



Competent Civic Auditing Mechanism with Preserving Identity Seclusion and Supporting Traceability

¹ Manikanta Reddy.V.K & ² M.Venkatesh Naik

¹PG Student, Dept. of CSE, CHIRANJEEVI REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY,
Affiliated to JNTUA , A.P, India

²Assistant Prof , Dept. of CSE, CHIRANJEEVI REDDY INSTITUTE OF ENGINEERING &
TECHNOLOGY, Affiliated to JNTUA, A.P, India

Abstract—

With cloud computing and storage services, data is not only stored in the cloud, but routinely shared among a large number of users in a group. It remains elusive, however, to design an efficient mechanism to audit the integrity of such shared data, while still preserving identity privacy. In this paper, we propose Knox, a privacy-preserving auditing mechanism for data stored in the cloud and shared among a large number of users in a group. In particular, we utilize group signatures to construct homomorphic authenticators, so that a third party auditor (TPA) is able to verify the integrity of shared data for users without retrieving the entire data. Meanwhile, the identity of the signer on each block in shared data is kept private from the TPA. With Knox, the amount of information used for verification, as well as the time it takes to audit with it, are not affected by the number of users in the group. In addition, Knox exploits homomorphic MACs to reduce the space used to store such verification information. Our experimental results show that Knox is able to efficiently audit the correctness of data, shared among a large number of users.

Keywords- Auditing; Privacy-Preserving; Shared Data; Cloud Computing

I INTRODUCTION

Cloud computing is the newest term for the long-dreamed vision of computing as a utility. The cloud provides convenient, on-demand

network access to a centralized pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead.¹ With its unprecedented advantages, cloud computing enables a fundamental paradigm shift in how we deploy and deliver computing services—that is, it makes possible computing outsourcing such that both individuals and enterprises can avoid committing large capital outlays when purchasing and managing software and hardware, as well as dealing with the operational overhead therein. Although cloud computing benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption.² because cloud service providers (CSPs) are separate administrative entities, moving to the commercial public cloud deprives users of direct control over the systems that manage their data and applications. Even if CSPs' infrastructure and management capabilities are much more powerful and reliable than those of personal computing devices, the cloud platform still faces both internal and external security and privacy threats, including media failures, software bugs, malware, administrator errors and malicious insiders. With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. With Dropbox, for example, data is stored in the cloud (operated by Amazon), and shared among a group of users in a collaborative manner. It is natural for users to wonder whether their data

remain intact over a prolonged period of time: due to hardware failures and human errors in an untrusted cloud environment [2], the integrity of data stored in the cloud can become compromised. To protect the integrity of data in the cloud and to offer “peace of mind” to users, it is best to introduce a third party auditor (TPA) to perform auditing tasks on behalf of users. Such a third party auditor enjoys amply computation/communication resources that users may not possess. Provable data possession (PDP) [3], first proposed by Ateniese et al., allows a verifier to perform public auditing on the integrity of data stored in an untrusted server without retrieving the entire data. Subsequent work focused on how dynamic data and data privacy can be supported during the public auditing process. However, most of previous work only focus on auditing the integrity of personal data. Recently, Wang et al. [6] first design a privacy-preserving public auditing mechanism (named Oruta) for shared data in an untrusted cloud, so that the identity of the signer on each block in shared data is not disclosed to the third party auditor (TPA) during an auditing task. By preserving identity privacy, the TPA cannot figure out which user in the group or which block in shared data is a higher valuable target than others. Unfortunately, Oruta fails to scale well to a large number of users allocation data in a group. In Oruta, information used for verification are computed with ring signatures; as a result, the size of verification information, as well as the time it takes to audit with it, are linearly increasing with the number of users in a group. To make matters worse, when adding new users to a group, all the existing verification information will need to be re-computed if ring signatures are used, introducing a significant computation burden to all users. In addition, the identities of signers are unqualified [8] protected by ring signatures, which avert the group manager to trace the individuality when someone in the group is misbehaved. In this paper, we suggest Knox, a new privacy-preserving mechanism to audit data stored in an

