ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Privacy Preserving Encryption against Untrusted Servers

Basani Archana Reddy
B.Tech 4th Year
Department of CSE
GITAM University,
Visakhapatnam.

Annepu Divya Lekha Sree
B.Tech 4th Year
Department of CSE
GITAM University,
Visakhapatnam.

Aditya Tarigopula
B.Tech 4th Year
Department of CSE
GITAM University,
Visakhapatnam.

V.Gopi Krishna
Project Guide
Department of CSE
GITAM University,
Visakhapatnam.

Abstract:

To provide security for the outsourced data in cloud storage against various problems and provide data integrity becomes difficult. Fault tolerance is also important issue for protecting data in the cloud. Now a day's regenerating codes got importance because of their lower repair bandwidth while providing fault tolerance. Previous remote checking methods for regenerating coded data only provide private auditing, requiring data holders to always keep online and handle auditing, as well as repairing, which is sometimes difficult. In this paper we are going to implement a public auditing scheme for the regenerating code based cloud storage. To obtain solution for regeneration problem of failed authenticators in the absence of data holders, we make a proxy, which is privileged to regenerate the authenticators, in the traditional public auditing system model. We also design a novel public verifiable authenticator, which is made by some keys. Thus, this scheme can almost release data holders from online burden. We also randomize the encode coefficients with a pseudorandom function to sure data privacy. Extensive security analysis shows this scheme is secure and provable under random oracle model. Experimental evaluation model indicates that this scheme is highly efficient and can be feasibly integrated into the regenerating cloud based storage.

Keywords: ***Cloud storage, regenerating codes, public audit, privacy preserving, authenticator regeneration Proxy.***

Introduction:

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server).

This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rack space, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

Cloud storage is based on highly virtualized infrastructure and is like broader cloud computing in terms of accessible interfaces, near-instant elasticity and scalability, multi-tenancy, and metered resources. Cloud storage services can be utilized from an off-premises service or deployed on-premises.

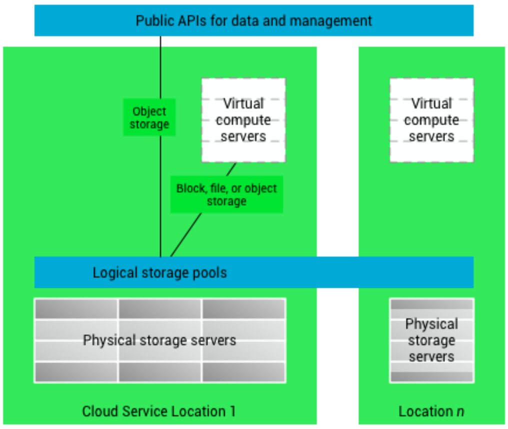
Cloud storage typically refers to a hosted object storage service, but the term has broadened to include other types of data storage that are now available as a service, like block storage.

Object storage services like Amazon S3 and Microsoft Azure Storage, object storage software like Openstack Swift, object storage systems like EMC Atmos and Hitachi Content Platform, and distributed storage research projects like OceanStore[5] and VISION Cloud [6] are all examples of storage that can be hosted and deployed with cloud storage characteristics.

Cloud storage is:

- Made up of many distributed resources, but still acts as one - often referred to as federated storage clouds
- Highly fault tolerant through redundancy and distribution of data
- Highly durable through the creation of versioned copies
- Typically eventually consistent with regard to data replicas

High level cloud storage architecture



In this paper, we concentrate on the integrity verification problem in regenerating-code-based cloud storage, specialy with the functional repair strategy. Similar studies have been performed by Chen et al. [7] and Chen and Lee [8] individually. [7] extended the single-server CPOR scheme (private version) to the regenerating code-scenario; [8] designed and implemented a data integrity protection (DIP) scheme for FMSR-based cloud storage and the scheme is adapted to the thin-cloud setting.1 However, both of them are designed for private audit, only the data owner is allowed to check the integrity and repair the damaged servers. Considering the huge size of the outsourced data and the user's constrained resource capability, the tasks of auditing and reparation in the cloud can be formidable and costly for the users. The overhead of using cloud storage should be minimized as much as possible such that a user does not need to perform so many operations to their outsourced data (in additional to retrieving it). In particular, users may not want to go through the difficulties in verifying and reparation. The auditing schemes in [7] and [8] imply the problem that users need to always stay online, which may impede its adoption in practice, specially for long-term archival storage.

