International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Vol.05,Issue.01,
January-2017,
Pages:0085-0088

Maintaining Secrecy and Public Integrity of Cloud Data using Regenerating Codes

B. SRINIVAS[1], SHAIK FAYAZ[2]
[1]Research Scholar, Eswar College of Engineering, Narasaraopet, Guntur(Dt), AP, India.
[2]Assistant Professor & Research Supervisor, Eswar College of Engineering, Narasaraopet, Guntur(Dt), AP, India.

Abstract: To provide security for the outsourced data in cloud storage against various problems and provide data integrity becomes difficult. Fault tolerance is also important issue for protecting data in the cloud. Now a days regenerating codes got importance because of their lower repair bandwidth while providing fault tolerance. Previous remote checking methods for regenerating coded data only provide private auditing, requiring data holders to always keep online and handle auditing, as well as repairing, which is sometimes difficult. In this paper we are going to propose a public auditing scheme for the regenerating code based cloud storage. To obtain solution for regeneration problem of failed authenticators in the absence of data holders, we make a proxy, which is privileged to regenerate the authenticators, in the traditional public auditing system model. We also design a novel public verifiable authenticator, which is made by some keys. Thus, this scheme can almost release data holders from online burden. We also randomize the encode coefficients with a pseudorandom function to sure data privacy. Extensive security analysis shows this scheme is secure and provable under random oracle model. Experimental evaluation model indicates that this scheme is highly efficient and can be feasibly integrated into the regenerating cloud based storage.

Keywords: Cloud Storage, Regenerating Codes, Public Auditing, Privacy Preserving, Proxy.

I. INTRODUCTION

Cloud storage got importance because of various benefits: relief of the burden for storage management, open access with location independence, and avoidance of capital expenditure on hardware, software, and personal maintenance, etc. Sometimes data owners lose their control over the fate of their outsourced data; thus, the correctness, availability and integrity of the data are being put at risk. Sometimes the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt users' data; and sometimes 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. Thus it is useful for users to implement an efficient protocol to perform periodical verifications of their outsourced data to ensure data integrity. Some mechanisms dealing with the integrity of outsourced data without a local copy have been proposed under various system and security models up to now. The most important work from 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. Imagine 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 various redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes.

In this paper, we concentrate on the integrity verification problem in regenerating-code-based cloud storage, specialy with the functional repair strategy [11]. Similar studies have been performed by Chen et al. [7] and Chen and Lee [8] individually. [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 check the integrity and repair the damaged servers. Considering the huge 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 costly 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 so many operations to their outsourced data (in additional to retrieving it) [15]. In particular, users may not want to go through the difficulties 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 adoption in practice, specially for long-term archival storage. To fully ensure the data integrity and save the users' computation resources as well as online burden, we propose public auditing scheme for the regenerating-code-based cloud storage, in which the 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 directly applying the old public auditing scheme [12] to the multi-server setting, we design a novel authenticator, which is more suitable for regenerating codes.

II. LITERATURE SURVEY

C.Wang, Q.Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," presented privacy-preserving public auditing system for data storage security in Cloud Computing. C. Wang, S. S. M.

Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," proposed that a secure cloud storage system supporting privacy-preserving public auditing. K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," proposed an efficient and inherently secure dynamic auditing protocol. It protects the data privacy against the auditor by combining the cryptography method with the bilinearity property of bilinear paring, rather than using the mask technique. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," Proposed flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server.

In this paper we are going to propose a public auditing scheme for the regenerating code based cloud storage. To obtain solution for regeneration problem of failed authenticators in the absence of data holders, we make a proxy, which is privileged to regenerate the authenticators, in the traditional public auditing system model. We also design a novel public verifiable authenticator, which is made by some keys. Thus, this scheme can almost release data holders from online burden. We also randomize the encode coefficients with a pseudorandom function to sure data privacy. Extensive security analysis shows this scheme is secure and provable under random oracle model. Experimental evaluation model indicates that this scheme is highly efficient and can be feasibly integrated i regenerating cloud based storage.

III. SYSTEM MODEL

We have proposed auditing system model for Regenerating-Code-based cloud storage as Fig.1 [34], which consist of four blocks: data owner which consist of large amount of data stored in the cloud; the cloud, which provides cloud services; provide storage service and have significant computational resources; the third party auditor (TPA) conducts public audits on the coded data in the cloud, its audit results are unbiased for both data owner and cloud servers; and proxy agent, who is semi-trusted and acts on behalf of the data owner to regenerate authenticators and data blocks on the failed servers during the repair procedure. The data owner is restricted in computational and storage resources compared to other entities and may becomes off-line even after the data upload procedure. The proxy is supposed to be much more powerful than the data owner but less than the cloud servers in terms of computation and memory capacity, who would be always online. The periodic auditing and accidental repairing is used to save resources and online burden. The data owners resort to the TPA for integrity verification and delegate the reparation to the proxy. As compare to the traditional public auditing system model, our system model involves an additional proxy agent.

