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

Ainampudi Lavanya Satya , Dr.I.R.Krishnam Raju , Sri.V.Bhaskara Murthy
MCA Student, Professor, Associate Professor
Dept Of MCA
B.V.Raju College, Bhimavaram

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. 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. 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 regeneratingcode- based cloud storage.

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

Data 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, 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 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 regeneratingcode- 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 setting1. 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 usersneed 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, we propose a public auditing scheme for the regenerating-code-based cloud storage, in which the integrity checking and regeneration (of failed 1Indicating that the cloud servers are only provided with the RESTful interface.

- We design a novel homomorphic authenticator based on BLS signature [17], which can be generated by a couple of secret keys and verified publicly. 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 regeneratingcode-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 illustrated in Section II-C.
- Moreover, we make a comparison with the state of the art and experimentally evaluate the performance of our scheme.

The rest of this paper is organized as follows: Section II introduces some preliminaries, the system model, threat model, design goals and formal definition of our auditing scheme. Then we provide the detailed description of our scheme in Section III; Section IV analyzes its security and Section V evaluates its performance. Section VI presents a review of the related work on the auditing schemes in cloud storage. Finally, we conclude this paper in Section VII.

V. EXISTING APPROACH

Existing framework outlined and actualized an information trustworthiness assurance (DIP) plan for FMSR [6]-based distributed storage and the plan is adjusted to the meager cloud setting. Be that as it may, the two are intended for

private review, just the information owner is permitted to check the trustworthiness and repair the faultyservers. Considering the expansive size of the outsourced information and the client's obliged asset ability, the undertakings of examining and reparation in the cloud can be impressive and costly for the clients [7]. The overhead of utilizing distributed storage ought to be reduced however much as could reasonably be expected such that a client does not have to perform an excess of operations to their outsourced information. [8] Specifically, clients might not have any desire to experience the unpredictability in checking and reparation. The reviewing plans in [9] and [1] suggest the issue that clients need to continuously stay on the web, which might hinder its selection by and by, particularly for long haul authentic capacity.

VI. PROPOSED APPROACH

Proposed framework use Elliptic bends to build people in general key cryptography framework. The key size for this calculation is little henceforth information transmission required less data transfer capacity and time .Public-key cryptography depends on the obstinacy of certain numerical issues. Early open key frameworks, for example, the RSA calculation, are secure expecting that it is hard to figure a huge whole number made out of two or all the more substantial prime elements. For elliptic-bend based conventions, it is expected that finding the discrete logarithm of an arbitrary elliptic bend component concerning an openly known base point is infeasible. The measure of the elliptic bend decides the trouble of the issue. It is trusted that the same level of security managed by a RSA-based framework with an extensive modulus can be accomplished with a much littler elliptic bend bunch. Utilizing a little gathering lessens capacity and transmission necessities. For current cryptographic purposes, an elliptic bend is a plane bend which comprises of the focuses fulfilling the mathematical statement.

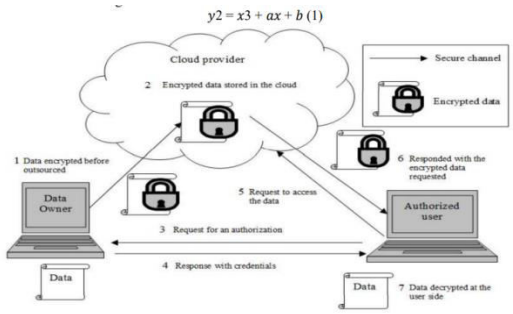
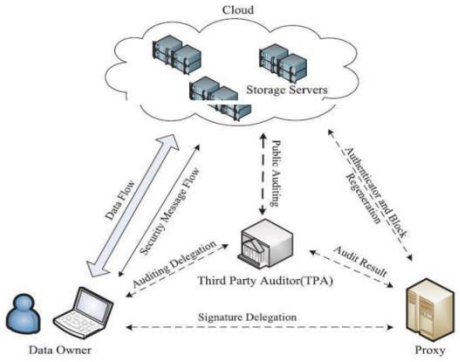


Fig 1. Proposed System Flow

VII. SYSTEM ARCHITECTURE



VIII. MODULE INFORMATION

1. Cloud Server: Which are managed by the cloud service provider, provide storage service and have significant computational resources. Responsibility: 1. Dealing with receive data from Data Owner. 2. Store the Data in encrypted form. 3. Give the Data read permissions to authorized User. 4. Accept and Replacement of Data through Proxy Agent. 2. Data Owner / Cloud Client: Data Owner owns large amounts of data files to be stored in the cloud.Data owner refers to both the possession of and responsibility for information. Data Owner implies power as well as control. The control of information includes not just the ability to access, create, modify, package, derive benefit from, sell or remove data, but also the right to assign these access privileges to others. Responsibility: 1. Use or Data Owner able to Outsource their Data. 2. Encrypt the Data while Outsourcing of it. 3. Delegation between Data Owner and Proxy Agent. 4.Generate ask secrete key and Assign to the corresponding Authenticators present in PA. 5. Data User able see data Stored on cloud Server and can make

request to the Data or file. 3. Third Party Auditor: TPA has expertise 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. Responsibility: 1. Examining the outsourced data and data owner Data to ensure the Data Integrity. 2. Public Auditing by checking h(.) code. 3. Send Acknowledgement to the Proxy for decision making.

VII. CONCLUSION

In this paper, we propose 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 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 practise, 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 design 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.

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 Comput. 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. Oprea, "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 geographicallydispersed 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 regeneratingcoding- 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, andW. Lou, “Privacy-preserving public auditing for secure cloud storage,” *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, 2013.