Review Paper on Privacy Preserving Public Auditing For Regenerating Code Based Cloud Storage

Mrs. Swapnali Manish Nehete[1], Prof. Yogesh S. Patil[2], Prof. Dinesh D. Patil[3]

[1]*M.E. Student, Dept. of Computer Science and Engineering, SSGBCOET, Bhusawal, Maharashtra, India*
[2]*Assistant Professor, Computer Science and Engineering, SSGBCOET, Bhusawal, Maharashtra, India*
[3]*HOD, Computer Science and Engineering, SSGBCOET, Bhusawal, Maharashtra, India*

ABSTRACT

In this paper, we propose a public auditing scheme for the regenerating-code-based cloud storage. 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. To solve the regeneration problem of failed authenticators in the absence of data owners, we introduce a proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Moreover, we design a novel public verifiable authenticator, which is generated by a couple of keys and can be regenerated using partial keys. Thus, our scheme can completely release data owners from online burden. In addition, we randomize the encode coefficients with a pseudorandom function to preserve data privacy. Extensive security analysis shows that our scheme is provable secure under random oracle model and experimental evaluation indicates that our scheme is highly efficient and can be feasibly integrated into the regenerating-code-based cloud storage.

Keywords: *Cloud storage, regenerating codes, public audit, privacy preserving, authenticator regeneration, proxy, privileged, provable secure*

1. INTRODUCTION

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 paper, we 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, we introduce a proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. The promise to deliver IT as a service is addressed a large range of consumers, from small and medium-sized enterprises (SMEs) and public administrations to end-users. According to industry analysts, the ICT sector is poised for strong growth of cloud services [11]. Users are creating an ever-growing quantity of personal data. IDC predicts the amount of information in the digital universe would fill a stack of iPad Air tablets reaching 2/3 of the way to the moon (253,704 Kilometer). By 2020, there will be 6.6 stacks. Today, the average household creates enough data to fill 65 iPhones

(32 GB) per year. In 2020, this will grow to 318 iPhones. Today, if a byte of data were a gallon of water, in only 10 seconds there would be enough data to fill an average house. In 2020, it will only take 2 seconds.

2. OVERVIEW

We provide secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. To achieve secure data sharing for dynamic groups in the cloud, we expect to combine the group signature or group master key and dynamic broadcast encryption techniques. Specially, the group signature scheme enables users to anonymously use the cloud resources, and the dynamic broadcast encryption technique allows data owners to securely share their data files with others including new joining users.

The proxy server computes the revocation parameters and make the result public available by migrating them into the cloud. Such a design can significantly reduce the computation overhead of users to encrypt files and the cipher text size. Specially, the computation overhead of users for encryption operations and the cipher text size is constant and independent of the revocation users. The proxy maintain the signature delegation work which generates the private and public key of each group so that the permission for the access of file can be restricted. Revocation is user is performed if any user make unauthenticated action on any data in the cloud. Also if a data has been modified by the user it will be detected, penalized and the code will be regenerated by the proxy.

In this the user revocation is performed by the group manager via a public available revocation list (RL), based on which group members can encrypt their data files and ensure the confidentiality against the revoked users. The list is characterized by time stamp t1, t2 …tr. In the proposed system once the user time stamp over does not wait for the group manager to update the time stamp or revocation list here once the time over the user immediately send request for extra time for access the data to the cloud. Then the cloud will send that request to the group manager once the see it and give permission then the cloud will time to access the data but if the group manager did not give permission then the cloud will not give permission for access of the data.

3. RELATED WORKS

Advances in networking technology and an increase in the need for computing resources have prompted many organizations to outsource their storage and computing needs. This new economic and computing model is commonly referred to as cloud computing and includes various types of services such as: infrastructure as a service (IaaS), where a customer makes use of a service provider’s computing, storage or networking infrastructure; platform as a service (PaaS), where a customer leverages the provider’s resources to run custom applications; and finally software as a service (SaaS), where customers use software that is run on the provider’s infrastructure. Cloud infrastructures can be roughly categorized as either private or public. In a private cloud, the infrastructure is managed and owned by the customer and located on premise (i.e., in the customer’s region of control). In particular, this means that access to customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider and is located off-premise (i.e., in the cloud service provider’s region of control).

