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 7, July 2016

Verifying, Regenerating and Preserving Data in Cloud Storage by using Public Auditing, Regenerating Code

Ch.Ramesh Kumar[1], Tadikonda Nagamalleswararao[2]

Associate Professor & Head, Department of CSE, Malla Reddy Engineering College &Management Sciences, Kistapur, Medchal, Hyderabad, India

M.Tech Student(CSE) Department of CSE, Malla Reddy Engineering College & Management Sciences, Kistapur, Medchal, Hyderabad, India

ABSTRACT: To shield outsourced data in cloud storage against corruptions, adding up fault tolerance to cloud storage jointly with data integrity checking and failure reparation becomes vital. In recent times, regenerating codes have gained fame due to their lower repair bandwidth at the same time as providing fault tolerance. Existing remote checking methods for regenerating-coded data only provide personal auditing, requiring data owners to always keep online and handle auditing, as well as repairing, which is sometimes not practical. In this paper, we recommend a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration crisis of failed authenticators in the lack of data owners, we commence a proxy, which is confidential to regenerate the authenticators, into the usual public auditing system model. Introduction of cloud audit server eliminates the contribution of user in the auditing and in the pre-processing phases. In our scheme client is not must to store any large set of data locally except a secret key which is required for encryption. Contrast to previous method, we also avoid the requirement of encrypting complete data at client side, by this means saving client computational assets. The planned scheme is applicable for big static data such as video files, audio files and social networking data etc.

KEYWORDS: Code regeneration, public auditing, and cloud storage etc.

I. INTRDOUCTION

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. Recently, regenerating codes have gained popularity due to their lower repair bandwidth while providing fault tolerance. Existing remote checking methods for regenerating-coded data only provide private auditing, requiring data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical. In this , to propose a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owners, to introduce a proxy ,which is privileged to regenerate the authenticators, into the traditional public auditing system model. Thus, this scheme can completely release data owners from online burden. And randomize the encode coefficients with a pseudorandom function to preserve data privacy. Cloud storage is now gaining popularity because it offers a flexible on-demand data outsourcing service with appealing benefits: 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., Nevertheless, this new paradigm of data hosting service also brings new security threats toward users data, thus making individuals or enter priers still feel hesitant. It is noted 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. On the one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt users' data; on the other hand, the cloud service providers may act dishonestly, attempting to hide data loss or corruption and claiming that the files are still correctly stored in the cloud

DOI: 10.15680/IJIRCCE.2016. 0407038

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.

1.1 AIM AND OBJECTIVE:

Remote checking methods for regenerating-coded data only provide private auditing, requiring data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical. In this ,to propose a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owner, in this introduce a proxy, which release data owner from online burden.

1.2SCOPE OF RESEARCH

a) We design a novel homomarphic authenticator based on BLS signature, which can be generated by a couple of secret keys and verified publicly. Utilizing the linear subspace of the regenerating codes, the authenticators can be computed efficiently. Besides, it can be adapted for data owners equipped with low end computation devices (e.g. Tablet PC etc.) in which they only need to sign the native blocks.

b) To the best of our knowledge, our scheme is the first to allow privacy-preserving public auditing for regenerating- code-based cloud storage. The coefficients are masked by a PRF(Pseudorandom Function) during the Setup phase to avoid leakage of the original data. This method is lightweight and does not introduce any computational overhead to the cloud servers or TPA.

c) Our scheme completely releases data owners from online burden for the regeneration of blocks and authenticators at faulty servers and it provides the privilege to a proxy for the reparation. Optimization measures are taken to improve the flexibility and efficiency of our 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.

II. LITERATURE SERVEY

2.1 An efficient and secure dynamic auditing protocol for data storage in cloud computing.

In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. In this paper, we first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, to extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model. We further extend our auditing protocol to support batch auditing for both multiple owners and multiple clouds, without using any trusted organizer. The analysis and simulation results show that our proposed auditing protocols are secure and efficient, especially it reduce the computation cost of the auditor.

2.2 Cooperative Provable Data Possession for Integrity Verification in Multi cloud Storage.

Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. We present a cooperative PDP (CPDP) scheme based on homomarphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi prove zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non cooperative approaches.

