ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

An Effective Method to Release Data Owners from Online Burden Using Public Auditing Scheme for Regenerating-Code-Based Cloud Storage

Samruddhi Pode

M.E, Dept. of Computer Engineering, RMD SSOE, Warje, Pune, India

ABSTRACT: User can remotely store their data and enjoy the on demand high-quality applications and services by using cloud storage. This can be achieving from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. With cloud storage services, it is common place for data to be not only stored in the cloud, but also shared across multiple users. However, public auditing for such shared data while preserving identity privacy remains to be an open challenge. Existing system cannot solve the regenerating problem. This system proposed a public auditing scheme for the regenerating-code-based cloud storage to solve the existing issues.

KEYWORDS: Cloud storage, data integrity, public auditing, regenerating-code-based cloud.

I. Introduction

Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centres. Cloud storage is a model of data storage where the digital data is stored in logical pools. Cloud storing become popular among different sectors of business as well as daily life since it provides a flexible demanded data outsourcing servicing. It also provide good benefits; relief of the burden for service management department, global data access with location independence, and to avoid capital expenditure on hardware and software both. It allows users to upload files that could then be accessed over the internet from a different computer, tablet, smart phone or other networked device, by the same user or possibly by other users, after a password or other authentication is provided [1].However, this concept of data hosting service also brings new security threats toward users' data, thus making individuals or organization still feel hesitant.

The cloud technology have benefits like relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personal maintenances, etc. It is observed that data owners lose ultimate control over the fate of their outsourced data; thus, the correctness, availability and integrity of the data are being put at risk. Outsourcing data storage increases the attack surface area. When data has been distributed, it is stored at more locations increasing the risk of an authorized physical access to the data. On one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt user's data; on the other hand, the cloud service providers may act dishonestly, attempting to hide data corrupted and claiming that the files are still correctly stored in the cloud for reputation or monetary reasons. Thus it makes great sense for users to implement an efficient protocol to perform periodical verifications of their outsourced data to ensure that the cloud indeed maintains their data correctly.

There are many technologies which are used for preserving the privacy of data information on a cloud. There are also many mechanisms, which are dealing with the integrity of outsourced data, proposed under different system and security models up to now. Some of the technologies are described as below-

In 2008, K. D. Bowers, A. Juels, and A. Oprea[1], explored the idea of providing appealing benefits like relief of the burden for storage management, universal data access with location independence, and avoidance of capital

DOI: 10.15680/IJIRCCE.2016. 0406125

expenditure on hardware and software. It allows users to upload files that could then be accessed over the internet from a different computer, tablet, smart phone or other networked device, by the same user or possibly by other users, after a password or other authentication is provided. However, this concept of data hosting service also brings new security threats toward users data, thus making individuals or organization still feel hesitant. Hence we have to more focus on the different methods by using which one can preserve the privacy of their own data on a cloud.

In 2009, H. Shacham and B. Waters[2], suggested a proof-of- retrievability system, where a data storage center convinces a verifier that he is actually storing all of a client's data. They built BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-retrievability with public veriability. Their second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-retrievability scheme with private variability. Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value. But these schemes are only for single-server scenario.

In 2010,C.Wang, Q.Wang, K. Ren, andW. Lou[3], The auditing schemes in this paper consider the large size of the outsourced data as well as users constrained resource capability. But the tasks of auditing and reparation in the cloud can be formidable and expensive for the users.

In 2012, Y. Zhu, H. Hu, G.-J. Ahn, and M.Yu[4], explore the idea of Provable data possession (PDP) which is a technique for ensuring the integrity of data in storage outsourcing for multi-clouds with different redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes by Considering that files are usually striped and redundantlystored across multi-servers or multi-clouds.

In 2012, Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang[5], presented a proxy-based system for multiple-cloud storage called NCCloud (Network Code Cloud). It aims to achieve cost-effective repair for a permanent single-cloud failure. They proposed the implementable design for the functional minimum-storage regenerating code (FMSR), it maintains double fault tolerance and has the same storage cost as in traditional erasure coding schemes. However, these are designed for private audit, only the data owner is allowed to verify the integrity and repair the faulty servers.

In existing system, the system proposed a formal definition of the PDP model for ensuring possession of files on untrusted storage. And also it introduced the concept of RSA-based homomorphic tags and suggested randomly sampling a few blocks of the file. In that work, the system proposed a dynamic version of the prior PDP scheme based on MAC. This scheme allows very basic block operations with limited functionality but block insertions. And it used merkle hash tree to improve the efficiency of dynamic PDP. However, the existing system cannot release the data owner from online burden.

II. RELATED WORK

There are several systems proposed for multiple-cloud storage.

HAIL A High-Availability and Integrity Layer for Cloud Storage places the task of file-integrity checking in the hands of the client or some other trusted, external service and avoids communication among servers. Unlike previous work, which verifies integrity at the level of individual file blocks, HAIL provides assurance at the granularity of a full file [1].

