ISSN No. 0976-5697
Volume 7, No. 2, March-April 2016
International Journal of Advanced Research in Computer Science
CASE STUDY AND REPORT
Available Online at www.ijarcs.info

Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage

I.Lavanya[1], B.Janaki[2], [3]G.SureshGopi, [4]J.Suresh
[1, 2, 3,4]Pursuing B.Tech (CSE) from St. Ann's College of Engineering & Technology Chirala, Andhra Pradesh - 523 187 India
[5]T.Y.Srinivasa Rao, Associate Professor (CSE) St. Ann's College of Engineering & Technology Chirala, Andhra Pradesh - 523 187 India
[6]Dr P.Harini , Prof & HOD(CSE) St. Ann's College of Engineering & Technology Chirala, Andhra Pradesh - 523 187 India

Abstract: 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. The proposed system, Privacy preserving public auditing scheme for the regenerating-code-based cloud storage is to solve the regeneration problem of failed authenticators in the absence of data owners with the help of proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Moreover, the design consists of a novel public verifiable authenticator, which is generated by a couple of keys and can be regenerated using partial keys. Thus, the scheme can completely release data owners from online burden. In addition, the randomized encode coefficients with a pseudorandom function is to preserve data privacy. Extensive security analysis shows that this scheme is provable secure under random oracle model and experimental evaluation indicates that this 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, Proxy, Provable secure.

I. INTRODUCTION

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., [1]. Nevertheless, this new paradigm of data hosting service also brings new security threats toward users data, thus making individuals or enterprisers 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 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.

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. [2] and Juels *et al* [3], respectively. Considering that files are usually striped and redundantly stored across multi-servers or multi-clouds, [4]–[10] explore integrity verification schemes suitable for such multi-servers or multiclouds setting with different redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes.

In this paper, the focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy [11]. Similar studies have been performed by Bo Chen et al. [7] and H. Chen el al. [8] separately and independently. [7] extended the single-server CPOR scheme(private version in [12]) to the regenerating code-scenario; [8] designed and implemented a data integrity protection(DIP) scheme for FMSR [13]-based cloud storage and the scheme is adapted to the thin-cloud setting. However, both of them 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 [14]. The overhead of using cloud storage should be minimized as much as possible such that a user does not need to perform too many operations to their outsourced data (in additional to retrieving it) [15]. In particular, users may not want to go through the complexity in verifying and reparation. The auditing schemes in [7], [8] imply the problem that users need to always stay online, which may impede its adoption in practice, especially for long-term archival storage.

To fully ensure the data integrity and save the users computation resources as well as online burden, the proposed system privacy-preserving 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 [12] to the multi-server setting, the design consists of a novel authenticator, which is more appropriate for regenerating codes. Besides, the data owner "encrypt" the coefficients to protect data privacy against the auditor, which is more lightweight than applying the proof blind technique in [14], [15] and data blind method in [16].

Storage Privacy: Storage on the public cloud is subject to five privacy requirements.

- *Public Auditability*: 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 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.

1. PROPOSED MODEL

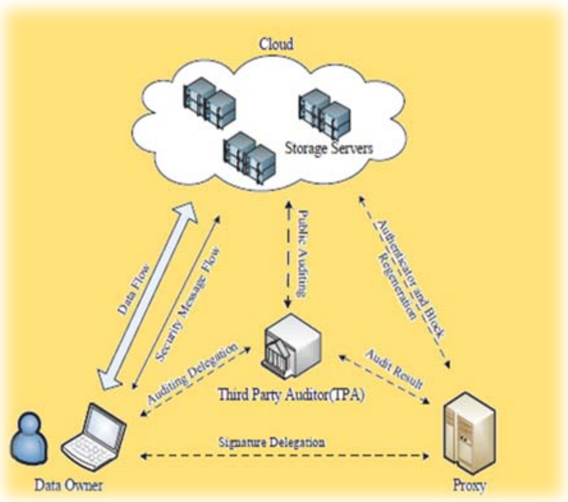

Fig 1: Proposed system Architecture

Security is provided by aggregating the multiple keys into a single key. By this it is very easy to manage the keys and provide security to the users by using aggregated based encryption. The focus is only on AES with verifiable outsourced decryption. The same approach applies to AES with verifiable outsourced decryption. To assess the performance of our AES scheme with verifiable outsourced decryption, this project is implemented with the AES scheme with verifiable outsourced decryption.