2.3 PDP: Provable Data Possession

Many storage systems rely on replication to increase the availability and durability of data on un trusted storage systems. At present, such storage systems provide no strong evidence that multiple copies of the data are actually stored. Storage servers can collude to make it look like they are storing many copies of the data, whereas in reality they only store a single copy. We address this shortcoming through multiple-replica provable data possession (PDP): A provably-secure scheme that allows a client that stores t replicas of a file in a storage system to verify through a challenge-response protocol that each unique replica can be produced at the time of the challenge and that the storage system uses t times the storage required to store a single replica. PDP extends previous work on data possession proofs for a single copy of a file in a client/server storage system. Using PDP to store t replicas is computationally much more efficient than using a single-replica PDP scheme to store t separate, unrelated files (e.g., by encrypting each file separately prior to storing it). Another advantage of PDP is that it can generate further replicas on demand, at little expense, when some of the existing replicas fail.

III. PROBLEM STATEMENT

The data/file is stored directly into cloud, so there is no backup of same data/file in any other machine. This will affect the user to access data whenever needed. There is no as such scenario exists which will regenerate lost data/file in case of system failure. So concept of privacy preserving public auditing along with regeneration code has been introduced.

IV. EXISTING SYSTEM

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.

a) Existing remote checking methods for regenerating-coded data only provide private auditing, requiring data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical .

b) Many mechanisms dealing with the integrity of outsourced data The most significant work among these studies are the PDP (provable data possession) model and POR model, which were originally proposed for the single-server scenario.

Disadvantages:-

i. They are designed for private audit, only the data owner is allowed to verify the integrity and repair the faults .

ii. The auditing schemes in existing imply the problem that the data owner need to always stay online.

iii. Privacy is low.

V. PROPOSED SYSTEM

a) In this, to focus on the integrity verification problem in regenerating-code-based cloud storage.

b) To propose a public auditing scheme for the regenerating-code-based cloud storage, in which the integrity checking and regeneration (of failed data blocks and authenticators) are implemented by a third-party auditor and a semi-trusted proxy separately on behalf of the data owner.

c) To randomize the encode coefficients with a pseudorandom function to preserve data privacy.

Advantages:

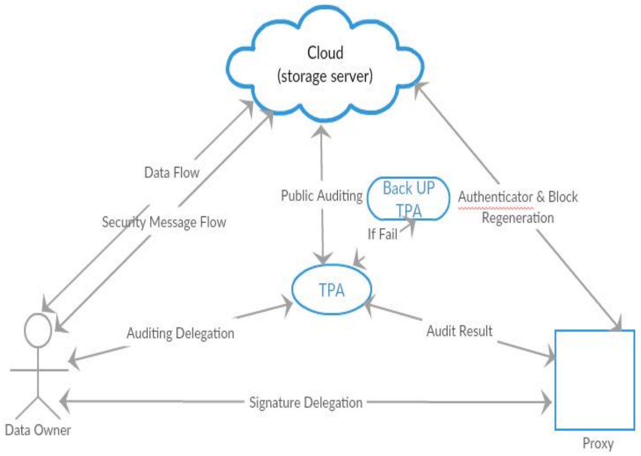
i. In the absence of data owner, to introduce a proxy, this is privileged to regenerate the authenticators.

ii. Data owner is released from online burden

iii. Privacy is more

iv. Avoid the leakage of original data

5. 1 System architecture:-



We consider the auditing system model for Regenerating-Code-based cloud storage, which involves four entities: the data owner, who owns large amounts of data files to be stored in the cloud; the cloud, which are managed by the cloud service provider, provide storage service 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 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. Notice that the data owner is restricted in computational and storage resources compared to other entities and may becomes off-line even after the data upload procedure. The proxy, who would always be online, is supposed to be much more powerful than the data owner but less than the cloud servers in terms of computation and memory capacity. To save resources as well as the online burden potentially brought by the periodic auditing and accidental repairing, the data owners resort to the TPA for integrity verification and delegate the reparation to the proxy.

