Public Auditing for Shared Data in the Cloud by Preserving Privacy

Ihtesham Azhar
M. Tech, CSE student
T. John Institute of Technology
Bangalore, India

Narasimhayya B E
Associate Professor, Dept. of CSE
T. John Institute of Technology
Bangalore, India

***Abstract*— The cloud storage service is a common place to store the data as well as share the data across multiple users. Whenever the concept of sharing of data among multiple users come, public auditing of such a shared date by preserving the identity privacy remains to be an open challenge . We propose the first privacy-preserving mechanism that allows public auditing on shared data stored in the cloud by introducing a TPA (third party auditor). The shared file is partitioned into several smaller blocks and the identity of the signer of each block is kept private from the third party auditor(TPA), with this mechanism the TPA is able to verify the integrity of the shared data without retrieving the entire file ,this is due to exploiting signatures to compute the verification information needed to audit the integrity of shared data.**

Keywords*— *Public auditing , cloud computing , shared data, privacy-preserving.

I. INTRODUCTION

Cloud service providers manage an infrastructure that offers a highly reliable, secure, and scalable and environment for users, at a much lower cost this is due to the sharing of resources.

The integrity of data in stored cloud, is subject to distrust, as data stored in an cloud can easily be lost or corrupted, either due to hardware failures or due to human errors [1] or due to both. To protect the integrity of cloud data, the best consideration for public auditing is to introduce a third party auditor (TPA).

To verify the correctness of data stored in an server ,the provable data possession (PDP) mechanism [2] for public auditing is designed, which checks the correctness of the data stored on the untrusted cloud without even retrieving the entire data. Which in this paper is referred to as WWRL, is designed to construct a mechanism for public auditing in cloud data, so that during performing public auditing, the data belonging to a personal user is not disclosed to the third party auditor.

A unique problem is introduced that is how the identity privacy is preserved from the third party auditor during performing the public auditing process for shared data in the cloud, because the identity of the signer on the shared data might indicate that a particular user in a group or a particular block in a shared data is higher priority target than the others. Consider an example, Alice and Bob are two people who work together as a group and they both share a common file in the cloud. Where the file shared between them is divided or we can say partitioned into a number of small blocks, which are signed independently by them. If a particular block in this shared file is modified by any group user either alice or bob, then this user using the private key needs to sign the newly modified block. To audit the integrity of the whole file based on the users requests from any of the group users. The signer's identity on each block in this shared file is needed to be known to the third party auditor.

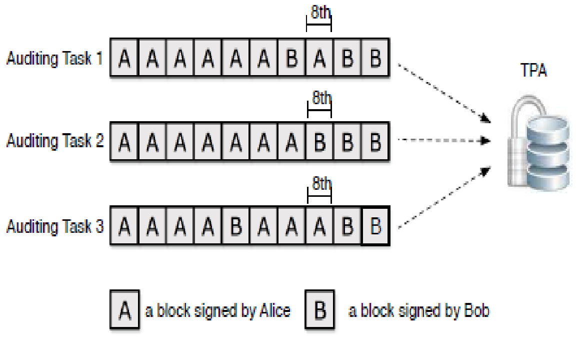
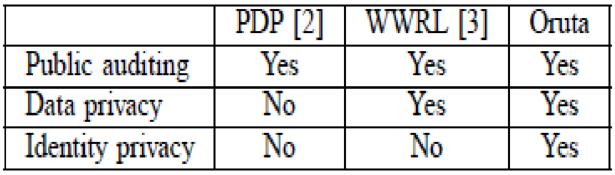


Fig. 1. Alice and Bob share a file in the cloud, and the TPA verifies the integrity of the shared data with existing mechanisms.

As shown in the above Fig. 1, after doing several auditing tasks, some of the private and sensitive information may be revealed to the TPA. Its also seen that, most of the blocks in shared file are signed by the user Alice, which may indicate that the user Alice is playing an important role to this group, which would be a group leader. And also, the 8-th block of the shared file is frequently modified by different users. Which means this block may contain a very highly valuable data, that the users of the group, Alice and Bob need to discuss over it and make change to this block many a times. As said in the above example, the identity of signer over each block of the shared data might indicate that which of the users in the group or which of the block in shared data is a higher value target than others, and such kind of confidential information belonging to the group should not be disclosed to the third party auditor. While still preserving identity privacy and to perform public auditing on shared data in the cloud we need a mechanism. However in the literature there exists no such mechanism. Here, we propose a very first privacy-preserving mechanism for public auditing of shared data in a cloud (Oruta).

