ISSN:0975-9646

Ashwini H et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 2566-2569

# Private Preserving Publically Checking for Shared Data in the Cloud Using Oruta

Ashwini H[1], Mrutyunjay B[2], Sangamesh P[3], Smitha D[4], Veeresh Biradar[5]

[1,2,3,4]*Student*
[5]*Asst. professor*
*Department Of Computer Science and Engineering,*
*Lingaraj Appa Engineering College, Bidar, Karnataka, India*

**Abstract— As per the 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 doubt about the truth of something due to the presence of hardware/software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently check the cloud data integrity without retrieving the entire data from the cloud server. In this paper, we propose a novel private-preserving mechanism that supports publically checking on shared data stored in the cloud. 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 check the shared data integrity without retrieving the entire file. Our experimental results show the effectiveness and efficiency of our mechanism when checking shared data integrity.**

**Index Terms—Public checking, private-preserving, shared data, cloud computing, Ring signature.**

## 1 INTRODUCTION

Cloud service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approach. As data sharing becomes a standard feature in most cloud the 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. 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.

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 of the entire data. Certainly, this conventional approach is able to successfully check the correctness of cloud data.

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 users amounts of computation and communication resources, especially when data have been corrupted in the cloud. Besides, many uses of cloud data 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 checking. 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 or a third-party auditor (TPA) who can provide expert integrity checking services. We believe that sharing data among multiple users is perhaps one of the most engaging features that motivate cloud storage. In this paper, to solve the privacy issue on shared data, we propose Oruta a novel private-preserving public checking mechanism.

More specifically, we utilize ring signatures to construct homomorphic authenticators in Oruta, 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 checking tasks simultaneously and improve the efficiency of verification for multiple checking tasks. Meanwhile, Oruta 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 checking solution to support dynamic data. A high-level comparison among Oruta and existing mechanisms is presented.

## 2. MODELS

### *2.1 System Model*

The system model in the below figure (figure 1) involves three parties: the cloud server, a group of users and a public verifier.

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 (i.e., signatures) 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.

After receiving the auditing challenge, the 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.

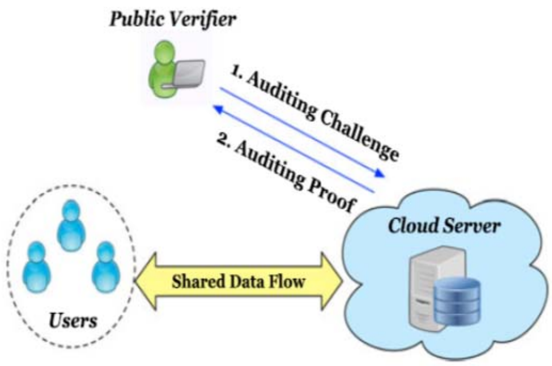


**Figure 1: system model**

Essentially, the process of public auditing is a challenge and response protocol between a public verifier and the cloud server.

*2.2 Threat Model*

*Integrity Threats:* Two kinds of threats related to the integrity of shared data are possible. First, an adversary may try to corrupt the integrity of shared data. Second, the cloud service provider may inadvertently corrupt (or even remove) data in its storage due to hardware failures and human errors. Making matters worse, the cloud service provider is economically motivated, which means it may be reluctant to inform users about such corruption of data in order to save its reputation and avoid losing profits of its services.

*Privacy Threats:* The identity of the signer on each block in shared data is private and confidential to the group. During the process of checking, 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 (a particular user in the group or a special block in shared data) from others.

## 3 PRELIMINARIES

*3.1Ring Signatures*

The concept of ring signatures was first proposed by Rivest et al. in 2001. With ring signatures, a verifier is convinced that a signature is computed using one of group members' private keys, but the verifier is not able to determine which one. More concretely, given a ring signature and a group of d users, a verifier cannot distinguish the signer's identity with a probability more than 1=d. This property can be used to preserve the identity of the signer from a verifier. The ring signature scheme introduced by

Boneh et al (referred to as BGLS in this paper) is constructed on bilinear maps.

We will extend this ring signature scheme to construct our public auditing mechanism.

## 4 NEW RING SIGNATURE SCHEME

*4.1 Overview*

As we introduced in previous sections, we intend to utilize ring signatures to hide the identity of the signer on each block, so that private and sensitive information of the group is not disclosed to public verifiers. However, traditional ring signatures cannot be directly used into public auditing mechanisms, because these ring signature schemes do not support block less verifiability.

Without block less verifiability, a public verifier has to download the whole data file to verify the correctness of shared data, which consumes excessive bandwidth and takes very long verification times. Therefore, we design a new homomorphic authenticable ring signature (HARS) scheme, which is extended from a classic ring signature scheme. The ring signatures generated by HARS are not only able to preserve identity privacy but also able to support block less verifiability.

We will show how to build the privacy-preserving public auditing mechanism for shared data in the cloud based on this new ring signature scheme in the next section.

*4.2 Construction of HARS*

HARS contains three algorithms: **KeyGen**, **RingSign** and **RingVerify**. In **KeyGen**, each user in the group generates his/her public key and private key. In **RingSign**, 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. A verifier is able to check whether a given block is signed by a group member in **RingVerify**.