This means that customer data is outside its control and could potentially be granted to untrusted parties. Storage services based on public clouds such as Microsoft’s Azure storage service and Amazon’s S3 provide customers with scalable and dynamic storage. By moving their data to the cloud customers can avoid the costs of building and maintaining a private storage infrastructure, opting instead to pay a service provider as a function of its needs. For most customers, this provides several benefits including availability (i.e., being able to access data from anywhere) and reliability (i.e., not having to worry about backups) at a relatively low cost. While the benefits of using a public cloud infrastructure are clear, it introduces significant security and privacy risks. In fact, it seems that the biggest hurdle to the adoption of cloud storage (and cloud computing in general) is concern over the confidentiality and integrity of data.

While, so far, consumers have been willing to trade privacy for the convenience of software services (e.g., for web-based email, calendars, pictures etc…), this is not the case for enterprises and government organizations. This reluctance can be attributed to several factors that range from a desire to protect mission-critical data to regulatory obligations to preserve the confidentiality and integrity of data. The latter can occur when the customer is responsible for keeping personally identifiable information (PII), or medical and financial records. So while cloud storage has enormous promise, unless the issues of confidentiality and integrity are addressed many potential customers will be reluctant to make the move.

To address the concerns outlined above and increase the adoption of cloud storage, we argue for designing a virtual private storage service based on new cryptographic techniques. Such a service should aim to achieve the "best of both worlds" by providing the security of a private cloud and the functionality and cost savings of a public cloud. More precisely, such a service should provide (at least): confidentiality: the cloud storage provider does not learn any information about customer data integrity: any unauthorized modification of customer data by the cloud storage provider can be detected by the customer non repudiation: any access to customer data is logged, while retaining the main benefits of a public storage service:

Availability: customer data is accessible from any machine and at all times
Reliability: customer data is reliably backed up efficient retrieval: data retrieval times are comparable to a public cloud storage service
Data sharing: customers can share their data with trusted parties.

An important aspect of a cryptographic storage service is that the security properties described above are achieved based on strong cryptographic guarantees as opposed to legal, physical and access control mechanisms. We believe this has several important benefits. There are number of similar works has been contributed by number of users. A secure data regeneration scheme for cloud storage has been developed which includes a new cryptographic method for secure Proof of Ownership (PoW), based on the joint use of convergent encryption and the Merkle-based Tree, for improving data security in cloud storage systems, providing dynamic sharing between users and ensuring efficient data deduplication. This idea consists in using the Merkle-based Tree over encrypted data, in order to derive a unique identifier of outsourced data. On one hand, this identifier serves to check the availability of the same data in remote cloud servers. On the other hand, it is used to ensure efficient access control in dynamic sharing scenarios.

It propose a solution that provides both security and regeneration and retains benefits offered by each technique. ClouRegen makes use of convergent encryption but prevents the dictionary attacks. The components involved in ClouRegen are: the basic cloud storage provider, a metadata manager and an additional server. The server guarantees data confidentiality even for predictable files. The metadata manager provides a system for key-management and block-level regeneration. Convergent encryption (CE) is a technique that can meet the requirements of two conflicting solutions between regeneration and encryption. In CE, the encryption key is derived and computed based on the data provided. For instance, the key can be the result of the hash of the data segment. However, convergent encryption has various well-known weaknesses despite of its suitability. One common vulnerability is the dictionary attack, in which an attacker manages to generate a potential encryption key and, by comparing the two cipher texts, check whether a file has already been stored or not.

A Fast and secure backups with encrypted de- regeneration has also been focused on the security and efficiency of cloud storage, namely that clients outsource their data to cloud storage servers. While cloud storage offers compelling scalability and availability advantages over the current paradigm of "one storing and maintaining its own IT systems and data", it does not come without security concerns. This has led to studies on cloud storage security and efficiency, which are, however, addressed separately as we discuss below. From the perspective of cloud storage security, there have been two notable notions:

Proof of Data Possession (PDP): This notion was introduced by Ateniese et al. [2]. It allows a cloud client to verify the integrity of its data outsourced to the cloud in a very efficient way (i.e., far more efficient than the straightforward solution of downloading the data to the client-end for verification). This notion has been enhanced in various ways [8, 3, and 15].

Proof of Retrievability (POR): This notion was introduced by Juels and Kaliski [10]. Compared with PDP, POR offers an extra property that the client can actually "recover" the data outsourced to the cloud (in the flavor of "knowledge extraction" in zero-knowledge proof).

This notion has been enhanced and extended in multiple aspects from the perspective of cloud storage efficiency, deduplication technique has become a common practice of many cloud vendors. In our data integrity protocol the TPA needs to store only a single cryptographic key irrespective of the size of the data file F and two functions which generate a random sequence. The TPA does not store any data with it.