In Oruta, we make use of ring signatures [4], [5] so as to construct homomorphic authenticators [2], [6], due to which the TPA, without retrieving the entire data is still able to verify the integrity of shared data belonging to a group of users on the untrusted cloud. And also the identity of the signer over each block of shared data is kept private which means it is not revealed to the TPA, and also Oruta can continue to support dynamic operations on data during public auditing and data privacy.



	PDP [2]	WWRL [3]	Oruta
Public auditing	Yes	Yes	Yes
Data privacy	No	Yes	Yes
Identity privacy	No	No	Yes

Table 1.Comparison with existing mechanisms

A comparison between newly proposed Oruta and already existing mechanisms is shown in Table I. So far Oruta is the first attempt to design an effective privacy-preserving mechanism for public auditing of shared data in the cloud.

II. RELATED WORK

Ateniese et al. [2] first proposed provable data possession (PDP), which allows the verification of the integrity data without retrieving the entire data stored in an untrusted server. PDP is the first mechanism that provides public auditing .However, its limited only to supports static data.

To improve the efficiency of verification. Ateniese et al. [9] Using symmetric keys constructed a scalable and efficient PDP. This mechanism is able to partially support dynamic data operations. But it does not support public verifiability,and offers each user only a limited number of verification requests.

Juels and Kaliski [10] is another much similar model which is called as proof of retrievability (POR), which is also able to check the correctness of data which is stored in an untrusted server. Where the original file is added to sentinels which is a set of randomly-valued blocks.The user asks the server to return specific sentinel values and based upon these sentinal values The user verifies the integrity of data.

Shacham and Waters [6] designed improved POR, which are built on BLS signatures [11] and pseudorandom functions.

Wang et al. [7] so as to support fully dynamic data,which makes use of the Merkle Hash Tree so as to construct a public auditing mechanism.

Erway et al. [12] based on the rank-based authenticated dictionary presented a fully dynamic PDP.

Zhu et al. [8] in order to support fully dynamic data during the process of public auditing process, have exploited index hash.

More recently, Wang et al. [3] ,where the TPA is able to verify the integrity of cloud data without obtaining the private data and hence is considered as public auditing for data stored in cloud along with preserving the data privacy. In this mechanism. Additionally, they have also extended their mechanism to enable batch auditing by making use of the strength of aggregate signatures [5] ,in order to operate multiple users' auditing tasks simultaneously .

Our recent work [13], it is able to audit the integrity of shared data for large groups in the cloud, but Unfortunately it cannot support public auditing,.

III. ARCHITECTURE AND DESIGN

As illustrated in following Fig. 2, our work here involves three parties: users, the cloud server and the third party auditor (TPA). There are two kind of users in a group: A number of group users and the original user. Both the The original user and group users are members of the group. The original user creates the shared data and based upon the access control polices Group members are allowed to access and modify shared data which is created by the original user. The cloud server stores both the Shared data as well as its verification information (i.e. signatures),on behalf of group members, the third party auditor is able to verify the integrity of shared data of the group stored in the cloud server.

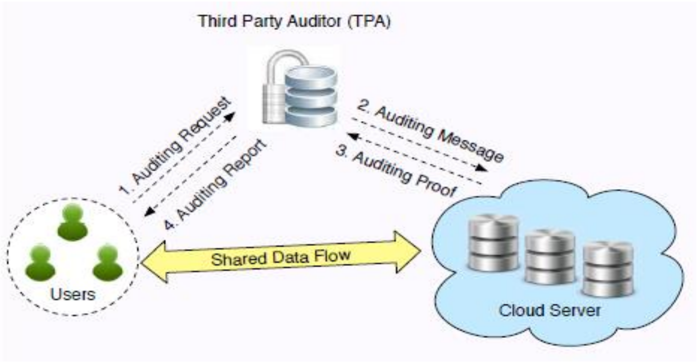


Figure. 2. System model has the cloud server, the third party auditor and users.

Here in our paper, we only consider the static groups in order to audit the integrity of shared data stored in the cloud. Which means that before the users shared data is created in the cloud, the group is pre-defined and during data sharing the membership of users belonging to the group is not changed. Its the responsibility of the original user to decide that who is able to share the data before the data is outsourced to the cloud.

In dynamic groups new users can be added into the group as well as an existing group member can be revoked from the group during sharing of data. And interesting problem is how to preserve identity privacy during auditing the integrity of shared data which is stored in the cloud with such dynamic groups, We leave this problem as our future work .

Whenever a user wants to check the integrity of shared data, where the user can be either the original user or a group user. First of all the user sends an auditing request to the TPA .And after receiving the auditing request, an auditing message is generated by the TPA which is sent to the to the cloud server, after which auditing proof of shared data is retrieved from the cloud server by the TPA. Then the correctness of the auditing proof is verified by the TPA . Finally, an auditing report is sent to the user according to the result obtained by the verification process, and this auditing report is sent to the user by the TPA.