To fully ensure the data integrity and save the users' computation resources as well as online burden, we propose public auditing scheme for the regenerating-codebased cloud storage, in which the integrity checking and regeneration are implemented by a third-party auditor and a semi-trusted proxy separately on behalf of the data owner. Instead of directly applying the old public auditing scheme [12] to the multi-server setting, we design a novel authenticator, which is more suitable for regenerating codes.

Literature Survey

C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," presented privacy-preserving public auditing system for data storage security in Cloud Computing.

C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," proposed that a secure cloud storage system supporting privacy-preserving public auditing.

K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," proposed an efficient and inherently secure dynamic auditing protocol. It protects the data privacy against the auditor by combining the cryptography method with the bilinearity property of bilinear paring, rather than using the mask technique.

C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, “Toward secure and dependable storage services in cloud computing,” Proposed flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server.

In this paper we are going to propose a public auditing scheme for the regenerating code based cloud storage. To obtain solution for regeneration problem of failed authenticators in the absence of data holders, we make a proxy, which is privileged to regenerate the authenticators, in the traditional public auditing system model. We also design a novel public verifiable authenticator, which is made by some keys. Thus, this scheme can almost release data holders from online burden. We also randomize the encode coefficients with a pseudorandom function to sure data privacy.

Extensive security analysis shows this scheme is secure and provable under random oracle model. Experimental evaluation model indicates that this scheme is highly efficient and can be feasibly integrated i regenerating cloud based storage.

Existing System:

- Many mechanisms dealing with the integrity of outsourced data without a local copy have been proposed under different system and security models up to now. The most significant work among these studies are the PDP (provable data possession) model and POR (proof of retrievability) model, which were originally proposed for the single-server scenario by Ateniese et al. and Juels and Kaliski, respectively.

- Considering that files are usually striped and redundantly stored across multi-servers or multi-clouds, explore integrity verification schemes suitable for such multi-servers or multi-clouds setting with different redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes.

- Chen et al. and Chen and Lee separately and independently extended the single-server CPOR scheme to the regeneratingcode- scenario; designed and implemented a data integrity protection (DIP) scheme for FMSR-based cloud storage and the scheme is adapted to the thin-cloud setting.

Disadvantages of Existing System:

- They are designed for private audit, only the data owner is allowed to verify the integrity and repair the faulty servers.

- Considering the large size of the outsourced data and the user’s constrained resource capability, the tasks of auditing and reparation in the cloud can be formidable and expensive for the users

- The auditing schemes in existing imply the problem that users need to always stay online, which may impede its adoption in practice, especially for long-term archival storage.

Proposed System:

- In this paper, we focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy. To fully ensure the data integrity and save the users’ computation resources as well as online burden, we 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.
- Instead of directly adapting the existing public auditing scheme to the multi-server setting, we design a novel authenticator, which is more

appropriate for regenerating codes. Besides, we “encrypt” the coefficients to protect data privacy against the auditor, which is more lightweight than applying the proof blind technique and data blind method.

- We design a novel homomorphic authenticator based on BLS signature, which can be generated by a couple of secret keys and verified publicly.

Advantages of Proposed System:

- 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.

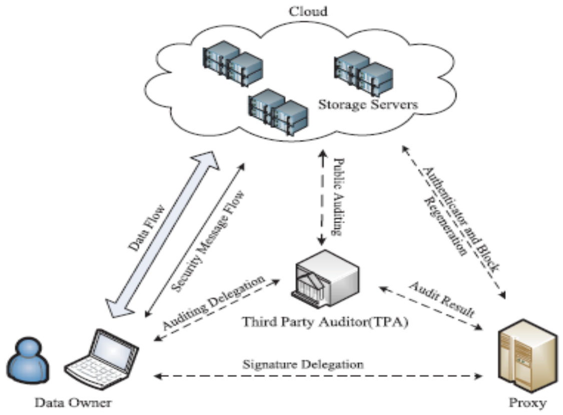
- To the best of our knowledge, our scheme is the first to allow privacy-preserving public auditing for regenerating code- based cloud storage. The coefficients are masked by a PRF (Pseudorandom Function) during the Setup phase to avoid leakage of the original data. This method is lightweight and does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

- 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.

