR is to verify the task without downloading the files. Hence, it provides quality-of-service guarantees.

Several research works have been carried out in the data integrity verification process. Data that has been stored remotely has to be checked for it's intact. It acts as both prevention tool and repair tool. At the beginning phase, Remote Data Checking (RDC) [7] was implemented for the single server and later on it has been extended to distributed storage systems that rely on replication and on erasure coding to store data redundantly at multiple servers. The servers that were corrupted could be repaired with the inclusion of the redundancy feature on network coding. Various studies have been carried out in such a way that the cost could be minimized in the prevention phase and as well as in repair phase too. A novel secure and an efficient RDC strategy for network coding-based distributed storage systems have been proposed in [4]. From the experimental results, it is observed that the scheme is inexpensive for both servers and clients.

When the size of the outsourced data is large and the user's constrained resource capability, the verification task in the cloud becomes expensive for the users. Overhead that occurred in the cloud storage must be minimized in such a way that there is no need for the users to perform many operations on the outsourced data. Wang et al suggested a random blind technique so that the problem is resolved with the help of BLS signature based auditing scheme.

III. MR-PDP: MULTIPLE-REPLICA PROVABLE DATA POSSESSION

In the case of distributed storage systems, there might be a chance of multiple unique replicas of the file. Hence, a query could be posted by the client to the distributed system in order to assure the storage of multiple unique copies of the files that are stored in the network even at the time of collision of storage sites. The main feature of any service provider is to maintain the data and do a constant introspection as they must be able to provide the data requested by the data owner at any cost and at any time.

These techniques are applicable in all distributed, untrusted storage systems, replication-based and peer-to-peer storage systems. To ensure the durability and availability of the data, replication is considered as the main feature that has to be followed by an organization. During this process, placement and the count of replicas becomes an issue. Several facilities like re-replicating the data have to be performed as and when replicas fail, evaluation of correctness of replicas and replica movement among sites must be done in order to meet out the data availability. Recently, many investigations have been carried out to check for the strong evidence of storing multiple copies of data on untrusted storage systems. The main cause of replication is to ensure the data availability and durability of the data. An illusion had been made in such a way that the servers store multiple copies of data, but in a real scenario, a single copy might be maintained. This could be overcome by proposing the multiple-replica provable data possession (MR-PDP). The main features of this algorithm are

1. 'n' replicas of files are stored in the system and this could be verified with the help of challenge-response protocol.
2. At the time of query, each unique replica could be provided.
3. 't' time has been calculated for storing a single replica.

This work has been extended on data possession proofs for storing a single copy of a file in a client/server storage system. MR-PDP is considered as an efficient scheme since the number of replicas that have been stored is more rather than the single-replica PDP scheme. The main advantage of this scheme is that this is possible of producing more replicas on demand with less cost even at the time of failure of existing replicas.

A. Distributed data possession checking for securing multiple replicas in geographically-dispersed clouds

Multiple replicas geographically distributed clouds so that the latency has been reduced. Each replica has to be checked for its data integrity and availability. Each replica should have to ensure availability and data integrity features. To verify the replica's integrity and availability, remote data possession checking is used. In order to reduce the I/O cost, an efficient data-possession verifying method is used to generate and check a small hash of the data. Remote checking is considered as a time-consuming due to the several features such as limited bandwidth and a huge data volume. Multiple replicas data possession checking is treated as a challenging task because the optimization of the remote communication cost among multiple geographically dispersed clouds is very difficult. A novel efficient Distributed Multiple Replicas Data Possession Checking (DMRDPC) scheme is proposed in order to resolve new issues. The efficiency could be improved by identifying the optimal spanning tree so that partial order of scheduling multiple replicas data possession checking scheme is performed.

