International Journal on Emerging Technologies (Special Issue on NCRIET-2015) **6**(2): 48-50(2015)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Record Maintenance and Secure Preserving Of Shared Data in Public Auditing

Md. Rafeeq
Asst. Prof Dept. of CSE BKIT, Bhalki, Tamilnadu, INDIA

(Corresponding author: Md. Rafeeq)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. With cloud storage services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. However, public auditing for such shared data, while preserving identity privacy, remains to be an open challenge. In this paper, we propose the first privacy-preserving mechanism that allows public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute the verification information needed to audit the integrity of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from a third party auditor (TPA), who is still able to publicly verify the integrity of shared data without retrieving the entire data.

Key Terms: Secure Preserving, Public Auditing, Shared Data, Cloud Computing, TPA.

I. INTRODUCTION

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures (e.g., RSA) or hash values (e.g., MD5) of the entire data. Certainly, this conventional approach able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt.

The integrity of data in cloud storage, however, is subject to uncertainty. Data stored in the cloud can easily be lost or corrupted due to the unavoidable hardware/ software failures and human errors. To make this matter even bad, cloud service providers may be unwilling to inform users about these data errors in order to maintain the good status of their services and avoid profit loss.

Therefore the integrity of data should be verified before any utilization of data. A public verifier could be a data user (e.g., researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking has been utilized checking service. We believe that sharing data among multiple users is perhaps one of the most engaging features that motivates cloud storage. Therefore it is also necessary to ensure the integrity of shared data in the cloud is correct. secure preserving is compatible with random masking which has been utilized in WWRL and can preserve data privacy from public verifiers. The comparison among different mechanisms as shown in table1.

TABLE 1
Comparison among Different Mechanisms

	PDP [9]	WWRL [5]	Oruta
Public Auditing	√	√	√
Data Privacy	X	√	√
Identity Privacy	X	X	√

II. ARCHITECTURE

A. System Model

In this model we introduce three parties: the cloud server, a group of users and a public verifier. and also it uses two types of users in a group whereas the original user initially creates shared data in the cloud and shares it with group users both original and group users are members of the group.

A public verifier is a third party auditor providing data checking services for publicly verify the integrity of shared data stored in the cloud server TPA wishes to check the integrity of shared data it first sends an auditing challenge i.e. response from pv to cs and after receiving auditing challenge the cloud server will send request to pv to verify and learn that it is the proxy signs the data.

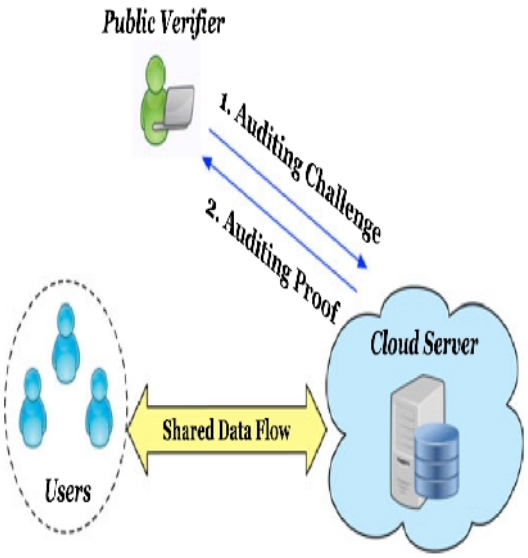

Fig. 1. Our system model includes cloud server, a group of users, and public verifier.

B. Threat model

Integrity threats: In this model first an adversary may try to corrupt the integrity of shared data, second, the cloud service provider may inadvertently corrupt data in its storage due to Hardware failures or human errors.

Privacy threats: The identity of the signer on each block in shared data is private and confidential to the group. During the process of auditing, a public verifier, who is only allowed to verify the correctness of shared data integrity, may try to reveal the identity of the signer on each block in shared data based on verification metadata. Once the public verifier reveals the identity of the signer on each block, it can easily distinguish a high-value target from others.

C. Design Objectives

Our mechanism should be designed to achieve following properties: (1) Public Auditing: A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud. (2) Correctness: A public verifier is able to correctly verify shared data integrity. (3) Unforgeability: Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data. (4) Identity Privacy: A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

D. Construction of Hars

HARS contains three algorithms: Key Gen, Ring Sign and Ring Verify.

