ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Shared Data Based Privacy Preserving Authentication in Cloud for Public Auditing

M. Krishna Kumar, S. Britto Raj, K. Ravikumar

Post Graduate Student, Department of CSE, Rrase College of Engineering, Chennai, India

Assistant Professor, Dept of C.S.E Rrase College of Engineering Chennai, India

Professor, Department of CSE, Rrase College of Engineering, Chennai, India

ABSTRACT: With cloud data services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. Unfortunately, the integrity of cloud data is subject to skepticism due to the existence of hardware/software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently audit cloud data integrity without retrieving the entire data from the cloud server. However, public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information—identity privacy—to public verifiers. In this paper, we propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity.

KEYWORDS: Public Auditing; Privacy Preserving; Shared Data; Cloud Computing.

I. INTRODUCTION

Cloud service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approaches. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes a standard feature in most cloud storage offerings, including Drop box, 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. To make this matter even worse, cloud service providers may be reluctant to inform users about these data errors in order to maintain the reputation of their services and avoid losing profits. Therefore, the integrity of cloud data should be verified before any data utilization, such as search or computation over cloud data.

II. RELATED WORK

Provable data possession (PDP), proposed by Ateniese et al., 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 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

DOI: 10.15680/ijircce.2015.0305132

sentinels and asking the untrusted server to return the associated sentinel values. Shacham and Waters designed two improved schemes. The first scheme is built from BLS signatures, and the second one is based on pseudo-random functions.

III. PROPOSED ALGORITHM

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 or hash values of the entire data. Certainly, this conventional approach is able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt. The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste user’s amounts of computation and communication resources, especially when data have been corrupted in the cloud. Besides, many uses of cloud data (e.g., data mining and machine learning) do not necessarily need users to download the entire cloud data to local devices. It is because cloud providers, such as Amazon, can offer users computation services directly on large-scale data that already existed in the cloud.

Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. 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 services. Moving a step forward, Wang et al. designed an advanced auditing mechanism(named as WWRL in this paper), so that during public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers. Unfortunately, current public auditing solutions mentioned above only focus on personal data in the cloud. We believe that sharing data among multiple users is perhaps one of the most engaging features that motivate cloud storage. Therefore, it is also necessary to ensure the integrity of shared data in the cloud is correct. Existing public auditing mechanisms can actually be extended to verify shared data integrity. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers.

For instance, Alice and Bob work together as a group and share a file in the cloud (as presented in Fig. 1). The shared file is divided into a number of small blocks, where each block is independently signed by one of the two users with existing public auditing solutions. Once a block in this shared file is modified by a user, this user needs to sign the new block using his/her private key. Eventually, different blocks are signed by different users due to the modification introduced by these two different users. Then, in order to correctly audit the integrity of the entire data, a public verifier needs to choose the appropriate public key for each block (e.g., a block signed by Alice can only be correctly verified by Alice’s public key). As a result, this public verifier will inevitably learn the identity of the signer on each block due to the unique binding between an identity and a public key via digital certificates under public key infrastructure (PKI). Failing to preserve identity privacy on shared data during public auditing will reveal significant confidential information (e.g., which particular user in the group or special block in shared data is a more valuable target) to public verifiers., after performing several auditing tasks, this public verifier can first learn that Alice may be a more important role in the group because most of the blocks in the shared file are always signed by Alice; on the other hand, this public verifier can also easily deduce that the eighth block may contain data of a higher value (e.g., a final bid in an auction), because this block is frequently modified by the two different users. In order to protect this confidential information, it is essential and critical to preserve identity privacy from public verifier during public auditing.

In this paper, to solve the above privacy issue on shared\ data, we propose, novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data— while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, we further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Meanwhile, it is compatible with random masking, which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a

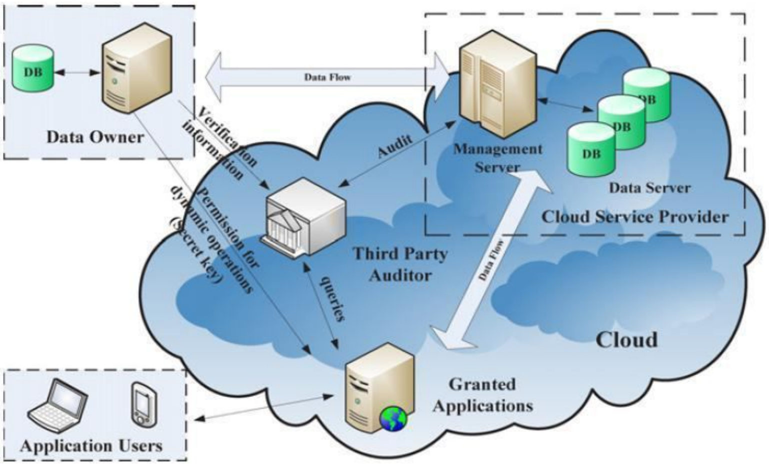
previous public auditing solution to support dynamic data. The proposed system, to solve the above privacy issues on shared data, we propose, a novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures to construct holomorphic authenticators, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, we further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Moreover, we also index hash tables from a previous public auditing solution to support dynamic data. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata are both stored in the cloud server. A public verifier, such as a third party auditor providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. Cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof.