HAIL offers the benefits like:

- *Strong file-intactness assurance*: It enables a set of servers to prove to a client through a challenge-response protocol that a stored file *F* is fully intact such that the client can recover *F* with overwhelming probability.
- *Strong adversarial model:* HAIL protects against an adversary that is *active*. Adversaries can corrupt servers and alter file blocks and mobile, i.e., can corrupt every server over time.
- *Direct client-server communication:* HAIL involves one-to-one communication between a client and servers. Servers need not intercommunicate or even be aware of other servers' existence.

HAIL provides assurance at the granularity of a full file.

NCCloud It is implemented as a proxy that connects user applications and multiple clouds. NCCloud is built on top of network-coding-based storage schemes called regenerating codes. It is built on three layers. The file system layer presents NCCloud as a mounted drive, which can thus be easily interfaced with general user applications. The coding layer deals with the encoding and decoding functions. The storage layer deals with read/write requests with different clouds. NCCloud not only achieves fault tolerance of storage, but also allows cost-effective repair when a cloud permanently fails.

Remote Data Checking scheme Remote data checking (RDC) schemes is presented for distributed storage systems based on network coding. When the client detects the failure of a server, it needs to take measures to ensure the data recovery condition is maintained. Recently, RDC schemes were proposed for replication-based and erasure coding-based distributed storage systems. To the best of our knowledge, RDC was not considered for network coding-based distributed storage systems.

Regenerating-code-based The regenerating codes are the code that achieves the min-cut in the information flow graph. There are two types of regenerating code based one minimum storage requirement space at storage nodes and the minimum bandwidth consumption during repair. Regenerating codes are based on the concept of network coding and tradeoff the repair traffic is reduced among storage nodes. The storage cost and repair traffic achieve optimal, and the optimal points are two, one optimal point to the minimum storage regenerating (MSR) codes, focus on minimize the repair bandwidth the condition that each node stores the minimum amount of data as in Reed-Solomon codes. Another optimal point is the minimum bandwidth regenerating (MBR) codes, which minimize the repair bandwidth further allowing each node to store more data.

III. PROBLEM DEFINITION

To propose a public auditing scheme for the regenerating- code-based cloud storage. The proposed scheme encrypts the coefficients to protect data privacy against the auditor. This scheme can be effectively reduce the computational overhead of the data owner and communication overhead during the audit phase.

Goals and objectives

1. To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage.
2. To release data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical.
3. To take optimization measures to improve the flexibility and efficiency of auditing scheme; thus, the storage overhead of servers, the computational overhead of the data owner and communication overhead during the audit phase can be effectively reduced.
4. To fully ensure the data integrity and save the users computation resources as well as online burden.

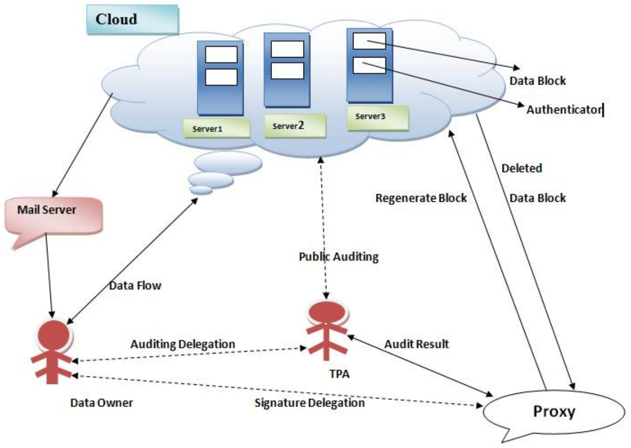
Statement of scope
The system can setup the auditing scheme where the data owner has to initialize its public and secret parameters. In this auditing scheme the cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness. In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.

IV. SYSTEM OVERVIEW AND ARCHITECTURE

We are going to create regenerating code based cloud storage. In this, our system is going to generate clean data which is affected by virus or hacker.
1. Setup: The procedure of initializing the auditing scheme is done by the data owner in the first stage of module. This module will work on various tasks of data owner.
2. Audit: In this module the interaction between cloud servers and TPA is generated using specific algorithms.This interaction with one another aims to take a random sample on the blocks and check the data intactness in this procedure.
3. Repair: When the data owner is offline, the proxy interacts with the cloud servers to repair the wrong server detected by the auditing process. This would be the final module.

Initially, we are providing appropriate regenerating-code-scenario and designing our authenticator based on the BLS signature. This authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. We aim to evaluate the performance of our privacy-preserving public audit scheme during the Setup, Audit and Repair procedure.

We are considering the system model for Regenerating-Code-based cloud storage as Figure below. It consist of four entities: the data owner, who owns large amount of data files which stored in the cloud; the cloud, which are managed by the cloud service provider, provide storage services and have significant computational resources; the third party auditor (TPA), who has expertise and capabilities to conduct public audits on the coded data in the cloud, the TPA is trusted party and its audit result is unbiased for both data owners and cloud servers; and a proxy agent, who is semi-trusted and acts on behalf of the data owner to regenerate authenticators and data blocks on the failed servers during the repair procedure.