untrusted cloud and shared among a large number of users in a group. In Knox, we take advantage of group signatures to construct homomorphic authenticators, so that the third party auditor is able to confirm the integrity of shared data without retrieving the entire data, but cannot reveal the identities of signers on all blocks in shared data. Meanwhile, the size of verification information, as well as the time it takes to audit with it, is not affected when the number of users sharing the data increases. The original user, who creates and shares the data in the cloud, is able to add new users into a group not including re-computing any verification information. In addition, the original user (acts as the group manager) can trace group signatures on shared data, and reveal the identities of signers when it is necessary. We also utilize homomorphic MACs [1] to effectively reduce the amount of storage space needed to store verification information. As an essential trade-off, we allow the third party auditor to share a secret key pair with users, which we refer to as official auditing. Although we allow an authorized TPA to possess the secret key pair, the TPA cannot compute valid group signatures as group users because this secret key pair is only a part of a group user's private key. To our best knowledge, we present the first mechanism designed with scalability in mind at what time it comes to support auditing data shared among a large number of users in a privacy-preserving fashion.

II RELATED WORK

Ateniese et al. first proposed provable data possession (PDP), which allows a client to verify the integrity of her data stored at an untrusted server without retrieving the entire file. However, this mechanism is only suitable for static data. To improve the efficiency of verification, Ateniese et al. constructed scalable and efficient PDP using symmetric keys. Unfortunately, it cannot support public verifiability, and only offers each user a limited number of verification requests. Juels and Kaliski

defined another similar model called proofs of retrievability (POR), which is also able to check the correctness of data on an untrusted server. The original file is added with a set of randomly-valued check blocks called sentinels. The verifier challenges the untrusted server by specifying the positions of a collection of sentinels, and by asking the untrusted server to return the associated sentinel values. Shacham and Waters designed two improved POR mechanisms, which are built on BLS signatures and pseudo-random functions. Wang et al. leveraged the Merkle Hash Tree to construct a public auditing mechanism with fully dynamic data. Hao et al. also designed a dynamic public auditing mechanism based on RSA. Erway et al. presented a dynamic PDP based on the rank-based authenticated dictionary. Zhu et al. exploited index hash tables to support fully dynamic data. To ensure the correctness of users' data stored on multiple servers, Wang et al. utilized homomorphic tokens and erasure codes in the auditing process. An excellent survey of previous work about data auditing can be found in [1]. Wang et al. considered data privacy with public auditing in the cloud. In their mechanism, the TPA is able to check the integrity of cloud data but cannot obtain any private data. Zhu et al. also designed a mechanism to preserve data privacy from the TPA. Recent work, Oruta, represents the first privacy-preserving public auditing mechanism for shared data in the cloud. In this mechanism, the TPA can verify the integrity of shared data, but is not able to reveal the identity of the signer on each block. Unfortunately, it is not readily scalable to auditing the integrity of data shared among a large number of users in the group. As individuals and enterprises produce more and more data that must be stored and utilized (emails, personal health records, photo albums, tax documents, financial transactions, and so on), they're motivated to outsource their local complex data management systems to the cloud owing to its greater flexibility and cost-efficiency. However, once users no longer

physically possess their data, its confidentiality and integrity can be at risk. For the former concern, data encryption before outsourcing is the simplest way to protect data privacy and combat unsolicited access in the cloud and beyond. But encryption also makes deploying traditional data utilization services —such as plaintext keyword search

over textual data or query over database — a difficult task. The trivial solution of downloading all the data and decrypting it locally is clearly impractical, due to the huge bandwidth cost resulting from cloudscaled systems. Moreover, aside from eliminating local storage management, storing data in the cloud serves no purpose unless people can easily search and utilize that data. This problem on how to search encrypted data has recently gained attention and led to the development of *searchable encryption* techniques. At a high level, a searchable encryption scheme employs a prebuilt encrypted search index that lets users with appropriate tokens securely search over the encrypted data via keywords without first decrypting it.

