Framework for Secure and Dynamic Auditing for Regenerating Code Based Data Storage in Cloud Platform

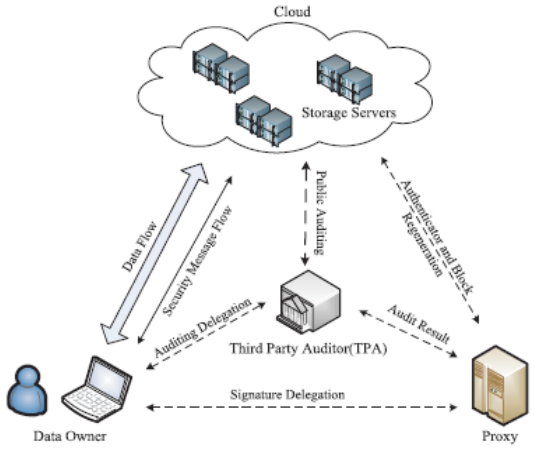
Dr.V.Goutham [1], J.Rachana [2], M.Nikhila[3]
[1,2,3]*Department of Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Meerpet, Telangana, India*

Abstract— To defend outsthiscing data in cloud storage in contradiction of venalities, adding fault tolerance to cloud storage organized with data integrity testing and let-down reparation befits precarious. Restoring codes have extended popularity due to their lower repair bandwidth while providing fault tolerance. Current remote checking methods are presented for regenerating-coded data only provide private auditing, requiring data owners to always stay online and handle auditing. A public auditing system for the regenerating-code-based cloud storage is projected to elucidate the regeneration problem of unsuccessful authenticators in the absence of data owners. A unique public verifiable authenticator, which is produced by a couple of keys and can be regenerated using partial keys so that it can totally release data owners from online burden. This scheme is highly proficient and can be practicably combined into the regenerating-code-based cloud storage

Index Terms— Cloud storage, regenerating codes, public audit, privacy preserving, authenticator regeneration, proxy, privileged, provable secure.

1 INTRODUCTION

CLOUD storage is now gaining popularity because it offers a flexible on-demand data outsthiscing service with alluring benefits. Some of them were, reprieve of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personal maintenances[1].Nonetheless, this new prototype of data hosting service also brings new security threats toward users data, thus making individuals or enterprisers still feel cautious. Many mechanisms dealing with the integrity of outsthisced 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 and Kaliski [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 **multi-clouds** setting with different redundancy schemes, such as *replication*, *erasure codes*, and, more recently, *regenerating codes.* In this paper, we 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 Chen *et al.* [7] and Chen and Lee [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.1 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 outsthisced data and the user's constrained resthisce 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 outsthisced 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] and [8] imply the problem that users need to always stay online, which may impede its embracing in practice, especially for long-term archival storage.



Let us have a glance on the following aspects:

• A novel homomorphic authenticator based on BLS signature [17], which can be generated by a couple of secret keys and verified publicly is designed. 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.

• This 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) dur-

ing 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.

• This 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 this 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.

2 RELATED WORK

The problem of remote data checking for integrity was first proposed in [16] and [20]. Then Ateniese et al. [2] and Juels and Kaliski [3] gave rise to the same notions provable data possession (PDP) and proof of retrievability (POR), respectively. Ateniese et al. [2] proposed a formal definition of the PDP model for ensuring possession of files on untrusted storage, introduced the concept of RSA based homomorphic tags and adviced randomly sampling a some blocks of the file. In their subsequent work [18], they presented a dynamic version of the prior PDP scheme based on MAC, which permits very basic block operations with limited functionality but block insertions. At the same time, Erway et al. [9] gave a formal framework for dynamic PDP and provided the first fully dynamic solution to support provable updates to stored data using rankbased authenticated skit lists and RSA trees. To improve the efficiency of dynamic PDP, Wang et al. [20] Presented a new method which uses merkle hash tree to support fully dynamic data. To release the data owner from online burden for verification, [2] considered the public auditability in the PDP model for the first time. However, their variant protocol exposes the linear combination of samples and thus gives no data privacy guarantee. Then Wang et al. [14], [15] proposed a random blind technique to address that problem in their BLS signature based public auditing scheme. Similarly, Worku et al. [11] introduced another privacy-preserving method, which is more efficient since it avoids involving a computationally intensive pairing operation for the sake of data blinding. Yang and Jia [9] presented a public PDP scheme, where the data privacy is provided through combining the cryptography method with the bilinearity property of bilinear pairing. [16] used random mask to blind data blocks in error-correcting coded data for privacy preserving auditing with TPA. Zhu et al. [10] presented a formal framework for interactive provable data possession (IPDP) and a zero-knowledge IPDP solution for private clouds. Their ZK-IPDP protocol supports fully data dynamics, public verifiability and is also privacy preserving against the verifiers. 7) 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 this 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 this scheme is provable secure, and the performance evaluation shows that this scheme is highly efficient and can be feasibly integrated into a regenerating-code-based cloud storage system

3 SYSTEM MODEL

The auditing system model for Regenerating- Code-based cloud storage, which includes of this entities: *the data owner*, who owns large quantities 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 resthisces; *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 neutral for both data owners and cloud servers; and *a proxy agent*, who is semi-trusted and acts depending on 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 resthisces 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 resthisces 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. Compared with the traditional public auditing system model, this system model involves an additional proxy agent.