V. ALGORITHM OF AUDITING

The polynomial-time algorithms for auditing scheme are consists of: **Setup**, **Audit** and **Repair**.
1)Setup: The data owner maintains the procedure to initialize the auditing scheme.
KeyGen(1^k) → (*pk*, *sk*)*:* This polynomial-time algorithm is run by the data owner to initialize its public and secret parameters by taking a security parameter κ as input.
Degelation(sk) → *(x):* This algorithm represents the interaction between the data owner and proxy. The data owner delivers partial secret key *x* to the proxy through a secure approach.
SigAndBlockGen(sk, F) → *($\oslash$, Ψ, t):* This polynomial time algorithm is run by the data owner and takes the secret parameter *sk* and the original file *F* as input, and then outputs a coded block set , an authenticator set and a file tag *t*.
2)Audit: The cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness in this procedure.
Challenge(F_{info}) → *(C):* This algorithm is performed by the TPA with the information of the file *Fin f o* as input and a challenge *C* as output.
ProofGen(C,$\oslash$,Ψ)→(P): This algorithm is run by each cloud server with input challenge *C*, coded block set and authenticator set , then it outputs a proof *P*.
*Verify(P, pk, C)→ (*0, 1*):* This algorithm is run by TPA immediately after a proof is received. Taking the proof *P*, public parameter *pk* and the corresponding challenge *C* as input, it outputs 1 if the verification passed and 0 otherwise.

3)Repair: In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.
ClaimForRep(Fin f o) → *(Cr):* This algorithm is similar with the *Challenge()* algorithm in the Audit phase, but outputs a claim for repair *Cr* .
GenForRep(Cr,$\oslash$,Ψ) → *(BA):* The cloud servers run this algorithm upon receiving the *Cr* and finally output the block and authenticators set *BA* with another two inputs.

BlockAndSigReGen(Cr , BA) → (ø,Ψ,⊥): The proxy implements this algorithm with the claim *Cr* and responses *BA* from each server as input, and outputs a new coded block set øand authenticator set Ψ if successful, outputting ⊥ if otherwise.

VI. Mathematical Model

For repair bandwidth,

- n- Storage servers
- γ' repair bandwidth
- F- Data file encoded and stored redundantly across n servers
- This F then can be retrieved by taking to any *k*-out-of-*n* servers.
- When data corruption at a server is detected, the client will contact ℓ healthy servers and download β' bits from each server.
- Repair bandwidth will be now,

  $\gamma' = \ell \beta'$

For the fundamental trade off between the storage cost α' and the repair bandwidth γ',

- MBR-Minimum Bandwidth Regenerating point
- MSR-Minimum Storage Regenerating point

Inputs: n, k, ℓ, F
Outputs: $\alpha'_{MSR,} \gamma'_{MSR}$

VII. Simulation and Result

The existing work proposed a public auditing scheme for the regenerating-code-based cloud storage system. This can reduce the repairing bandwidth. Using auditing scheme we are trying to protect the original data privacy against the TPA by randomizing the coefficients in the beginning. We have observed that by using linear combinations of servers inside the cloud we can easily reduce the repairing bandwidth. Hence the memory space will be free and more users now can use the storage services. In our work we are implementing one mail server and using exact repair strategy for regeneration purpose. Hence this work will give efficiency in computing the regenerated blocks.

	Existing System	**Proposing System**
Public Audit ability	yes	yes
Privacy Preserving	yes	yes
Owners off line support	yes	yes
Inform to owners	no	yes
Authenticator availability	no	yes
Functional Repairing	no	yes

DOI: 10.15680/IJIRCCE.2016. 0406125

REFERENCES

1. K. D. Bowers, A. Juels, and A. Oprea, “HAIL: A high-availability and integrity layer for cloud storage,” in Proc. 16th ACM Conf. Comput. Commun. Secure. , 2009, pp. 187–198.
2. H. Shacham and B. Waters, “Compact proofs of retrievability,” in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.
3. O. Rahamathunisa Begam, T. Manjula, T. Bharath Manohar, B. Susrutha, “Cooperative provable data possession for integrity verification in multicloud storage,” IEEE Trans.Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec. 2012.
4. Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang, “NCCloud: Applying network coding for the storage repair in a cloud-of-clouds,” in Proc.USENIX FAST, 2012, p. 21.
5. 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.
6. 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.
7. Sophia S and Dr. Sharvani G S, “A Survey on Regenerating Codes for Distributed Cloud Storage”, IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717–1726, Sep. 2013.
8. Henry C.H. Chen and Patrick P.C. Lee, “Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage: Theory and Implementation”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014

BIOGRAPHY

Samruddhi Devidas Pode is a Master of Engineering student in the Computer Engineering Department, RMD Sinhgad School of Engineering, Savitribai Phule Pune University. She received Bachelor of Engineering degree in 2014 from BDCOE, Wardha. Her research interests are Cloud computing, Parallel Computing, Network security.