II. RESULTS & DESCRIPTION

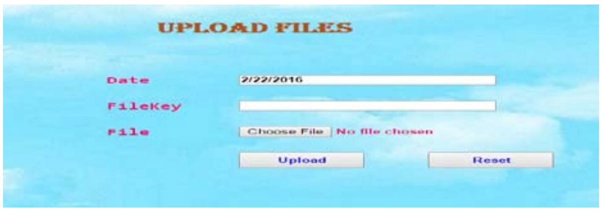

Fig 2: Owner Uploading File

Fig 3: Owner Sending File to Auditor

Fig 4: Auditor Sending File to Cloud after Decrypting

Fig 5: User Searching the Files Present in Cloud

Fig 6: User Requesting for File Key to Proxy

VIEW USER REQUEST

File ID	File Name	Owner Name	File Key	User request Status	Proxy Request Status	Reject Request
27	c lang.docx	bjanaki94@gmail.com	8787	YES	Accept	Reject
31	cloud.docx	bjanaki94@gmail.com	7893	YES	Accept	Reject
30	Memory.docx	bjanaki94@gmail.com	325	YES	Accept	Reject
31	cloud.docx	suresh@gmail.com	7893	YES	Accept	Reject
30	Memory.docx	suresh@gmail.com	325	YES	Accept	Reject
34	edc.docx	saikumar@gmail.com	89455	YES	Accept	Reject
28	storage.docx	suresh@gmail.com	5692	YES	Accept	Reject
31	cloud.docx	saikumar@gmail.com	7893	YES	Accept	Reject

Fig 7: Proxy View requests of Users

VIEW USER REQUEST

File ID	File Name	Owner Name	File Key
27	c lang.docx	bjanaki94@gmail.com	887285
27	c lang.docx	bjanaki94@gmail.com	635858
31	cloud.docx	bjanaki94@gmail.com	910367

Fig 8: User View Response from Proxy

III. CONCLUSION & FUTURE WORK

The proposed system Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage system where the data owners are privileged to delegate TPA for their data validity checking. Considering that the data owner cannot always stay online practically, in order to keep the storage available and verifiable after a malicious corruption, semi-trusted proxy is introduced into the system model and provide a privilege for the proxy to handle the keys to users based on user requests by providing different keys for different users.

TPA can upload data to cloud without intimation to data owner. This can be extended to notify the owner with an email that the file is uploaded to cloud. The data in the files that are present in the cloud are only viewed by the users but they cannot do any modifications. So this project can be extended to make any modifications by user and send intimation to data owner.

IV. REFERENCES

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Computer Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, p. 13, 2009.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th 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 large files," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 584–597.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: Multiplereplica provable data possession," in Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on. IEEE, 2008, pp. 411–420.

[5] K. D. Bowers, A. Juels, and A. Opera, "Hail: a high-availability and integrity layer for cloud storage," in Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009, pp. 187–198.

[6] 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 and System Sciences, vol. 78, no. 5, pp. 1345–1358, 2012.

[7] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in Proceedings of the 2010 ACM workshop on Cloud computing security workshop. ACM, 2010, pp. 31–42.

[8] H. Chen and P. Lee, "Enabling data integrity protection in regenerating coding-based cloud storage: Theory and implementation," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 2, pp. 407–416, Feb 2014.

[9] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 9, pp. 1717–1726, 2013.

[10] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 12, pp. 2231– 2244, 2012.

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

[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in Advances in Cryptology-ASIACRYPT 2008. Springer, 2008, pp. 90– 107.

[13] Y. Hu, H. C. Chen, P. P. Lee, and Y. Tang, "Nccloud: Applying network coding for the storage repair in a cloud-of-clouds," in USENIX FAST, 2012.

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

[15] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," Computers, IEEE Transactions on, vol. 62, no. 2, pp. 362–375, 2013. IEEE TRANSACTIONS ON INFORMATION AND SECURITY Vol 1 No 2015

[16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards secure and dependable storage services in cloud computing," Service Computing, IEEE Transactions on, vol. 5, no. 2, pp. 220–232, May 2012.