3.1 Design Goals

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

• *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 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 quick ly indicated when data corruption is detected.

4 Proposed System: Construction of This Auditing Scheme

To make the contribution and goals easier to understand, consider the reference scenario: The staffs (i.e., cloud users) first generate their public and private keys, and then delegate the authenticator regeneration to a proxy (a cluster or powerful workstation provided by the company) by sharing partial private key. After producing encoded blocks and authenticators, the staffs upload and distribute them to the cloud servers. Since that the staffs will be frequently off-line, the company employs a trust third party (the TPA) to interact with the cloud and perform periodical verification on the staffs' data blocks in a sampling mode. Once some data corruption is detected, the proxy is informed, it will act on behalf of the staffs to regenerate the data blocks as well as corresponding authenticators in a secure approach. So we could see that this scheme guarantees that the staffs can use the regenerating-code-based cloud in a practical and lightweight way, which completely releases the staffs from online burden for data auditing and reparation.

4.1 Enabling Privacy-Preserving Auditable

The privacy protection of the owner's data can be easily achieved through integrating with the random proof blind technique [15] or other technique [9]. Though, all these privacy-preservation methods present 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 lightweight, to alleviate 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 $\{w_i\}_{mi=1}$ with random coefficients. Supposing that the curious TPA has recovered m coded blocks by elaborately performing *Challenge-Response* procedures and solving systems of linear equations [14], the TPA still requires to solve another group of m linearly independent equations
to derive the m native blocks. We can utilize a keyed pseudo-random function mask the coding coefficients and thus prevent the TPA from correctly obtaining the original data. Specifically, the data owner maintains a secret key in the beginning of the **Setup** procedure and augments m original data blocks.

4.2 Mitigating the Overhead of Data Owner

Despite that the data owner has been released from online burden for auditing and repairing, it still makes sense to reduce its computation overhead in the **Setup** phase because data owners usually maintain very limited computational and memory resthisces. As previously described, authenticators are generated in a new method which can reduce the computational complexity of the owner to some extent; however, there exists a much more efficient method to introduce further reduction.

4.3 A Tradeoff between Storage and Communication

In this auditing scheme described above, we assume that each segment contains only one symbol for simplicity and is accompanied by an authenticator of equal length. This approach gives storage overhead twice as much as the length of the data block and the server's response.

4.4 Detection Probability

The TPA performs random spot checking on each coded block to improve the efficiency of this auditing scheme, while still achieving detection of faulty servers with high probability.

5 Experimental evaluation

we evaluate the efficiency of this privacy-preserving method and the results shows that this design is perfectly lightweight for the data owner to execute. Because this privacy-preservation method is implemented only once during the whole life of a user's file, while the random blind process in [9] and [15] would be performed in each Audit instance, apparently this scheme is much more efficient and thus we do not experimentally compare their performance. In addition, [16] introduced a similar privacy preserve method for their public auditing protocol. Both their scheme and this utilize random mask in linear-code-based distributed storage to avoid the auditor getting enough information for original data retrieve. However, there is one significant difference between the two, *i.e.*, [16]'s method used the PRF to blind the data blocks before the encoding process, while this scheme choose to mask the coding coefficients instead. Numerically comparing them, we can see that they need to execute PRF for $s \cdot m$ times and m times during the **Setup** phase, separately. Thus this privacy preserve method is more efficient and effective than [16].

5.1 Audit Computational Complexity

…vers are usually powerful in … analyzing the computational … omit those on the cloud side. …aggregated proof requires less …e most expensive operation …ntiations and multiplication,

TABLE III
COMPUTATIONAL COST INTRODUCED BY THE PRIVACY-PRESERVING METHOD

	Without privacy preserving	With privacy preserving
$s = 60$	7966 ms	7994 ms
$s = 80$	10642 ms	10653 ms

TABLE III
COMPUTATIONAL COST INTRODUCED BY THE PRIVACY-PRESERVING METHOD

	Without privacy preserving	With privacy preserving
$s = 60$	7966 ms	7994 ms
$s = 80$	10642 ms	10653 ms
$s = 100$	1329 6ms	13301 ms

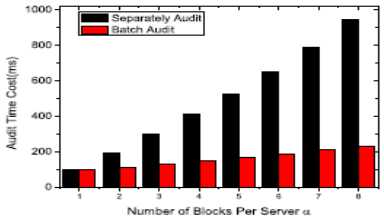


Fig. 6. Time for Audit with different α.