Most recent research on data possession checking considers only single replica. However, multiple replicas data possession checking is much more challenging, since it is difficult to optimize the remote communication cost among multiple geographically-dispersed clouds. In this paper, authors provide a novel efficient Distributed Multiple Replicas Data Possession Checking (DMRDPC) scheme to tackle new challenges. The goal is to improve efficiency by finding an optimal spanning tree to define the partial order of scheduling multiple replicas data possession checking. But since the bandwidths have geographical diversity on the different replica links and the bandwidths between two replicas are asymmetric, it must resolve the problem of finding an Optimal Spanning Tree in a Complete Bidirectional Directed Graph, which we call the FOSTCBDG problem. Particularly, authors provide theories for resolving the FOSTCBDG problem through counting all the available paths that viruses attack in clouds network environment. Also, it helps the cloud users to achieve efficient multiple replicas data possession checking by an approximate algorithm for tackling the FOSTCBDG problem, and the effectiveness is demonstrated by an experimental study. Both Chen *et al.* [8] and Chen and Lee [9] recommended verification methods with the help of regenerating code based cloud storage. But here in the work proposed by them the verification task has been handled by the data owner. To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage together with data integrity checking and failure reparation becomes critical. The proposed system should be explored in the way that suitable for multi–cloud scenario.

IV. REGENERATING CODE

Dimakis et al., is the first person to introduce the regenerating codes for distributed storage so as to reduce the bandwidth. Several notations used here are

'α' refers to the storage servers.

'F' denotes the data file that has to be encoded and stored redundantly across the servers

If any file 'F' that has been stored in the cloud storage has to be retrieved, then any k-out-of α servers could be contacted based on the maximum distance separable feature. If the data is found to be corrupted at a server is detected then the client will contact the remaining servers so that γ bits are downloaded from each server. Thus the corrupted block could be regenerated without recovering the entire original file.

Regenerating codes are important for distributed storage systems for the same reason, i.e., it might be practical to design codes regenerating the data in one node by communicating functions of the data in the surviving nodes with minimum communication cost. This serves as the main motivation for us to look at regenerating codes. However, instead of regenerating the data in one node that are exactly the same as the original data, one may generate another data fragment that is constituted by linear combinations of survival data fragments and preserves the ability to regenerate the original data. This property offers a huge flexibility, which yields a quite natural tradeoff between storage and communication cost to regenerate a data fragment.

V. AUTHENTICATOR

An authenticator is a way to prove to a computer system that you really are who you are (called authentication). It is either:

- A piece of data (often called a token) that you got from the last place where you proved who you are (to save you, or the software, the hassle of logging in again)
- A program, usually running somewhere on the computer network, that takes care of authentication

Authenticator tokens are common when one program needs to authenticate itself to a larger server or cloud repeatedly. For instance, you (the human) might sign on to a secure website with your name and password, after which you can surf around inside the secure server, visiting different web pages. Every time you move to a new page, however, the server must believe that you are the same person who originally signed in (otherwise it will refuse). Your browser keeps an authenticator token, which it sends upon every page request (often as a browser cookie), that does this.

More complex situations might involve a program that runs automatically that similarly requires authentication to get at the data it needs, but there's no human around to log in for them. An authenticator token must be prepared in advance that this program uses. Ultimately, some human must authenticate to create such a token.

- Authenticators are means used to confirm the identity or eligibility of a station, originator, or individual

An authenticator is an entity at one end of a point-to-point LAN segment that facilitates authentication of the entity attached to the other end of that link. In practice, the authenticator is usually a network switch or wireless access point that serves as the point of connection for computers joining the network. The authenticator receives connection requests from a supplicant on the connecting computer.

VI. PROXY

In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or another resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems.

Today, most proxies are web proxies, facilitating access to content on the World Wide Web and providing anonymity.A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

- A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes a tunneling proxy
- A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet)
- A reverse proxy is usually an Internet-facing proxy used as a front-end to control and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load-balancing, authentication, decryption or caching

VII. TYPES OF PROXY

A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

- A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes a tunneling proxy
- A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet)
- A reverse proxy is usually an Internet-facing proxy used as a front-end to control and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load-balancing, authentication, decryption or caching

VIII. USES OF PROXY SERVERS

B. Monitoring and filtering

Content-control software

A content-filtering web proxy server provides administrative control over the content that may be relayed in one or both directions through the proxy. It is commonly used in both commercial and non-commercial organizations (especially schools) to ensure that Internet usage conforms to acceptable use policy.

