International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 12
August 2016

Fault Tolerant finding in Code regeneration and auditing in Cloud storage

M N MALLIKARJUNA REDDY[1] & SRAVANA LAKSHMI DUBBA[2]

[1]ASSISTANT PROFESSOR Dept. of CSE SVR ENGINEERING COLLEGE, AYYALURU, and NANDYAL

Email: - malli51arjun@gmail.com

[2]PG-Scholar Dept. of CSE SVR ENGINEERING COLLEGE, AYYALURU, and NANDYAL

Email: - dsravanireddy57@gmail.com

Abstract

To protect the outsourced data in cloud storage against corruptions and inconsistencies, adding up fault tolerance to cloud storage jointly with data integrity checking and failure reparation becomes necessary. Recently, the regenerating codes have gained significance due to their lower repair bandwidth and due to their fault tolerance capabilities. The Earlier remote checking methods for regenerating coded data only provides personal auditing and requires data owners to always keep online and handle auditing, as well as repairing, which is sometimes not practical. In this paper, we recommend a public auditing scheme for the regenerating-code-based cloud storage. To overcome this regeneration problem of failed authenticators in the lack of data owners, we commence a proxy, which is confidential to regenerate the authenticators, into the usual public auditing system model. Introduction of cloud audit server eliminates the contribution of user in the auditing and in the pre-processing phases. In our approach, the client cannot store any large set of data locally except a secret key which is required for encryption. In contrast with the previous methods, we also avoid the requirement of encrypting complete data at client side thereby saving client computational time. The proposed approach is also applicable for big static data such as video files, audio files and social networking data etc.

Keywords: Code regeneration, public auditing, Cloud storage, Fault Tolerant.

1. Introduction

In recent years, the usage of computers, mobile devices and social sites has become a part of day to day activities. Distribution of information, photographs, video and audio files have permitted user to communicate and utilize effective storage space in the Internet without worrying to purchase physical storage locally. All these data can be stored anywhere in Internet and Cloud is found to be a default choice due to its mobility and transparency.

Cloud storage is now gaining attraction as it offers a flexible on-demand data outsourcing service with interesting benefits, release of the burden for storage management, worldwide data access with location independence, and avoidance of capital expenses on hardware, software, and personal cares, etc., [4]. However, this new paradigm of data hosting service also leads to new security threats in the direction of user's data, thus making individuals or enterprisers to feel quiet hesitant. It is well-known that data owners may lose control over their outsourced data; thus, the accuracy, accessibility and integrity of the data are being put at risk. On the one hand, the cloud service is usually met with a broad range of internal/external challenges, who would unkindly delete, intrude or corrupt users' data; on the other hand, the cloud service providers may act unfairly, attempting to skin data loss or corruption and requesting that the files are still properly stored in the cloud for status or financial reasons. Thus it makes excessive sense for users to implement an efficient protocol to reach periodical verifications of their outsourced data to confirm that the cloud indeed keeps their data properly. Numerous mechanisms dealing with the integrity of outsourced data without a local copy have been planned under dissimilar system and security models. The greatest significant work among these studies are the ODP (obvious data possession) model and POR (proof of irretrievability) model, which were originally advance for the single-server scenario by Attendees et al. [5] and Jules and Kaliski [2], respectively. Considering that files are usually stripy and redundantly stored across multi-servers or multicourse, [6]–[7] explore integrity verification policies suitable for such multi-servers or multi-clouds setting with different redundancy schemes, set for such multi-servers or multicourse setting with varied redundancy schemes, such as, copying erasure codes, and, more recently, regenerating codes. While cloud computing makes these favorable more appealing than ever, it also brings new and difficult security threats forward users' outsourced data. Since cloud service providers (CSP) are isolated administrative entities, data outsourcing is actually relinquishing user's ultimate deal over the fate of their data. As a result, the In this paper, we focus on the integrity confirmation problem in regenerating-code-based cloud storage, especially with the functional repair strategy [13]. Similar studies have been performed by Chen et al. [8] and Chen and Lee [9] separately and independently. [8] prolonged the single-server CPOR scheme (private version in [14) to there generating-code-scenario; [9] designed and invented a data

integrity protection (DIP) scheme for FMSR [15]-based cloud storage and the scheme is adapted to the thin-cloud setting. However, some of them are designed for private audit, only the data owner is allowed to verify the integrity and repair the faulty servers. Considering the big size of the outsourced data and the user's constrained resource capability, the tasks of testing and reparation in the cloud can be formidable and costly for the users [10]. The overhead of using cloud storage should be decreased as much as possible such that a user does not need to perform too many operations to their out sourced data (in additional to retrieving it) [11]. In particular, users may not want to go through the complexity in verifying and reparation. The auditing schemes in [8] and [9] imply the problem that users need to always stay online, which may prevent 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 public auditing scheme for the regenerating-code based cloud storage, in which the integrity checking and regeneration (of failed data blocks and authenticators) are invented by a third-party examiner and a semi-trusted proxy separately on behalf of the data publisher. Instead of directly adapting the existing public auditing scheme [14] to the multi-server setting, we design a novel authenticator, which is more proper for regenerating codes. Besides, we "encrypt" the coefficients to protect data privacy against the examiner, which is more lightweight than applying the confirmation blind technique in [10] and [11] and data blind method in [12].Several difficulties and threats spontaneously arise in our new system model with a proxy (Section II-C), and security analysis shows that our policy works well with these problems.