The TPA before storing the file at the archive, pre-processes the file and appends some Meta data to the file and stores at the archive. At the time of verification the TPA uses this Meta data to verify the integrity of the data. It is important to note that our proof of data integrity protocol just checks the integrity of data. But the data can be stored, that is duplicated at redundant data centers to prevent the data loss from natural calamities. If the data has to be modified which involves updating, insertion and deletion of data at the client side, it requires an additional encryption of fewer data bits. So this scheme supports dynamic behavior of data.

4. PROBLEM STATEMENT

The cryptographic storage system that enables secure file sharing on un-trusted servers. By dividing files into file groups and encrypting each file group with a unique file-block key, the data owner can share the file groups with others through delivering the corresponding lockbox key, where the lockbox key is used to encrypt the file-block keys. However, it brings about a heavy key distribution overhead for large-scale file sharing. Additionally, the file-block key needs to be updated and distributed again for a user revocation.

System Model

It consists of four modules,

- Cloud Module.
- Proxy Server Module.
- Group Member Module.
- 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.

This essentially means that the owner (client) of the data moves its data to a third party cloud storage server which are supposed to presumably for a fee truly store the data with it and provide it back to the owner whenever required. The cloud server provides privilege to generate secure multi-owner data sharing scheme called MONA. It denotes that any user in the group can securely share data with others by the cloud. This scheme is able to support dynamic groups comfortably. Respectively, new granted users can directly decrypt data files uploaded before their participation without contacting with data owners but within the group.

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 become 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. 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.

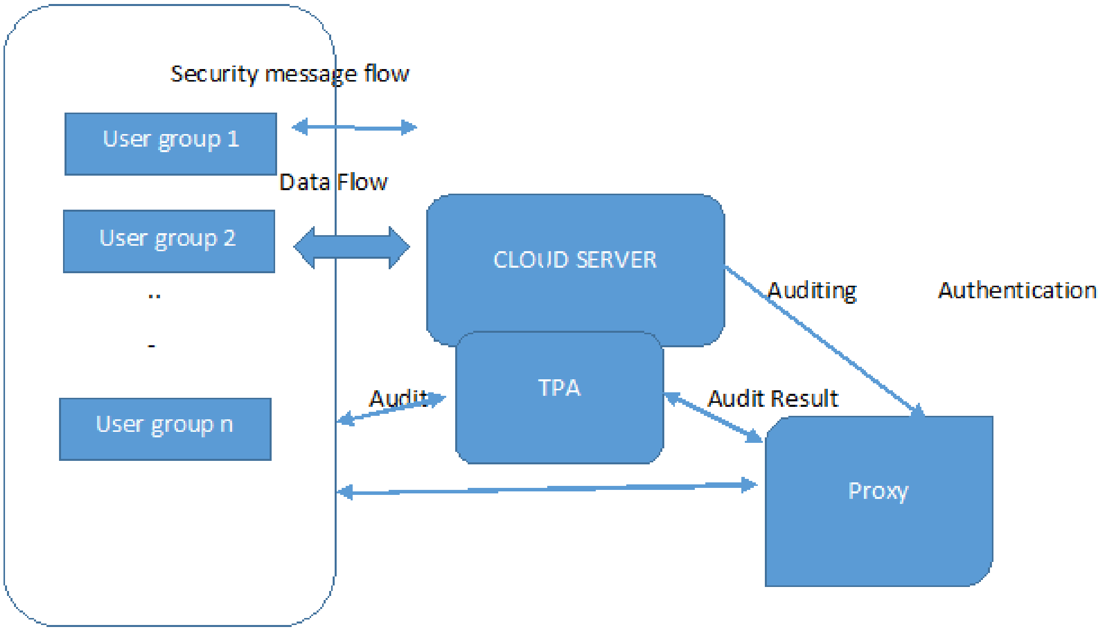

Fig. 1 Cloud Regeneration Architecture

Considering that the data owner cannot always stay online in practice, 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. It generates signature using OAEP based key delegation which provides unique private and public key for each group registered in the cloud. So the users can access the document provided by its own group only. The users can view other group’s document using private key of the other groups. If he modifies other group content he will be revoked by the cloud server.

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. Also each group will have private key and public key in it. The public key is used for viewing the document in the cloud whereas the private is the meant for providing modification rights for a user.

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. The user could never enter his login again. This function will be performed by the cloud server.

