The Journal of Systems and Software 113 (2016) 130–139

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability

Guangyang Yang[a], Jia Yu[a,b,*], Wenting Shen[a], Qianqian Su[a], Zhangjie Fu[b], Rong Hao[a]

[a] College of Information Engineering, Qingdao University, Qingdao 266071, China
[b] School of Computer and Software, Nanjing University of Information Science & Technology, 210044 Nanjing, China

ARTICLE INFO

Article history:
Received 6 July 2015
Revised 13 October 2015
Accepted 23 November 2015
Available online 28 November 2015

Keywords:
Cloud storage
Public auditing
Identity traceability

ABSTRACT

Nowadays, cloud storage service has been widely adopted by diverse organizations, through which users can conveniently share data with others. For security consideration, previous public auditing schemes for shared cloud data concealed the identities of group members. However, the unconstrained identity anonymity will lead to a new problem, that is, a group member can maliciously modify shared data without being identified. Since uncontrolled malicious modifications may wreck the usability of the shared data, the identity traceability should also be retained in data sharing. In this paper, we propose an efficient public auditing solution that can preserve the identity privacy and the identity traceability for group members simultaneously. Specifically, we first design a new framework for data sharing in cloud, and formalize the definition of the public auditing scheme for shared cloud data supporting identity privacy and traceability. And then we construct such a scheme, in which a group manager is introduced to help members generate authenticators to protect the identity privacy and two lists are employed to record the members who perform the latest modification on each block to achieve the identity traceability. Besides, the scheme also achieves data privacy during authenticator generation by utilizing blind signature technique. Based on the proposed scheme, we further design an auditing system for practical scenarios. Finally, we prove the proposed scheme is secure based on several security requirements, and justify its performance by concrete implementations.

1. Introduction

Cloud storage service provides users with enormous storage space to outsource data in an economical and scalable manner (Armbrust et al., 2010; Mell and Grance, 2011). Users can put their data in the cloud to avoid the large expenditure on local hardware/software deployment and data maintenance. While the cloud can be ubiquitously accessed, it is convenient for users to share data with each other through cloud services. In fact, data sharing is a traditional routine that allows a group of users to access the data that belong to this group. It is a common case that employees in the same department of a company store the common-used files or daily reports on a specified server, and they can conveniently access these data as needed. This is a significant desire for users who work together as a group, especially for those in international and collaborative enterprises. To meet this demand in cloud environment, many cloud storage service providers, like Dropbox and iCloud, are presenting cloud data sharing as a primary service.

Since data are stored on remote storage media in the cloud-based storage environment, users are unable to know whether their data are intact as intuitively as on local storage. Although the cloud platform is much more powerful and reliable than individual devices, the cloud might still be compromised because of the hardware failures, the flaws of software and the misbehaviors of system administrator (Donnelly, 2012; Kovacs, 2015; McCarthy, 2012; Miller, 2010; Wikipedia, 2014). Once one of these happens, a tremendous amount of data might get corrupted or lost. If cloud service provider (CSP) does not inform users this incident, users will not detect the abnormality until they access those data. Another concern is that the CSP may intentionally delete rarely accessed data to save storage space (Yang and Jia, 2012). Therefore, the cloud data auditing is proposed to check the integrity of users' data stored in the cloud (Sookhak et al., 2014). In cloud data auditing schemes (Ateniese et al., 2007; Deswarte and Saidane, 2004; Fillo and Baretto, 2006; Juels and Kaliski, 2007), an outsourced file is split into multiple blocks and each block is attached to an authenticator for integrity checking. To verify the cloud data file, a user specifies some random blocks to be checked. The cloud must return an integrity proof of those blocks. During such an auditing procedure, only when correct block-authenticator pairs are presented will users believe their data are properly stored. When

[*] Corresponding author at: College of Information Engineering, Qingdao University, Qingdao 266071, China. Tel.: +86 053285953165.
E-mail address: qduyujia@gmail.com (J. Yu).

http://dx.doi.org/10.1016/j.jss.2015.11.044
0164-1212/© 2015 Elsevier Inc. All rights reserved.

periodically undertakes the auditing procedure, the user will know whether their data become abnormal in time.

The periodic auditing workload, however, is burdensome for individuals who have only limited computing and network resources. Therefore, a third party auditor (TPA) is introduced to perform the auditing tasks for users in later researches (Erway et al., 2009; Guan et al., 2015; Ren et al., 2015; Shacham and Waters, 2008; Sookhak et al., In Press; Wang et al., 2013; Wang et al., 2014; Wang et al., 2013; Wang et al., 2013; Wang, 2013; Wang et al., 2011; Yang and Jia, 2012; Yang et al., 2015; Yu et al., 2015; Yu et al., 2014; Yu et al., In Press; Yu et al., 2015; Yuan and Yu, 2015; Zhang and Blanton, 2013), which is termed as public auditing. In order to check the data integrity, the TPA has to obtain user's public key for verification process. Due to the unique bind between public key and identity information in the public key infrastructure (PKI), the TPA will know who generates the authenticators. This may cause privacy issues in cloud data sharing scenarios, since the authenticators of shared data are generated by different people: the TPA can infer which member in the group is more important based on which member processes more data blocks, and which block is more valuable based on which block is modified more frequently (Wang et al., 2014; Wang et al., 2013). However, although identity privacy is necessary, it may cause another security issue that a member can maliciously modify the shared data without worrying about being found out. In reality, it is not a rare occurrence that an employee intentionally modifies certain critical data for financial interest. That is, a group member might be dishonest as he/she wants to modify data for its own benefits. These modifications may result in disputes among members due to data inconsistency, or even the financial loss of the company if the inconsistency is not properly resolved. This problem gets very tricky in the data sharing scenarios where identity privacy should be achieved, because the absolute identity privacy makes a dishonest member performing malicious modification indistinguishable from other members. As a result, the malicious modifications in such scenarios may become out of control, which will damage the usability of the shared data. Therefore, how to trace the identity of the member who maliciously modifies the shared data is essential in shared cloud data auditing with identity privacy. Yet very few works have considered this important problem before. Group signature seems to be a possible solution to deal with the problem. Unfortunately, group signature involves very complex computation, which is not suitable for constructing efficient auditing schemes for shared data in cloud storage.