Related Work

2.1 Existing System:

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 irretrievability) model, which were originally proposed for the single-server scenario by Ateniese et al [5]. and Juels and Kaliski [2], respectively. Considering that files are usually striped and redundantly stored across 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, erasure codes, and, more recently, regenerating

codes. Chen et al [8]. and Chen and Lee [9] separately and independently extended the 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.

Disadvantages Existing System:

- They 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.
- The auditing schemes in existing imply the problem that users need to always stay online, which may impede its adoption in practice, especially for long-term archival storage.

2.2 Proposed System:

In this paper, we focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy. 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 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 to the multi-server setting, we design a novel authenticator, which is more appropriate for regenerating codes. Besides, we "encrypt" the coefficients to protect data privacy against the auditor, which is more lightweight than applying the proof blind technique and data blind method. We design a novel holomorphic authenticator based on BLS signature, which can be generated by a couple of secret keys and verified publicly.

Advantages of proposed system:

- 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 regenerating code- 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.

- 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.

2.3 System Architecture:

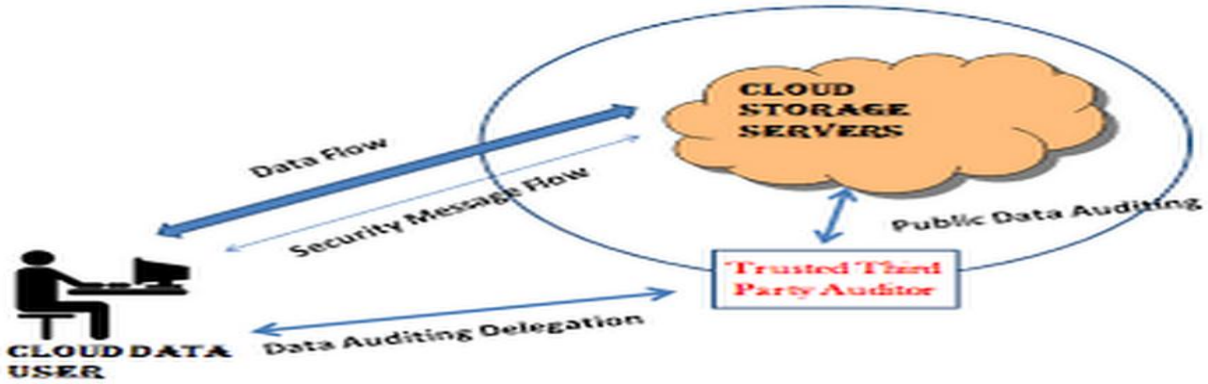

Fig 1: System Architecture.

3. Implementation

3.1 Owner Module:

Who owns large amounts of data files to be stored in the cloud and to save resources 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. Notice that 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.

3.2 Cloud:

Which are managed by the cloud service provider, provide storage service and have significant computational resources.

3.3 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 unbiased for both data owners and cloud servers.

3.4 Proxy:

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 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.

3.5 ALGORITHM:

Authentication, Authorization and Auditing for secure cloud storage is implemented on the basis of the following key points

- Our System Supports an External auditor to audit users outsourced data in the cloud without
- learning knowledge on the data content.
- The TPA supports scalable on request by cloud service provider for efficient public auditing
- in the cloud computing
- Auditing is the processes which is done for the cloud to achieve batch auditing where multiple
- delegated auditing tasks from different users can be performed simultaneously by the TPA
- The auditing is the intelligence based Dynamic data process for the data and information
- security in cloud computing
- data integrity algorithm such as Message Authentication Code (MAC code) by means of Hash
- Based Message Authentication Code (HMAC code) to check the integrity of the data being
- stored in the cloud.
- By means of MAC code, we enhance the data integrity of the cloud data.

Step 1: Start of an Algorithm

Step 2: Key Generation by Advanced Encryption Standard (AES) Algorithm16-bit Hexa Decimal keys are generated

Step 3: Map the Key to the files

Step 4: Divide the files into the blocks

Step 5: Each Encrypted Block is Associated with Key

Step 6: Store the data blocks to the Cloud Storage Server

Step 7: Simultaneously Intelligent system sends a copy of keys to TPA

Step 8: On request of Cloud Service Provider (CSP) the Auditing processes with be done by TPA

Step 9: Validate the data by signatures and data integrity proofs

Step 10: Successful validation, verification will be done for dynamic auditing by TPA

End of Algorithm.

4. Experimental Work

International Journal of Research

Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 12
August 2016

Fig 2: System Home page.

Fig 3: File Data with Encrypted form.

Fig 4: File Data in encrypted format with public Key.

5. Conclusion

In this paper, we propose a public examining scheme for the regenerating-code-in cloud storage system, where the data owners are confidential to delegate third party examiner (TPE) for their data validity checking. To defend the unique information privacy next to the third party examiner (TPE), we randomize the coefficients in the initially rather than applying the shade technique throughout the examining process behavior in mind that the information owner cannot forever stay operative in carry out, in order to keep the storage space existing and verifiable after a malicious corruption, we introduce a partially-trusted proxy into the structure model and provide a confidential for the proxy to handle the reparation of the implicit blocks and authenticators.

6. 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] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] Y. Zhu, H. Hu, G.-J.Ahn, and M. Yu, "Co-operative provable data possession for integrity verification in multi cloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec.2012.

[8] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," inProc.ACM Workshop Cloud Comput. Secur. Workshop, 2010, pp.31–42.

[9] 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.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.