Merkle Hash Tree based Archiving:
This technique tries to verify a proof that the data stored by a user at cloud is not modified and thereby the integrity of the data is assured. Cloud archive is not defrauding the owner, if cheating, in this context, means that the storage archive might delete some of the data or may magnify some of the data. While developing proofs for data possession at untrusted cloud storage servers we are often defined by the resources at the cloud server as well as at the client. In this scheme, unlike in the key hash way scheme, only a single key can be used irrespective of the size of the file or the number of files whose retrievability it is want to verify. Also the archive needs to access only a small portion of the file F unlike in the key has scheme which required the annals to process the entire file F for each protocol verification. If the prover had magnify or deleted a substantial allocation of F, then with high probability it will also have suppressed a number of sentinels.

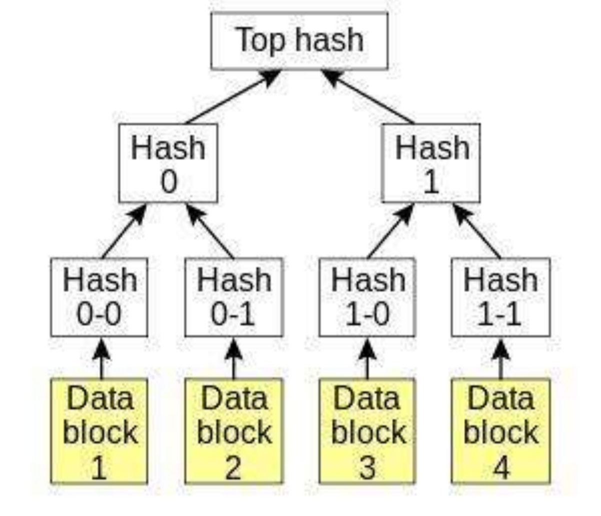

Fig 2 A Merkle Hash Tree

The users can view other group's document using private key of the other groups. If he modifies in this scheme, unlike in the key hash way scheme, only a single key can be used irrespective of the size of the file or the number of files whose retrievability it is want to verify. Also the archive needs to access only a small portion of the file F unlike in the key has scheme which required the annals to process the entire file F for each protocol verification. If the prover had magnify or deleted a substantial allocation of F, then with high probability it will also have suppressed a number of sentinels.

5. CONCLUSION

In this paper, we propose a public investigating scheme for the regenerating-code-based 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 in the beginning rather than applying the blind technique during the auditing process. Considering that the data owner cannot always stay online in practice, 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, we mapping our authenticator based on the BLS signature. This authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. Extensive analysis shows 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.

6. REFERENCES

[1] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data regeneration scheme for cloud storage," in Technical Report, 2013.
[2] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from regenerate files in a serverless distributed file system." in ICDCS, 2002, pp. 617– 624.
[3] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-regeneration," in Proc. of USENIX LISA, 2010.
[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Serveraided encryption for deregenerated storage," in USENIX Security Symposium, 2013.
[5] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology: Proceedings of CRYPTO '84, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds. Springer-Verlag Berlin/Heidelberg, 1985, vol. 196, pp. 242–268.
[6] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure regeneration with efficient and reliable convergent key management," in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol. 25(6), pp. 1615–1625.
[7] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems." in ACM Conference on Computer and Communications Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds. ACM, 2011, pp. 491–500.
[8] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "R-admad: High reliability provision for large-scale de-regeneration archival storage systems," in Proceedings of the 23rd international conference on Supercomputing, pp. 370–379.
[9] M. Li, C. Qin, P. P. C. Lee, and J. Li, "Convergent dispersal: Toward storage-efficient security in a cloud-of-clouds," in The 6 th USENIX Workshop on Hot Topics in Storage and File Systems, 2014.
[10] J. S. Plank and L. Xu, "Optimizing Cauchy Reed-solomon Codes for fault-tolerant network storage applications," in NCA- 06: 5 th IEEE International Symposium on Network Computing Applications, Cambridge, MA, July 2006.
[11] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Regeneration in cloud.
[12] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2," University of Tennessee, Tech. Rep. CS-08-627, August 2008.
[13] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," Journal of the ACM, vol. 36, no. 2, pp. 335–348, Apr. 1989.
[14] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

BIOGRAPHIES

Author Photo-1	Description about the author1 (in 5-6 lines)
Author Photo-2	Description about the author2 (in 5-6 lines)
Author Photo-3	Description about the author3 (in 5-6 lines)

Author Photo-4	Description about the author3 (in 5-6 lines)