In this paper, we propose a novel public auditing scheme for shared data in cloud storage supporting identity privacy and traceability. The contributions of this paper can be summarized as follows:

1 We design a proper framework for data sharing and formalize the definition of the public auditing scheme for shared cloud data which supports public auditability on remote data, and achieves both identity privacy and identity traceability in the sharing group at the same time. We also propose several security requirements that a robust public auditing scheme for shared cloud data should satisfy.
2 We construct a public auditing scheme for shared cloud data, in which the identities of group members are anonymous to the TPA and the group manager can open the identity of a dishonest member when dispute occurs. To protect the identity privacy of group members, a group manager is employed to help the members generate the authenticators of data blocks. At the same time, identity traceability can be achieved through the group manager who records the latest data modification of each data block in a list. Once there is a group member maliciously modifying the shared cloud data, the group manager can find out him/her by looking up this list. Besides, the scheme also achieves data privacy during authenticator generation by utilizing blind signature technique. We also construct an auditing system for practical data sharing applications in cloud environment.
3 We prove the proposed scheme to be secure based on the security requirements and justify its performance by concrete implementations. The detailed performance analysis and experimental results show that the proposed scheme incurs only little overhead to achieve the identity traceability.

Organization. The rest of this paper is organized as follows: Section 2 presents research background and related work; in Section 3, system model, security requirements and preliminaries are presented; our proposed scheme is introduced in Section 4; the security of our scheme is proved in Section 5; Section 6 analyzes the performance of our scheme and presents experimental results; Section 7 concludes this paper.

2. Related work

To preserve the integrity of data on remote storage, researchers have come up with many solutions based on different techniques (Ateniese et al., 2007; Deswarte and Saidane, 2004; Erway et al., 2009; Fillo and Baretto, 2006; Guan et al., 2015; Juels and Kaliski, 2007; Ren et al., 2015; Shacham and Waters, 2008; Sookhak et al., In Press; Wang et al., 2013; Wang et al., 2014; Wang et al., 2013; Wang et al., 2013; Wang, 2013; Wang et al., 2011; Yang and Jia, 2012; Yang et al., 2015; Yu et al., 2015; Yu et al., 2014; Yu et al., In Press; Yu et al., 2015; Yuan and Yu, 2015; Zhang and Blanton, 2013). Deswarte and Saidane (2004) and Fillo and Baretto (2006) employed hash functions to accomplish integrity checking, which has low communication overhead and storage requirement for client. However, like other proposals using traditional cryptographic techniques, the schemes require downloading the whole data for checking the integrity. This is not practical for cloud environment because the large scale storage will consume uncountable communicating and computing resources. In addition, the auditing tasks are cumbersome for the users who have limited computing power and storage capability. Therefore, it is crucial to realize the public auditability that enables users to delegate the periodic data checking workload to a public auditor.

Ateniese et al. (2007) firstly considered the public auditability, and proposed a public auditing model called Provable Data Possession (PDP). Because the time-consuming RSA algorithm is adopted to generate the authenticators of blocks in their scheme, the generation and the verification of data integrity proof are very inefficient. In the same year, Juels and Kaliski (2007) presented the notion of Proof of Retrievability (PoR) that ensures the possession and the retrievability of cloud data by employing spot-checking and error-correcting codes. However, the auditing times are prefixed and the public auditability is not supported in (Juels and Kaliski, 2007). Shacham and Waters (2008) presented a compact version of PoR, which efficiently realizes public auditing based on BLS signature (Boneh et al., 2001). They proved the security of their scheme according to the security model defined in (Juels and Kaliski, 2007). For data sharing consideration, Han et al. (2014) proposed a data sharing scheme utilizing the technique of identity-based proxy re-encryption, in which a proxy server translates encrypted data for receivers without knowing the content of data. In their design, data integrity is checked when users access the data and the original data are required for verification, which makes the data checking inefficient. Besides, sharing data in this way is also inefficient since the data owner has to stay online, and the data sharing procedure is complicated involving the data owner, the receiver and the proxy server. Wang et al. (2013) proposed a more practical way to share data in a group, and achieved user revocation using proxy re-signature technique. However, their scheme can be compromised by collusion attack between the revoked members and the cloud (Yu et al., 2015). Although Yu et al. (2015) proposes a secure user revocation scheme, the identity privacy of group members