## 5. BATCH AUDITING

Sometimes, a public verifier may need to verify the correctness of multiple auditing tasks in a very short time. Directly verifying these multiple auditing tasks separately would be inefficient.

By leveraging the properties of bilinear maps, we can further extend Oruta to support batch auditing, which can verify the correctness of multiple auditing tasks simultaneously and improve the efficiency of public auditing.

In simple words the explanation of batch auditing is just a batch checking.

Multiple batch checking is done without consuming the much time and by this technique we can hike the level of the correctness of the data present. The batch auditing or checking is more efficient, dependable and less time consuming then the auditing or checking one by one.

Following figures or snapshots shows the batch auditing mechanism process.


**Figure: Owner Login Form**

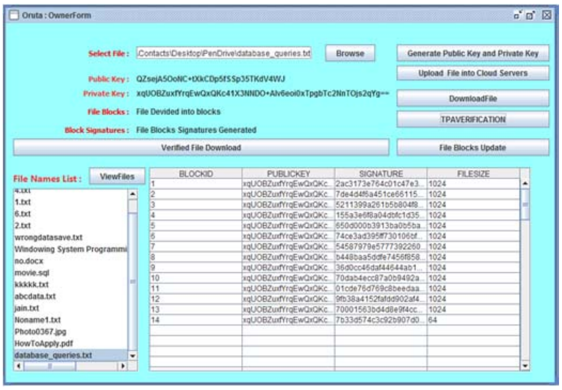


**Figure: Owner Home Page**


**Figure: Auditing graph**

## CONCLUSION AND FUTURE WORK

In this paper, we propose Oruta, 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 Oruta is based on ring signatures, where the identity of the signer is unconditionally protected, the current design of ours does not support traceability.

## REFERENCES

[1] 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.

[2] B. Wang, B. Li, and H. Li, “Certificateless Public Auditing for Data Integrity in the Cloud,” Proc. IEEE Conf. Comm. and Network Security (CNS’13), pp. 276-284, 2013.

[3] C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[4] B. Wang, B. Li, and H. Li, “Public Auditing for Shared Data with Efficient User Revocation in the Cloud,” Proc. IEEE INFOCOM, pp. 2904-2912, 2013.

[5] B. Wang, B. Li, and H. Li, “Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud,” IEEE Trans. Services Computing, 20 Dec. 2013, DOI: 10.1109/TSC.2013.2295611.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” Proc. 22nd Int’l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT’03), pp. 416-432, 2003.

[7] B. Wang, H. Li, and M. Li, “Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics,” Proc. IEEE Int’l Conf. Comm. (ICC’13), pp. 539-543, 2013.

[8] B. Wang, S.S. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” Proc. IEEE 33rd Int’l Conf. Distributed Computing Systems (ICDCS’13), pp. 124-133, 2013.

[9] D. Boneh, X. Boyen, and H. Shacham, “Short Group Signatures,” Proc. 24th Ann. Int’l Cryptology Conf. (CRYPTO’04), pp. 41-55, 2004.

[10] B. Wang, B. Li, and H. Li, “Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud,” Proc. 10th Int’l Conf. Applied Cryptography and Network Security (ACNS’12), pp. 507-525, June 2012.

[11] E. Brickell, J. Camenisch, and L. Chen, “Direct Anonymous Attestation,” Proc. 11th ACM Conf. Computer and Comm. Security (CCS’04), pp. 132-145, 2004.

[12] 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.

[13] R.L. Rivest, A. Shamir, and Y. Tauman, “How to Leak a Secret,” Proc. Seventh Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT’01), pp.552-565,2001

[14] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Remote Data Checking for Network Coding-Based Distributed Storage Systems,” Proc. ACM Workshop Cloud ComputingSecurity Workshop (CCSW’10), pp. 31-42, 2010.

[15] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, “Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds,” Proc. ACM Symp. Applied Computing (SAC’11), pp. 1550-1557, 2011.

[16] 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.
[17] B. Wang, B. Li, and H. Li, "Certificateless Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.
[18] C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[19] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013.
[20] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," IEEE Trans. Services Computing, 20 Dec. 2013, DOI: 10.1109/TSC.2013.
[21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT'03), pp. 416-432, 2003.
[22] B. Wang, H. Li, and M. Li, "Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics," Proc. IEEE Int'l Conf. Comm. (ICC'13), pp. 539-543, 2013.
[23] B. Wang, S.S. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13), pp. 124-133, 2013.
[24] [24] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," Proc. 24th Ann. Int'l Cryptology Conf. (CRYPTO'04), pp. 41-55, 2004.
[25] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS'12), pp. 507-525, June 2012.
[26] E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation," Proc. 11th ACM Conf. Computer and Comm. Security (CCS'04), pp. 132-145, 2004.
[27] 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.
[28] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 552-565, 2001.
[29] D. Cash, A. Kupcu, and D. Wichs, "Dynamic Proofs of Retrievability via Oblivious RAM," Proc. 32nd Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT), pp. 279-295, 2013.
[30] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, pp. 534-542, 2010.