Identification of Corrupted Cloud Storage in Batch Auditing for Multi-Cloud Environments

Sooyeon Shin, Seungyeon Kim, and Taekyoung Kwon(✉)

Graduate School of Information, Yonsei University, Seoul 120-749, Korea
{shinsy80,tribunus000,taekyoung}@yonsei.ac.kr

Abstract. In cloud storage services, users can store their data in remote cloud servers. Due to new and challenging security threats toward outsourced data, remote data integrity checking has become a crucial technology in cloud storage services. Recently, many integrity checking protocols have been proposed. Several protocols support batch auditing, but they do not support efficient identification when batch auditing fails. In this paper, we propose a new identification method for the corrupted cloud in multi-cloud environments without requiring any repeated auditing processes.

Keywords: Cloud computing · Provable data possession · Public auditing · Batch auditing · Identification for the corrupted clouds

1 Introduction

In cloud storage service, users' outsourced data can be lost or corrupted due to outside and inside threats [2,5] but cloud servers might hide data loss incidents by claiming that the data are still correctly in the cloud in order to maintain their reputation. Thus, users need to be able to verify that their outsourced data are correctly stored in the cloud. Recently, many remote integrity checking protocols have been proposed to support public auditability that allows a third party auditor to verify the correctness of outsourced data on demand without retrieving a copy of the whole data [1,3,6–12]. As cloud computing has been widely adopted, a third party auditor may take charge of multiple auditing delegations from different users. To improve auditor efficiency, several protocols support batch auditing, which allows the auditor to simultaneously handle multiple auditing delegations from a large number of different users [3,8–10,12]. For batch auditing, multiple proofs on distinct data of different users are aggregated into a single proof. If a single data block or authenticator has been corrupted or discarded, batch auditing will fail and the benefits of batch auditing could

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1008) supervised by the IITP(Institute for Information & communications Technology Promotion).

I. Khalil et al. (Eds.): ICT-EurAsia 2015 and CONFENIS 2015, LNCS 9357, pp. 221–225, 2015.
DOI: 10.1007/978-3-319-24315-3_22

be canceled out. Wang et al. suggested a divide-and-conquer approach (e.g., binary search) in order to identify the corrupted data [9] and Kai et al. recommended an encoding and decoding approach to reduce communication overhead when batch auditing fails [3]. However, both approaches are inefficient because repeated auditing processes are needed.

To address these problems, in this paper, we propose a new identification method for the corrupted cloud based via two batch auditing schemes [9,10] in multi-users and multi-cloud environments. When the data of users on the single cloud are corrupted, our protocol can identify the corrupted cloud without requiring any repeated auditing processes by utilizing an unique indexing value assigned to each cloud server.

We briefly introduce batch auditing and describe our identification method for the corrupted cloud server in Sects. 2 and 3, respectively. In Sect. 4, we conclude this paper.

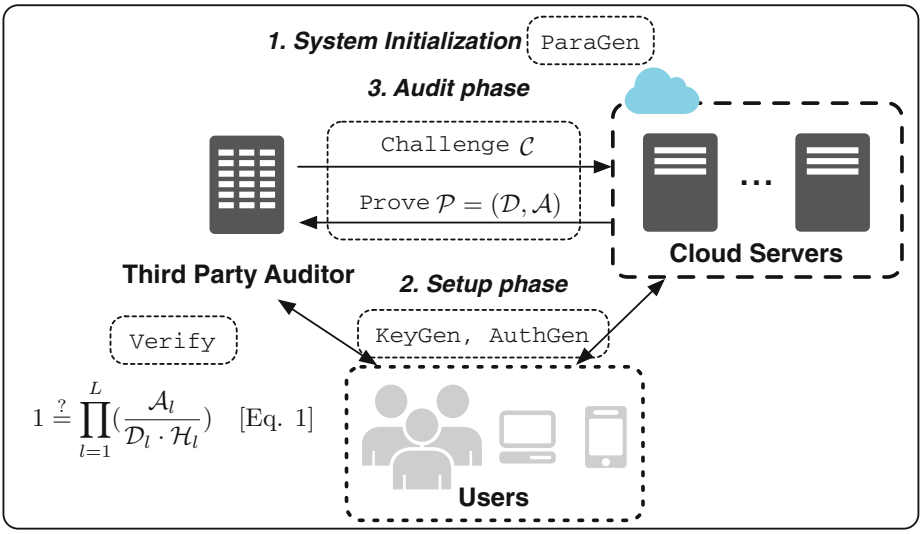

Fig. 1. System model of batch auditing.

2 Batch Auditing

We consider batch auditing for multi-cloud data storage systems involving multi-users and L multi-cloud servers, and the third-party auditor (TPA), as illustrated in Fig. 1. The users store their data in the remote cloud servers and rely on them for data maintenance. Each cloud server $CS_l (1 \leq l \leq L)$ provides powerful storage, computational resources, and the data access to users. The TPA has the expertise and capability to audit data storage on behalf of the user upon request. For batch auditing, we take two batch auditing schemes into consideration: Wang et al's privacy-preserving public auditing scheme (hereafter, W-BA) [9] and

Yang & Jia's efficient and secure dynamic auditing scheme. The original W-BA provides batch auditing for multi-users of single cloud server but it is possible to extend W-BA to support batch auditing for multi-users of multi-cloud servers.