III PROBLEM STATEMENT

3.1 System Model

In this paper, we consider data storage and sharing services in the cloud with three entities: the cloud, the third party auditor (TPA), and users who participate as a group (as shown in Fig. 1). Users in a group include one original user (OU) and a number of group users (GUs). The original user is the original owner of data, and shares data with other users. Based on access control policies [22], other users in the group are able to access, download and modify shared data. The cloud provides data storage and sharing services for users, and has ample storage space. The third party auditor is able to verify the integrity of shared data based on requests from users, without downloading the entire data. When a user (either the original user or a group user) wishes to check the integrity of

shared data, she first sends an auditing request to the TPA. After receiving the auditing request, the TPA generates an auditing message to the cloud, and retrieves an auditing proof of shared data from the cloud. Then the TPA verifies the correctness of the auditing proof. Finally, the TPA sends an auditing report to the user based on the result of the verification.

3.2 Threat Model

Integrity Threats In general, two kinds of threats related to the integrity of shared data are possible. First, an external adversary may try to pollute shared data in the cloud, and prevent users from using shared data correctly. Second, the cloud service provider may inadvertently corrupt or even remove shared data on the cloud server due to hardware failures and human errors. To make matters worse, in order to avoid jeopardizing its reputation, the cloud service provider may be reluctant to inform users about such corruption of data. **Privacy Threats** During an auditing task, a semi-trusted TPA, who is only responsible for verifying the integrity of shared data, may try to reveal the identity of the signer on each block in the shared data based on verification information (i.e. signatures). The identity of the signer on each block is private and sensitive information, which users in a group do not wish to be revealed to any third party.

3.3 Design Goals

To make it efficient and secure for the TPA to verify shared data with a large number of users in a group, Knox should be designed to achieve the following properties:

- (1) **Correctness:** The TPA is able to correctly detect any corrupted block in shared data.
- (2) **Efficiency:** The TPA is able to verify the integrity of shared data without retrieving the entire data from the cloud server.
- (3) **Identity privacy:** During an auditing task, the TPA cannot distinguish the identity of the signer on each block.

(4) **Support for large groups:** The TPA is able to efficiently audit data that are shared among a large number of users. In particular, the size of verification information, as well as the time it takes to audit with it, are not to be affected by the number of users in the group; the original user can add new users to the group without re-computing existing verification information.

(5) **Traceability:** The original user is able to trace a signature on a block and reveal the identity of the signer.

IV Homomorphic Authenticable Group Signatures

4.1 Overview

As introduced at the beginning of this paper, we expect to utilize group signatures for computing verification information, so that the identity of the signer on each block can be kept private from the TPA. However, traditional group signatures [4, 6, 10] cannot be directly used in Knox, since they are not blockless verifiable. Without blockless verification, a verifier has to download the entire data to check the integrity of shared data, which consumes excessive bandwidth and takes long verification times. Therefore, we first build a homomorphic authenticable group signature (HAGS) scheme in this section. Then we will present the full construction of our privacy-preserving auditing mechanism for shared data among a large number of users based on HAGS in the next. In HAGS, we extend BBS group signatures [6] to achieve blockless verification. Meanwhile, to keep enforceability (nobody outside the group can produce valid signatures) of HAGS, we leverage BLS signatures [9] as a part of our group signatures. BLS signatures, which are based on bilinear maps, are used in previous work [15, 19] to audit data integrity of personal users. In addition, we exploit batch verification methods of group signatures in [12] to improve the efficiency of HAGS for verifying multiple group signatures. Note that if only using BLS signatures among a group of users, which



means that all the users in the group generate signatures on messages only with a common private key, it is also possible to achieve identity privacy on messages. Unfortunately, traceability of the group manager on signatures generated by group members will be immediately

ost. Different from our previous work Oruta, where all the elements in a ring signature are homomorphic, there is only one element (generated by BLS) in a group signature under HAGS is homomorphic (further details are described in following of this section). However, based on our analysis and experiment results, we can still save communication and computation cost for users with this single homomorphic element.