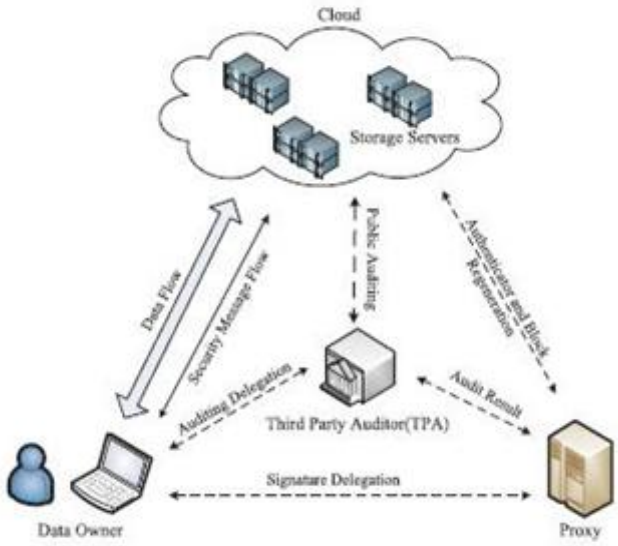

Fig.1. System Model.

In order to reveal the rationality of our design, we consider a scenario: A company employs a commercial regenerating-code-based public cloud and provides long-term archival storage service for its staffs, the staffs are equipped with low end computation devices (e.g., Laptop PC, Tablet PC, etc.) and will be frequently off-line. For public data auditing, the company relies on a trusted third party organization to check the data integrity; Similarly, to release the staffs from heavy online burden for data and authenticator regeneration, the company supply a powerful workstation (or cluster) as the proxy and provide proxy reparation service for the staffs' data.

A. Definitions of Auditing Scheme

Our auditing scheme consists three procedures: Setup, Audit, Repair.

- **Setup:** Data owner used this procedure is to initialize our auditing scheme.
- **Audit:** The cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness in this procedure.
- **Repair:** In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.

B. Design Goals

- Public Auditability: To permit 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 make sure 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 within 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 represented when data corruption is detected.

IV. SECURITY ANALYSIS

A. Correctness

There are two verification process in this scheme, one for spot checking within the Audit phase and another for block integrity checking within the Repair phase.

B. Soundness

We say that our auditing protocol is sound if any cheating server that convinces the verification algorithm that it is storing the coded blocks and corresponding coefficients is actually storing them.

C. Regeneration-Unforgeable

Noting that the semi-trusted proxy handles regeneration of authenticators in our model, we say our authenticator is regeneration-unforgeable.

D. Resistant to Replay Attack

Our public auditing scheme is resistant to replay attack mentioned in [7], since the repaired server maintains identifier η which is different with the corrupted.

V. EVALUATION

A. Comparison

Our proposed mechanism and makes a comparison with another remote data checking schemes[7], [8] for regenerating coding based cloud storage.

B. Performance Analysis

We focus on evaluating the performance of our privacy preserving public audit scheme during the Setup, Audit and Repair procedure.

VI. RELATED WORK

The problem of remote data checking for integrity was first proposed in [26] and [27]. 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 [28], 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. [29] gave a formal framework for dynamic PDP and provided the first fully dynamic solution to support provable updates to stored data using rank-based authenticated skit lists and RSA trees. To improve the efficiency of dynamic PDP, Wang et al. [30] 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. [31] 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.

VII. 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 our 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 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.

VIII. 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 geographically dispersed 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.
[25]P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in Selected Areas in Cryptography. Berlin, Germany: Springer-Verlag, 2006, pp. 319–331.
[26]Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Integrity and Internal Control in Information Systems VI. Berlin, Germany: Springer-Verlag, 2004, pp. 1–11.
[27]D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," Cryptology ePrint Archive, Tech. Rep. 2006/150, 2006. [Online]. Available: http://eprint.iacr.org/
[28]G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw., 2008, Art. ID 9.
[29]C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamicprovable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur., 2009, pp. 213–222.
[30]Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Computer Security. Berlin, Germany: Springer-Verlag, 2009, pp. 355–370.
[31]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.
[32]K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in Proc. ACM Workshop Cloud Comput. Secur., 2009, pp. 43–54.
[33]Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Theory of Cryptography. Berlin, Germany.
[34]Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage," in IEEE Trans. on information forensics and security , vol.. 10,no. 7, July 2015.

Author's Profile:

B.Srinivas is a student pursuing M.Tech (CSE) in Eswar College of Engineering, Narasaraopet, Guntur, India.

Shaik.Fayaz M.Tech(CSE), is having 04+ years of experience in the field of teaching in various Engineering Colleges and 02 years of Software Industrial experience. At present he is working as Asst. Prof. in Eswar College of Engineering, Narasarao-pet, Guntur, India. He published 2 international journals.