A. **Dynamic User Registration**: In this module, user registers his/her personal details in database. Each user has unique id, username and password and digital signature. If the users enter the wrong password it doesn't allow the user to access the cloud. If the new user enters then sign up with their details, for every unique digital signature is generated. The digital signature is used for encrypt and decrypt the data.

The user registration has three types of user are registered. They are Owner, user and public verifier. The mode of operation of the owner is uploads the data and provides access permission to all other users. The other user is entered with their corresponding username and password. After using these details he can request file from server. The data has been converted to the encoded format and it will store back to the database. The use of public verifier is to audit the shared data. The public auditor doesn't know the details about the data. The auditor audits the data once in a day. Here the log file is generated for the future references. User will download data later without any loss in data.

B. **Data Center**: Data Center is the location where the cloud activities are been handled. The request from the user is been retrieved at the data center and the process is been completely done at this area. The request is been individually identified by the request id, file name and the content for each respective request id. The data center is the central processing area where the file uploaded by the user is been converted to the zip format and is been stored back to the database. For security propose we are using four data centers. If the owner shared the file in the cloud then immediately the file is splitted into four parts. Three parts of data are store in each data center. There are two advantages. If the hacker hack the single data center then he/she doesn't have the full details. And also if any server failure occurs then we recover the data from the other server. Data centers are fully secured and the recovery of data is also done in a good manner.

C. **Public Verifier Audit Services**: A public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. A public Verifier could be a data user who would like to utilize the owner's data via the cloud that can provide expert integrity checking services. Public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers. Log file are maintained by the public verifier for the future reference. The log file provide the details about the which name of user, time and the operation that are done by the users. The public verifier provides the details to the user for the purpose of who is access the data that is shared by the owner. The public verifier receives the all login details from the cloud server and prepares the log file. The log file contains the actions done by the user, but it doesn't enclose the details of the user. The public verifier only able to perform auditing on the process.



Audit system architecture for cloud computing.

Fig 2. Audit System

IV. PSEUDO CODE

Homomorphic authenticators (also called homomorphic verifiable tags) are basic tools to construct public auditing mechanisms. Besides unforgeability (i.e., only a user with a private key can generate valid signatures), a homomorphic authenticable signature scheme, which denotes a homomorphic authenticator based on signatures, should also satisfy the following

properties:

Let (pk; sk) denote the signer's public/private key pair,
s1 denote a signature on block m1 ∑ Zp, s2 denote a
signature on block m2 ∑ Zp.

Blockless verifiability: Given s1 and s2, two random
values a1, a2   Zp and a block m'= a1m1+ a2m2 ∑ Zp, a verifier is able to check the
correctness of block m' without knowing block m1 and m2.

Non-malleability: Given s1 and s2, two random
values a1, a2 ∑ Zp and a block m'= a1m1 + a2m2
∑ Zp, a

A user, who does not have private key sk, is not able to generate a valid signature s1 on block m1 by linearly combining signature s1 and s2.

V. SIMULATION RESULTS

The stimulation result involves the user upload files to the cloud server. The user can upload new file, or edit the existing file and delete the file. The entire user (Users, Third Party Auditor and Admin) can use the same server to perform the task. The auditors should be login as third party auditor role. Every artifact in the system is

DOI: 10.15680/ijircce.2015.0305132

encrypted and spited into four and stored in different server so the auditor can easily audit the system without revealing any information. The auditor can able to view the process but unable to view the user information in the file. Since the Auditor and Users uses the same server the performance will be high and more cost efficient. The auditor can audit the system anytime and also in parallel. Multiple auditors can login simultaneously and audit the system. The reporting is also more efficient in the system

IV. CONCLUSION AND FUTURE WORK

In this paper, we propose a privacy-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. There are two interesting problems we will continue to study for our future work. One of them is traceability, which means the ability for the group manager (i.e., the original user) to reveal the identity of the signer based on verification metadata in some special situations. Since it is based on ring signatures, where the identity of the signer is unconditionally protected, the current design of ours does not support traceability. To the best of our knowledge, designing an efficient public auditing mechanism with the capabilities of preserving identity privacy and supporting traceability is still open. Another problem for our future work is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy.

REFERENCES

1. B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp. 295-302, 2012.
2. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.
3. K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
4. D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.
5. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.
6. B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. And Network Security (CNS '13), pp. 90-99, 2013.
7. The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.
8. G. 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.
9. 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.

BIOGRAPHY

Krishna Kumar M is a post graduate student in the department of computer science engineering, rrase college of engineering, Anna University. He received Bachelor of Engineering degree (B.E) in 2011 from Anna University, Chennai, India. His research interests are Data Mining, Security, Cloud computing etc.

DOI: 10.15680/ijircce.2015.0305132