Key Gen: Each user in the group generates his/her public key and private key.

Ring Sign: A user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group members public keys. A block identifier is a string that can distinguish the corresponding block from others.

Ring Verify: A verifier is able to check whether a given block is signed by a group member in Ring Verify.

E. Public Auditing Mechanism

Overview: The public verifier can verify the integrity of shared data without retrieving the entire data, meaning that the identity of the signer on each block in shared data is kept private from the public verifier during the auditing.

Reduce signature storage: According to the generation of ring signatures in HARS,a block m is an element of Zp & its ring signature contains d elements of G1 where G1 is a Cyclic group with order p.

Support Dynamic operations: Secure preserving should also support dynamic operations on shared data this operation includes an insert, delete, or update operation on a single block however the computation of ring signature includes an identifier of a block, which use only the index of a block as its identifier.

Security analysis of oruta: It includes correctness, unforgeability; identity privacy data privacy a public verifier is able to correctly audit the integrity of shared data under oruta.

Batch auditing: In this mechanism, this can verify the correctness of multiple auditing tasks & improve the efficiency of public auditing. to allow most of auditing proofs to still pass the verification when there exists only a small number of incorrect auditing proofs, we can utilize binary search during batch auditing.

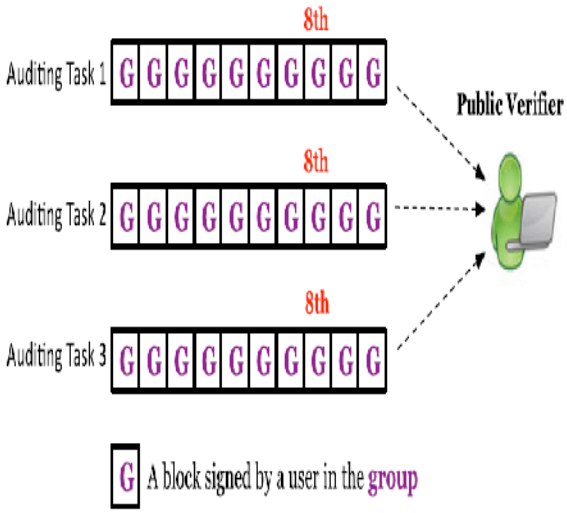

Fig. 2. Alice and Bob share a file in the cloud, and a public verifier audits the integrity of shared data with Oruta.

III. RELATED WORK

Provable data possession (PDP), proposed by Ateniese et al.[1], allows a verifier to check the correctness of a client's data stored at an untrusted server By utilizing RSA-based homomorphic authenticators and sampling strategies, the verifier is able to publicly audit the integrity of data without retrieving the entire data, which is referred to as public auditing. Unfortunately, their mechanism is only suitable for auditing the integrity of personal data. Juels and Kaliski [2] 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 asking the untrusted server to return the associated sentinel values. Shacham and Waters [3] designed two improved schemes. The first scheme is built from BLS signatures [7], and the second one is based on pseudo-random functions.

Wang et al. [4] leveraged homomorphic tokens to ensure the correctness of erasure codes-based data distributed on multiple servers. This mechanism is able not only to support dynamic data, but also to identify misbehaved servers. To minimize communication overhead in the phase of data repair, Chen et al. [5] also introduced a mechanism for auditing the correctness of data under the multi-server scenario, where these data are encoded by network coding instead of using erasure codes. More recently, Cao et al. [6] constructed an LT codes-based secure and reliable cloud storage mechanism. Compare to previous work [4], [5], this mechanism can avoid high decoding computation cost for data users and save computation resource for online data owners during data repair.

IV. CONCLUSION

In this paper, we propose a secure-preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing.

Considering

TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

REFERENCES

[1]. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," *Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07)*, pp. 598-610, 2007.

[2]. A. Juels and B.S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.

[3]. H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90- 107, 2008.

[4]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[5]. B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop Cloud Computing Security Workshop (CCSW'10), pp. 31-42, 2010.

[6]. N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," *Proc. IEEE INFOCOM*, 2012.

[7]. D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing, "Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 514-532, 2001.