A content filtering proxy will often support user authentication, to control web access. It also usually produces logs, either to give detailed information about the URLs accessed by specific users, or to monitor bandwidth usage statistics. It may also communicate to daemon-based and/or ICAP-based antivirus software to provide security against virus and other malware by scanning incoming content in real time before it enters the network.

Many work places, schools and colleges restrict the web sites and online services that are made available in their buildings. Governments also censor undesirable content. This is done either with a specialized proxy, called a content filter (both commercial and free products are available), or by using a cache-extension protocol such as ICAP, that allows plug-in extensions to an open caching architecture.

Requests may be filtered by several methods, such as a URL or DNS blacklists blacklist, URL regex filtering, MIME filtering, or content keyword filtering. Some products have been known to employ content analysis techniques to look for traits commonly used by certain types of content providers. Blacklists are often provided and maintained by web-filtering companies, often grouped into categories (pornography, gambling, shopping, social networks, etc.).

Assuming the requested URL is acceptable, the content is then fetched by the proxy. At this point a dynamic filter may be applied on the return path. For example, JPEG files could be blocked based on fleshtone matches, or language filters could dynamically detect unwanted language. If the content is rejected then an HTTP fetch error may be returned to the requester.

Most web filtering company’s use an internet-wide crawling robot that assesses the likelihood that content is a certain type. The resultant database is then corrected by manual labor based on complaints or known flaws in the content-matching algorithms.

Some proxies scan outbound content, e.g., for data loss prevention; or scan content for malicious software.

C. MD5 algorithm

To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage together with data integrity checking and failure reparation becomes critical. The proposed system should be explored in the way that suitable for multi–cloud scenario. A scheme could be recommended on the data integrity verification scheme in regenerating-code-based cloud storage particularly with the functional repair strategy**.** The scheme must ensure the data integrity and the users' computation resources also could be saved. In addition the online burden also might be resolved by the public auditing scheme. This auditing is executed by the TPA.:

IX. CONCLUSION

Through the proposed approach, data is shared effectively among the several groups in cloud. A user is able to share data with others in the group without revealing identity privacy to the cloud. It supports efficient user revocation and new user joining. Then, an effective data access control scheme for multi-group cloud storage systems has been constructed. A proxy could act as a verifier so that the storage is made available and verifiable even after a malicious corruption. Proxy takes the place of the data owner since data owner always stay online in practice. To handle the reparation of the coded blocks and authenticators, proxy is used. The revocable multi-authority is a promising technique, which can be applied in any remote storage systems and online social networks etc. This approach greatly reduces the workload on the storage servers. The proposed scheme is highly efficient and can be feasibly integrated into a regenerating-code-based cloud storage system.

REFERENCES

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

[2] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 584–597.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in Proc. 28th International Conference on Distributed Computing. Systems (ICDCS), Jun. 2008, pp. 411–420.

[5] J. He, Y. Zhang, G. Huang, Y. Shi, and J. Cao, "Distributed data possession checking for securing multiple replicas in geographically dispersed clouds," Journal of computer system science , vol. 78, no. 5, pp. 1345–1358, 2012.

[6] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," Proceedings of IEEE, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[7] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Integrity and Internal Control in Information Systems VI.Berlin, Germany: Springer-Verlag, 2004, pp. 1–11.

[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 Distributed. System, vol. 25, no. 2, pp. 407–416, Feb. 2014.

[9] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote datachecking for network coding-based distributed storage systems," in Proc.ACM Workshop Cloud Computing. Secure. Workshop, 2010, pp. 31–42.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[11] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou,"Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[12] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," IEEE Trans. Service Comput., vol. 5, no. 2, pp. 220–232, Apr./Jun. 2012.