International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

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

**S. Vaishnavee, G. Murali Krishnan, P. Ragul**
Computer Science and engineering, Sri Krishna College of Technology, Coimbatore, Tamilnadu, India

## ABSTRACT

To prevent 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 gain popularity due to their lower repair bandwidth while providing fault tolerance. Existing remote checking methods for regenerating-coded data only provide private auditing, required 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 regenerating problem of failed authenticators in the absence of data owners, we introduce a proxy, which is privileged to regenerate the authenticators, to 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. our scheme can completely avoid data owners from online burden. In addition, we randomize the encode coefficients with a pseudorandom function to preserve data privacy. Analysis shows that our scheme is 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, Auditing, TPA, Multi-Servers, Multi-Clouds, Regenerating-Code-Based Cloud Storage

## I. INTRODUCTION

Cloud storage is 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 the location independence, and the avoidance of capital expenditure on hardware, software, and personal maintenances, etc., Nevertheless, this new paradigm of data hosting service also bring new security threats toward users data, thus making individuals or enterprisers still feel hesitant. It is noted that data owners loss control over the fate of their outsourced data thus, the correctness, availability and integrity of the data are being at risk. On the one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would corrupt user 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 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 maintains their data correctly.

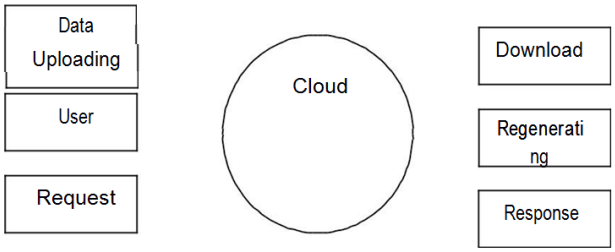
## II. METHODS AND MATERIAL

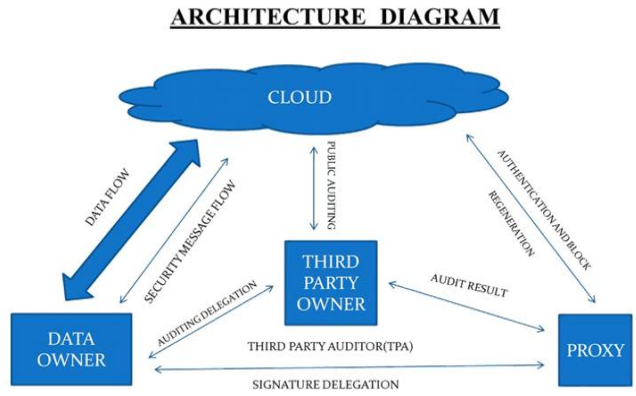
### 1. Auditing

The third party auditor (TPA), who has ability 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. 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, 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. When the auditor encounters any corruption in the servers, it sends an alert to the data owner and the proxy agent.

## 2. Regenerating Code

A proxy agent acts on behalf of the data owner to regenerate the authenticators and data blocks on the failed servers. Notice that the data owner is restricted in computational and storage resources compared to other entities may become off-line even after the data upload procedure. The proxy, who would always be online, is supposed to be much more powerful, then the data owner is lesser 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 restart to the TPA for integrity verification and delegate the reparation to the proxy. When the proxy agent gets alert from the TPA, proxy agent will regenerate the code that is corrupted in the server.




## 3. Existing System

Existing remote checking methods for regenerating-code data only provide private auditing, requiring data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical. 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 is the PDP (*provable data possession*) model and POR (*proof of retrievability*) model, which were originally proposed for the **single-server** scenario. Single-server CPOR scheme to the regenerating code-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. However, both of them are designed to private audit, only the data owner is allowed to verify the integrity and repair the faulty servers. The overhead of using cloud storage should be minimized as much as possible such that a user does