Batch auditing consists of three phases: system initialization, setup, and audit. The system initialization phase involves a `ParaGen` step to generate system parameters and the setup phase involves `KeyGen` and `AuthGen` steps. In `KeyGen` step, a user generates a key pair (public key and secret key) used for computing authenticators of data. In `AuthGen` step, the user computes authenticators used for further auditing then stores them with the data in the remote cloud servers. When multi-users of multi-cloud servers request auditing to the TPA, the TPA performs an audit phase through `Challenge`, `Prove`, and `Verify` steps. In `Challenge`, the TPA randomly selects the number of data blocks for each user and sends a challenge message $\mathcal{C}$ to each cloud server involved in batch auditing. In `Prove`, upon receiving a challenge message, each cloud server sends a response message $\mathcal{P}$ as a proof of possession to the TPA. The proof $\mathcal{P}$ includes the data proof $\mathcal{D}$ and the authenticator proof $\mathcal{A}$. In `Verify`, upon receiving the proofs from the challenged servers, the TPA performs batch verification to check the correctness of all proofs at the same time. The TPA firstly computes the challenged hash $\mathcal{H}_l$ for each cloud server in the challenged servers and simultaneously verifies all proofs via [Eq. 1] in Fig. 1. If the batch verification holds, it means that all challenged servers correctly maintain the data of users.

3 Identification for Single Corrupted Cloud

Batch verification only holds when all of the proofs are valid and fails when there is even one invalid proof of single user in batch auditing. In many situations, a proof collection may contain invalid proofs caused by malicious cloud server or accidental data corruption. If the batch verification equation ([Eq. 1]) fails, it means that a proof of single cloud server is invalid. In this case, the TPA runs `Identify` step to determine which server's proof is invalid.

1. The TPA sequentially allocates an index $I_l (I_l = 1, \cdots, \pi)$ to each CS_l, where π is the number of challenged cloud servers. For example, the second server in the challenged servers has 2 as the index.
2. Let T is the right side of [Eq. 1]. By exponentiation of the corresponding index I_l for each cloud server CS_l, the TPA computes T' as

$$T' = \prod_{l=1}^{L} \left(\frac{\mathcal{A}_l}{\mathcal{H}_l \cdot \mathcal{D}_l}\right)^{I_l} \quad \text{[Eq. 2]}.$$

3. When the data of a single server CS_o are corrupted, T will be Δo, the difference between the original data and the corrupted data . Similarly, T' will be Δo^{I_o}, the I_o-th power of Δo. The TPA can identify the corrupted server by multiplying T by T until the result equals to T' where $T' \stackrel{?}{=} \prod_{i=1}^{I_l} T$. Consequentially, $M - 1$ will be the index for the corrupted server CS_o, where M is the number of multiplications.

There can be several cases of corruption as follows: only data blocks are corrupted or discarded, only authenticators are corrupted or discarded, and both data blocks and authenticators are corrupted or discarded. To hide those corruptions and to deceive the TPA, the malicious cloud server may generate the proof using the user's another valid and uncorrupted pairs (data block and the corresponding authenticator), another user's pairs, random data blocks, or the previous proof. Δo can have several forms according to the above cases. However, it is possible for the proposed method to identify the corrupted server regardless of the forms of Δo.

4 Conclusion and Future Work

In this paper, we proposed a new identification method for the corrupted cloud in multi-users and multi-cloud environments. When the data of users on the single cloud are corrupted and thus batch auditing fails, our method can identify the corrupted cloud without requiring any repeated auditing processes by utilizing an unique index for each cloud server.

As part of future work, we would analyze the performance of the proposed identification method and compare it with the divide-and-conquer approach [9] and sequential re-verification for each server one by one.

References

1. Ateniece, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: CCS 2007, pp. 598–609. ACM press (2007)
2. Claycomb, W.R., Legg, P.A., Gollmann, D.: Guest editorial: emerging trends in research for insider threat detection. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. (JoWUA) **5**(2), 1–6 (2014)
3. Kai, H., Chuanhe, H., Jinhai, W., Hao, Z., Xi, C., Yilong, L., Lianzhen, Z., Bin, W.: An efficient public batch auditing protocol for data security in multi-cloud storage. In: IEEE ChinaGrid Conference, pp. 51–56. IEEE press (2013)
4. Kammüller, F., Probst, C.W.: Invalidating Policies using Structural Information. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. (JoWUA) **5**(2), 59–79 (2014)
5. Lindauer, B., Glasser, J., Rosen, M., Wallnau, K.: Generating Test Data for Insider Threat Detectors. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. (JoWUA) **5**(2), 80–94 (2014)
6. Shacham, H., Waters, B.: Compact proofs of retrievability. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008)
7. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009)
8. Wang, C., Wang, Q., Ren, K., Lou, L.: Privacy-preserving public auditing for data storage security in cloud computing. In: IEEE INFOCOM 2010, pp. 525–533. IEEE Press, New York (2010)

9. Wang, C., Chow, S.S.-M., Wang, Q., Ren, K., Lou, W.: Privacy- preserving public auditing for secure cloud storage. IEEE Trans. Comput. **62**(2), 362–375 (2013)
10. Yang, K., Jia, X.: An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. Parallel Distrib. Syst. **24**(9), 1717–1726 (2013)
11. Zhu, Y., Wang, H., Hu, Z., Ahn, G.-J., Hu, H., Yau, S.S.: Dynamic audit services for integrity verification of outsourced storages in clouds. In: ACM Symposium Applied Computing, pp. 1550–1557. ACM press (2011)
12. Zhu, Y., Hu, H., Ahn, G.-J., Yu, M.: Cooperative provable data possession for integrity verification in multicloud storage. IEEE Trans. Parallel Distrib. Syst. **23**(12), 2231–2244 (2012)