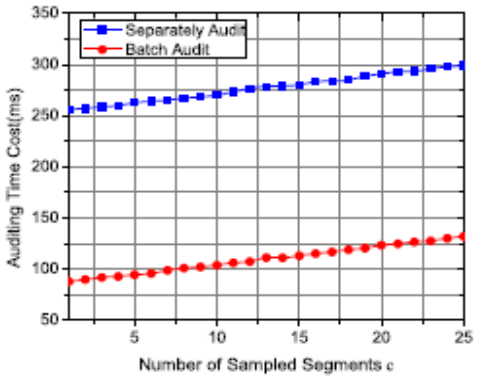


Fig. 7. Time for Audit with different *c*.

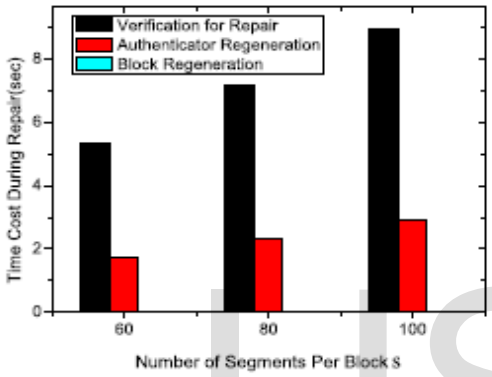
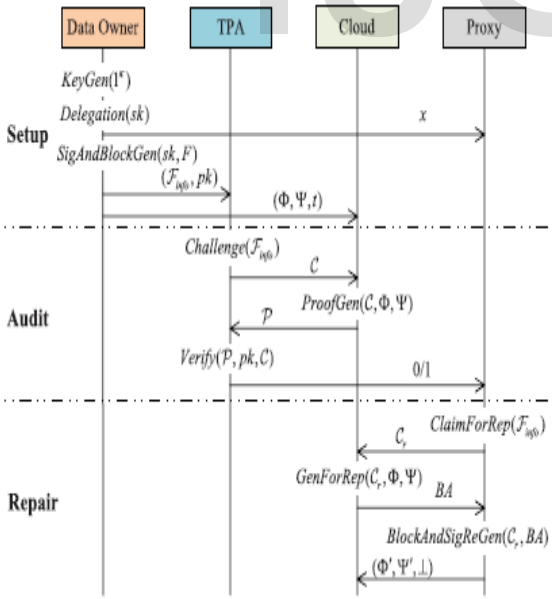


Fig. 8. Time for Repair with different *s*.



5.2 Repair Computational Complexity

The regeneration of the faulty blocks and authenticators is delegated to a proxy in this auditing scheme. Verification for Repair, Regeneration for Blocks and Regeneration for Authenticators, thus obtaining the results. Obviously, it takes the proxy much more time to verify the received blocks for repair, less to regenerate the authenticators, and negligible to regenerate the faulty blocks. This situation occurs mainly because there are a mass of expensive pairing operations and less expensive modular exponentiations during the verification, thus leading to the most time consuming sub-process.

6 CONCLUSION

A public auditing scheme for the regenerating-code-based cloud storage system, where the data owners are advantaged to represent TPA for their data validity checking. To protect the original data privacy against the TPA, we randomize the coefficients in the establishment rather than applying the blind technique through the auditing process. To restore appropriate for the regenerating-code-scenario, authenticator based on the BLS signature is designed. This authenticator can be capably produced by the data owner concurrently with the encoding process.

REFERENCES

1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang, "NCCloud: Applying network coding for the storage repair in a cloud-of-clouds," in *Proc. USENIX FAST*, 2012, p. 21.

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

[15] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[16] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Trans. Service Comput.*, vol. 5, no. 2, pp. 220–232, Apr./Jun. 2012.

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

[18] 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.

[19] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[20] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Public Key Cryptography*. Berlin, Germany: Springer-Verlag, 2009, pp. 68–87.

[21] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2001, pp. 213–229.

[22] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Trans. Fundam. Electron., Commun., Comput. Sci.*, vol. E84-A, no. 5, pp. 1234–1243, 2001.

[23] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, "Secure network coding over the integers," in *Public Key Cryptography*. Berlin, Germany: Springer-Verlag, 2010, pp. 142–160.

[24] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.

AUTHORS

[1] Dr V. Goutham is a Professor and Head of the Department of Computer Science And Engineering at Teegala Krishna Reddy Engineering College affiliated to J.N.T.U Hyderabad. He received Ph.d from Acharya Nagarjuna University, M.Tech from Andhra University and B.Tech from J.N.T.U Hyderabad. He worked for various MNC Companies in Software Testing and Quality as Senior Test Engineer. His research interests are Software Reliability Engineering, software testing, software Metrics, and cloud computing.

[2] Mrs.J.Rachana is working as a Assistant Professor in the Department of Computer Science And Engineering at Teegala Krishna Reddy Engineering College affiliated to J.N.T.U Hyderabad.

[3] Ms. M.Nikhila Department of Computer Science And Engineering at Teegala Krishna Reddy Engineering College affiliated to J.N.T.U Hyderabad.