- Our scheme is provable secure under random oracle model against adversaries

System Architecture:



System Model

It consists of four modules,

1. Cloud Module.
2. Proxy Server Module.
3. Group Member Module.
4. User Revocation Module.

Cloud Server

A local Cloud which provides priced abundant storage services are been created in this module. The users can upload their data in the cloud. This module can be developed where the cloud storage can be made secure. The cloud is not fully honorable by users since the CSPs are very likely to be outside of the cloud users’ trusted domain. Similar to that the cloud server is genuine but curious. That is, the cloud server will not maliciously delete or modify user data due to the protection of data investigating schemes, but will try to learn the content of the stored data and the identities of cloud users.

Proxy Server Deployment

Group manager takes charge of followings,

- Signature Generation
- Signature Verification
- Content Regeneration

A proxy agent acts on behalf of the data owner to regenerate authenticators and data blocks on the 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 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.

Group Member Generation

Group members are a set of registered users that will
1. Store their private data into the cloud server and
2. Share them with others in the group.

The group memberships are dynamically changed, due to the staff resignation and new employee participation in the company. The group member had the ownership of changing the files in the group. Whoever in the group can view the files which are uploaded in their group and also modify it.

User Revocation

User revocation is performed by the proxy via a public available RL based on which group members can encrypt their data files and assure the confidentiality against the revoked users. No unauthorized access to the document is encouraged in the cloud storage. So the data should be provided rights to modify only by the group's own users. Other members cannot modify the content. Once if any user tries to hack the private key of another group and trying to modify this will be detected by the cloud server and the user's account will be revoked by the user.

Conclusion

In this paper, we present a public auditing scheme for the regenerating-code-based cloud storage system, where the data owners are privileged to delegate TPA for their data validity checking. To provide security to the original data privacy against the TPA, we randomize the coefficients in the starting rather than applying the blind technique within the auditing process.

Data owner cannot always stay online always, in order to keep the storage available and verifiable after a malicious corruption, we present a semi-trusted proxy into the system model and give a privilege for the proxy to maintain the reparation of the coded blocks and authenticators. To better appropriate for the regenerating code-scenario, we design our authenticator based on the BLS signature. This authenticator can be easily generated by the data owner at the same time with the encoding procedure. Extensive analysis provides that our scheme is provable secure, and the performance evaluation shows that our scheme is highly efficient and can be feasibly integrated into a regenerating-code-based cloud storage system.

References:

[1] Li Weng, Laurent Amsaleg, and Teddy Furon "Privacy-Preserving Outsourced Media Search" IEEE Transactions on Knowledge and Data Engineering (Volume: 28, Issue: 10, Oct. 1 2016), 07 July 2016

[2] Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 7, JULY 2015.

[3] 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.

[4] 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.

[5] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MRPDP: Multiple-replica provable data possession," in Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2008, pp. 411–420.

[6] 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. Secur., 2009, pp. 187–198.

[7] J. He, Y. Zhang, G. Huang, Y. Shi, and J. Cao, “Distributed data possession checking for securing multiple replicas in geographically dispersed clouds,” J. Comput. Syst. Sci., vol. 78, no. 5, pp. 1345–1358, 2012.

[8] 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.

[9] T.Yamini Koteswari & Dr. B.Bhanu Prakash, Improving a New Approach For Providing Privacy Preserving With Attribute Based Encryption, IJMETMR, Vol 2, Issue 7, http://www.ijmetmr.com/oljuly2015/TYaminikoteswari-BBhanuPrakash-342.pdf

[10] K. Yang and X. Jia, “An efficient and secure dynamic auditing protocol for data storage in cloud computing,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717– 1726, Sep. 2013.

[11] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, “Cooperative provable data possession for integrity verification in multicloud storage,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec. 2012.

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

[13] H. Shacham and B. Waters, “Compact proofs of retrievability,” in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.