In order to hide the signer's identity on each block we utilize ring signatures, so that the sensitive and private information belonging to the group is not revealed to the third party auditor. However in public auditing mechanisms the traditional ring signatures [4],[5] cannot be directly used , that is because blockless verification is not supported by these ring signature schemes.

The TPA should download entire data file so as to verify the correctness of shared data , if the blockless verification is not used, which would take long verification times and would consume excessive bandwidth. Hence we construct a new scheme called homomorphic authenticable ring signature (HARS), which is the extension of classic ring signature scheme[5],which is also denoted as BGLS.

The ring signatures which are generated by HARS is able to support both blockless verification as well as its able preserve identity privacy.

The size of storage for ring signatures is the Another important issue which should be considered during the construction of Oruta. As per the generation of ring signatures in HARS, a block m is an element of Zp. And the ring signature of Zp contains d elements of G1, , where G1 is a cyclic group with order p. It means d × |p|-bit ring signature is required by a |p|-bit block, which leads to the requirement of huge amount of space on storing ring signatures, which is very frustrating to the users, because the cloud service providers such as Amazon, charges its users based upon the storage space they use.

In order to reduce the storage size for the ring signatures and also allow the TPA to still audit the shared data efficiently, we use an aggregated approach from [6].

A. Threat Model

1) Integrity Threats : As per the integrity of shared data, Two kinds of threats are possible. First, the integrity of the shared data might be corrupted by the adversary which prevents the users from using data correctly. Second, the data in the storage may be corrupted or may even removed from the cloud storage by the cloud service providers which would be due to human errors or hardware failures.

2) Privacy Threats : Based on verification information a semi-trusted TPA, who is responsible for the verification of the integrity of shared data, may try revealing the identity of the signer on each and every block of the shared data, where the signer 's identity over each block in shared data is very much confidential and is private to the group. The TPA can easily distinguish a high-value target once he reveals signer's identity over each block of the data shared on the cloud.

B. Design Objectives

Oruta must be designed in order to achieve following properties, To enable the TPA for efficient and secure verification of shared data for a group of users :

1) Public Auditing : Here the third party auditor is able to verify the integrity of the data shared among a group of users which is stored in the cloud, without even retrieving the complete data from the cloud storage.

2) Correctness: If the shared data contains any corrupted blocks, The third party auditor is able to detect such kind of corrupted blocks correctly.

3) Unforgeability : The valid verification information on shared data can only be generated by a user in the group.

4) Identity Privacy: During auditing, the signer's identity over each block of shared data cannot be distinguished by the TPA.

IV. CONCLUSIONS AND FUTURE WORK

Here we propose the first privacy-preserving mechanism for public auditing of shared data in the cloud. With this, the TPA is able to perform auditing for verifying the integrity of shared data in a efficient way, and also the signer's identity over each block in shared data cannot be distinguished by the TPA, which preserves the identity privacy for users.

In the dynamic groups new users can be added as well as existing group member can be revoked from the group during sharing of data. And interesting problem is how to preserve identity privacy during performing the auditing task so as to integrity of shared data in the cloud with such dynamic groups, We leave this problem to our future work .

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith “A View of Cloud Computing,” Communications of the ACM, vol. 53, no. 4, pp. 50–58, April 2010.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” in Proc. ACM CCS, 2007, pp. 598–610.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” in Proc. IEEE INFOCOM, 2010, pp. 525–533.

[4] R. L. Rivest, A. Shamir, and Y. Tauman, “How to Leak a Secret,” in Proc. ASIACRYPT. Springer-Verlag, 2001, pp. 552–565. [5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in Proc. EUROCRYPT. Springer-Verlag, 2003, pp. 416–432.

[6] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” in Proc. ASIACRYPT. Springer-Verlag, 2008, pp. 90–107.

[7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing,” in Proc. European Symposium on Research in Computer Security. Springer-Verlag, 2009, pp. 355–370.

[8] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds,” in Proc. ACM Symposium On Applied Computing, 2011, pp. 1550–1557.

[9] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” in Proc. ICST SecureComm, 2008.

[10] A. Juels and B. S. Kaliski, “PORs: Proofs pf Retrievability for Large Files,” in Proc. ACM CCS, 2007, pp. 584–597.

[11] D. Boneh, B. Lynn, and H. Shacham, “Short Signature from the Weil Pairing,” in Proc. ASIACRYPT. Springer-Verlag, 2001, pp. 514–532.

[12] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” in Proc. ACM CCS, 2009, pp. 213–222.

[13] B. Wang, B. Li, and H. Li, “Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud,” in Proc. ACNS. SpringVerlag, 2012.