V Privacy-Preserving Auditing for Shared Data
We now present Knox, our privacy-preserving auditing mechanism for shared data among a large number of users. Using HAGS, we can preserve the identity of the signer on each block from the TPA. Meanwhile, the original user, who is the group manager and shares data with other group users, can reveal an identity of a signer. Moreover, the length of each group signature is independent from the number of group users, which is a desirable property for large groups to share their data in the cloud. If users wish to protect the privacy of shared data during an auditing task, users can encrypt data using encryption techniques, such as the combination of symmetric encryption and attribute-based encryption [22], before outsourcing data to the cloud server. The main objective of designing Knox is to provide identity privacy for users. To reduce the storage space of group signatures on shared data, we utilize homomorphic MACs [1] to compress each block into a small value, and then sign this small value instead of signing the entire block. As a necessary trade-off, Knox does not support public auditing, since the TPA in our mechanism needs to share a secret key pair with all group users, referred to as authorized auditing. This secret key pair is used to compute homomorphic

MACs. Although we allow an authorized TPA to possess this secret key pair, the TPA cannot compute valid group signatures as group users because this secret key pair is only a part of a group user's private key. Because the computation of a signature includes an identifier of a block (as we described in HAGS), conventional methods, which only use the index of a block as its identifier, are not suitable for dynamic data. The reason is that when a user modifies shared data by performing an insert or delete operation on a single block, the indices of blocks that after the modified block are all changed, and the change of these indices requires users to re-compute the signatures of these blocks, even though the content of these blocks are not modified. To avoid this type of re-computation and support dynamic data for users, we take advantage of index hash tables [16, 24] as identifiers of blocks. Further explanations about index hash tables can be found in [16, 24]. In addition, we continue to use sampling strategies as previous work [3] to detect any corrupted block in shared data with a high probability, by only choosing a subset of all blocks in each auditing task. For example, if 1% of all the blocks are corrupted, the TPA can detect this misbehavior with probability greater than 99% by choosing only 460 random selected blocks, where the number of selected blocks is independently with the total number of blocks in shared data if the percentage of corrupted blocks is constant [3]. To improve the detection probability, the TPA can increase the number of selected blocks in each auditing task [3, 23]. In some emerging applications cases, the auditor may need to achieve a 100% detection probability if only one corrupted block exists, then all the blocks in shared data should be selected during an auditing task. As a trade-off, the computation and communication cost are significantly increased.



VII CONCLUSION

In this paper, we propose Knox, a privacy-preserving auditing scheme for shared data with large groups in the cloud. We utilize group signatures to compute verification information on shared data, so that the TPA is able to audit the correctness of shared data, but cannot reveal the identity of the signer on each block. With the group manager's private key, the original user can efficiently add new users to the group and disclose the identities of signers on all blocks. The efficiency of Knox is not affected by the number of users in the group.

REFERENCES

- [1]. Agrawal, S., Boneh, D.: Homomorphic MACs: MAC-based Integrity for Network Coding. In: Proc. ACNS. pp. 292–305. Springer-Verlag (2009)
- [2]. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A View of Cloud Computing. Communications of the ACM 53(4), 50–58 (April 2010)
- [3]. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable Data Possession at Untrusted Stores. In: Proc. ACM CCS. pp. 598–610 (2007)
- [4]. Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G.: Scalable and Efficient Prov-able Data Possession. In: Proc. ICST SecureComm. pp. 1–10 (2008)
- [5]. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Proc. CRYPTO. pp. 41–55. Springer-Verlag (2004)
- [6]. Boneh, D., Freeman, D.M.: Homomorphic Signatures for Polynomial Functions. In: Proc. EUROCRYPT. pp. 149–168. Springer-Verlag (2011)
- [7]. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Proc. EUROCRYPT. pp. 416–432. Springer-Verlag (2003)
- [8]. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In: Proc. ASIACRYPT. pp. 514–532. Springer-Verlag (2001)
- [9]. Yang, K., Jia, X.: Data storage auditing service in cloud computing: challenges, methods and opportunities. World Wide Web 15(4), 409–428 (2012)