VI. IMPLEMENTATION

6.1 Regenerating code:

Regenerating codes are first introduced for distributed storage to reduce the repair bandwidth. Viewing cloud storage to be a collection of n storage servers, data file F is encoded and stored redundantly across these servers. Then F can be retrieved by connecting 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, thus regenerating the corrupted blocks without recovering the entire original file.

6.2 Design goals:

To correctly and efficiently verify the integrity of data and keep the stored file available for cloud storage, our proposed auditing scheme should achieve the following properties:

Public Auditable : To allow TPA to verify the intactness of the data in the cloud on demand without introducing additional online burden to the data owner.

Storage Soundness: to ensure that the cloud server can never pass the auditing procedure except when it indeed manage the owner's data intact.

Privacy Preserving: to ensure that neither the auditor nor the proxy can derive users' data content from the auditing and reparation process.

Authenticator Regeneration: the authenticator of the repaired blocks can be correctly regenerated in the absence of the data owner.

Error Location: to ensure that the wrong server can be quickly indicated when data corruption is detected.

6.3 Definition of our auditing scheme:

Our auditing scheme consists of three procedures: Setup, Audit and Repair. Each procedure contains certain polynomial-time algorithms as follows:

Setup: The data owner maintains this procedure to initialize the auditing scheme. KeyGen(1κ) → (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.

Delegation(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.

Sig And Block Gen (sk, F) → (φ,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.

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 F info as input and a challenge C as output.

Proof Gen(C,_,) → (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.

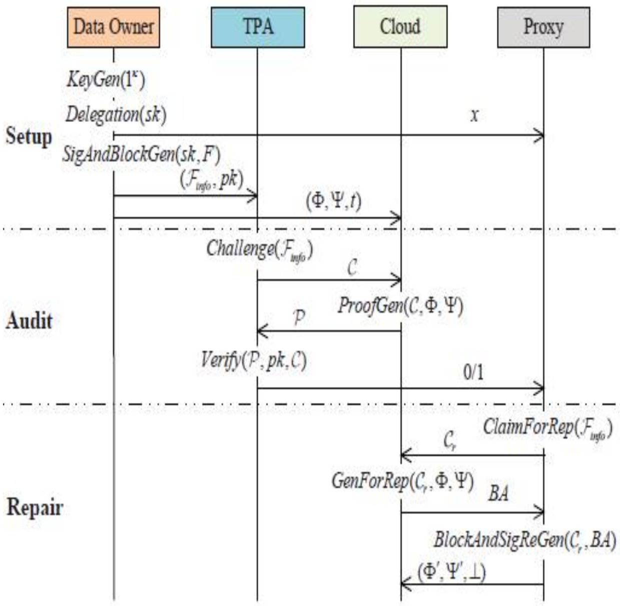


Fig:- The Sequence Chart of Our Scheme

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.

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.

6.4 Enabling privacy –preserving auditable:

The privacy protection of the owner's data can be easily achieved through integrating with the random proof blind technique or other technique . However, all these privacy-preservation methods introduce additional computation overhead to the auditor, who usually needs to audit for many clouds and a large number of data owners; thus, this could possibly make it create a performance bottleneck. Therefore, we prefer to present a novel method, which is more light-weight, to mitigate private data leakage to the auditor. Notice that in a regenerating-code-based cloud storage, data blocks stored at servers are coded as linear combinations of the original blocks Supposing that the curious TPA has recovered m coded blocks by elaborately performing Challenge-Response procedures.

VII. CONCLUSION

In this , to propose a public auditing scheme for the regenerating-code-base cloud storage system, where the data owners are privileged to delegate TPA for their data validity checking. To protect the original data privacy against the TPA, we randomize the coefficients during the auditing process. Considering that the data owner cannot always stay online in practise, in order to keep the storage available and verifiable after a malicious corruption, we introduce a semi-trusted proxy into the system model and provide a privilege for the proxy to handle the reparation of the coded blocks and authenticators. To better appropriate for the regenerating-code-scenario, This authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. Extensive analysis shows that the scheme is provable secure, and the performance evaluation shows that the scheme is highly efficient and can be feasibly integrated into a regenerating-code-based cloud storage system.