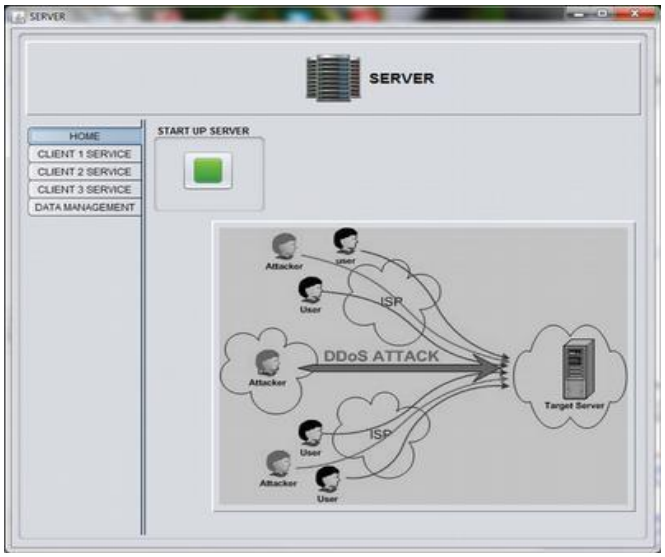
In this system, we propose a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owner, we introduce a proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Considering that files are usually striped and redundantly stored through 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 and erasure codes*, and more recently *regenerating codes*. We focus on the integrity verification problem in **regenerating-code-based cloud storage**, especially with the functional repair strategy. To fully ensure that 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 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 direct adapting the existing public auditing scheme to the multi-server, we design a novel authenticator, which is more appropriate for regenerating codes. we ***encrypt*** the coefficients to protect data privacy against the auditor, which is more lightweight than applying the proof of blind technique and data blind method.

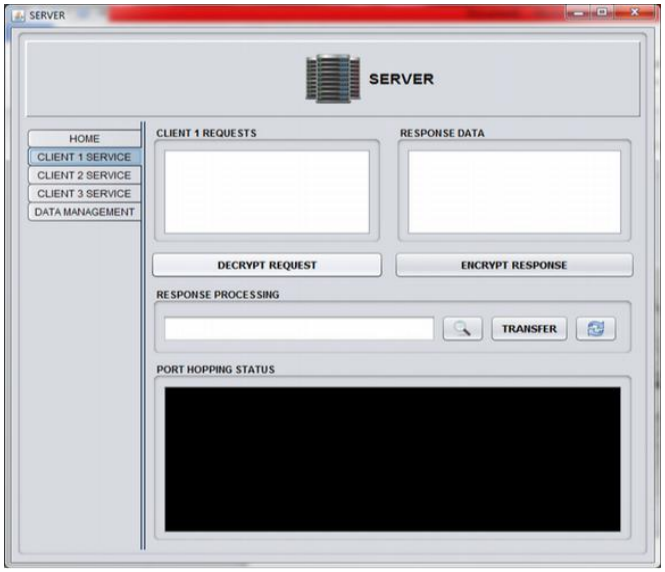
# III. RESULTS AND DISCUSSION

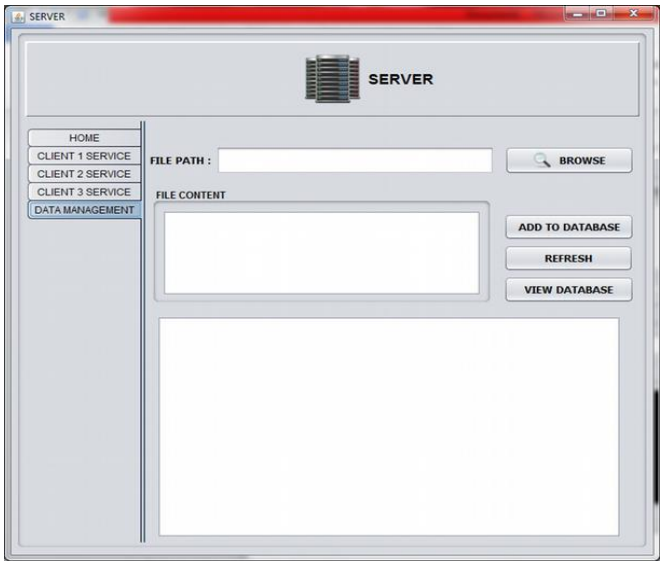
**Modules**

- ✓ Data Owner
- ✓ Cloud Server
- ✓ Auditing
- ✓ Regenerating Code


**Proposed System**




## IV.CONCLUSION

A public auditing skill or RCB storage system where data owners are privileged to delegate TPA for validating checking .To product the original data privacy against TPA we randomize the coefficient. A semi trusted proxy handling reparation of coded blocks and authenticators is implemented. Authenticators are based on AES algorithm and analysis show that scheme is efficient and can be integrated RCB storage system.

## V. REFERENCES

[1] 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.

[2] S. G. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy preserving public auditing scheme for cloud storage," Comput. Elect. Eng., vol. 40, no. 5, pp. 1703–1713, 2013..

[3] M. Armbrust et al., "Above the clouds: A Berkeley view of cloud computing," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.

[4] H. C. H. Chen and P. P. C. Lee, "Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 407–416, Feb. 2014.

[5] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptol., vol. 17, no. 4, pp. 297–319, 2004.

[6] 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.

[7] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

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