VIII. FUTURE ENHANCEMENT

As the task of checking the dynamic data integrity is done by TPA, that is, third party auditor, on behalf of cloud client, the involvement of the client can be eliminated. Moreover, there are a number of challenges in implementing data dynamics. Generating data integrity proofs while considering dynamic nature of the cloud is also contemplated as a challenge for integrity maintenance .Furthermore, block level checking schemes are a bit complex and implementing those in an efficient way can be also regarded as a challenging task .Security and complexity are two contradictory terms. A level of balance has to be established between them .Thus being developed for remote data integrity checking should be time and storage efficient and well suited. Thus ,Future work aims at implementing these at minimal costs.

ACKNOWLEDGMENT

It is with great pleasure that I acknowledge the enormous assistance excellent co-operation extended to by the following respected personalities. Firstly I would like to express my sincere thanks to our respected Principal **Dr. A.Siva kumar** for providing us the necessary infrastructure, laboratories and requirement. I am greatly indebted to, Head of Computer Department, and my respected Guide. **Ch. Ramesh Kumar** for providing valuable guidance.

REFERENCES

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee,D. Patterson, A. Rabkin, and I. Stoica, "Over the mists: A Berkeley perspective of distributed computing," Dept. Electrical Eng. also, Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, p. 13, 2009.
[2] G. Ateniese, R. Blazes, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Tune, "Provable information ownership at untrusted stores," in Proceedings of the fourteenth ACM Conference on Computer and Communications Security, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 598–609.
[3] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for extensive records," in Proceedings of the fourteenth ACM meeting on Computer and correspondences security. ACM, 2007, pp. 584–597.
[4] R. Curtmola, O. Khan, R. Blazes, and G. Ateniese, "Mr-pdp: Multiple imitation provable information ownership," in Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on. IEEE, 2008,pp. 411–420.
[5] K. D. Nooks, A. Juels, and A. Musical show, "Hail: a high-accessibility and uprightness layer for distributed storage," in Proceedings of the sixteenth ACM gathering on Computer and correspondences security. ACM, 2009, pp. 187–198.
[6] J. He, Y. Zhang, G. Huang, Y. Shi, and J. Cao, "Circulated information ownership checking for securing different copies in topographically scattered mists," Journal of Computer and System Sciences, vol. 78, no. 5, pp. 1345–1358, 2012.
[7] B. Chen, R. Curtmola, G. Ateniese, and R. Blazes, "Remote information checking for system coding-based appropriated stockpiling frameworks," in Proceedings of the 2010 ACM workshop on Cloud processing security workshop. ACM, 2010, pp. 31–42.
[8] H. Chen and P. Lee, "Empowering information uprightness security in recovering coding-based distributed storage: Theory and execution," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 2, pp. 407–416, Feb 2014.
[9] K. Yang and X. Jiao, "An effective and secure element evaluating convention for information stockpiling in distributed computing," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 9, pp. 1717–1726, 2013.
[10] Y. Zhu, H. Center point, G.- J. An, and M. Yu, "Agreeable provable information ownership for respectability check in multicloud stockpiling," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 12, pp. 2231– 2244, 2012.

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 7, July 2016

BIOGRAPHY

Ch. Ramesh Kumar, working as Assoc. Prof & Head of the Department of Computer Science and Engineering in Malla Reddy Engineering College & Management Sciences, Kistapur, Medchal, Hyderabad. Affiliated to JNTUH, HYDERABAD, TELANGANA. India. he has several international publications to his credit. His research interests include Software reuse, Software performance, Software testing ,Data Mining and cloud computing

Tadikonda Nagamalleswararao, completed his B.Tech in Lenora college of engineering ,2011. He is pursuing M.Tech. in Computer Science & Engineering from Department of Computer Science & Engineering in Malla Reddy Engineering College & Management Sciences, Kistapur, Medchal, Hyderabad. Affiliated to JNTUH, HYDERABAD, TELANGANA., India.. His research interest include cloud, data mining, big data, wireless , knowledge & data engineering and networking.

DOI: 10.15680/IJIRCCE.2016. 0407038