PRIVACY-PRESERVING PUBLIC AUDITING FOR SHARED DATA WITH EFFICIENT USER REVOCATION

Alabazar Ramesh[1], V.Sujatha[2]
[1]M.Tech student, CSE, Kottam college of Engineering, Chinnatekuru(V),Kurnool,Andhra Pradesh, India
[2]Assistant Professor, CSE , Kottam college of Engineering, Chinnatekuru(V),Kurnool,Andhra Pradesh, India

Abstract- **Users in a particular group need to compute signatures on the blocks in shared data,so that the shared data integrity can be confirmed publicly,.Various blocks in shared data are usually signed by various vast number of users due to data alterations performed by different users. Once a user is revoked from the group, an existing user must resign the data blocks of the revoked user in order to ensure the security of data. Due to the massive size of shared data in the cloud, the usual process, which permits an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient. 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.**

Index Terms- **Public auditing, privacy-preserving, shared data, cloud computing**

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 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. 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 owners data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. Existing public auditing mechanisms can actually be extended to verify shared data integrity and data freshness. 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 .To protect the confidential information, it is essential and critical top reserve identity privacy from public verifiers during public auditing. In our model, privacy is accomplished by allowing the parties to upload their data in multi clouds and data is split into multiple parts so it gives more protection.The critical reasons due to which our above system is beneficial as:

1. Current working scenario involves paper based work for Data analysis and verification.

2. Data Storage is one way to mitigate the privacy concern.

3. Unauthorized users can leak or misuse the data, this problem still remains due to the paper based work.

These are the above reasons which compel us to propose Oruta, a novel privacy preserving public auditing 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.

For the first time data inserting the Encryption service generate encryption key and this key is stored separately on Key Storage area, and encrypted data is stored on the cloud storage area.In decryption process when the user request for the data, then key and data are collected at the Decryption service but the service will not immediately decrypt the data, until and unless user insert the OTP sent on his mail. When user will enter this OTP correctly then the data is decrypted by Decryption service and data is provided to the user. Some researchers have suggested that user data stored on a Service- provider’s equipment must be encrypted.

Encrypting data prior to storage is a common method of data protection, and service providers may be able to build firewalls to ensure that the decryption keys associated with encrypted user data are not disclosed to outsiders. However, if the decryption key and the encrypted data are held by the same service provider, it raises the possibility that high-level administrators within the service provider would have access to both the decryption key and the encrypted data, thus presenting a risk for the unauthorized disclosure of the user data.

Existing methods for protecting data stored in cloud environment are user authentication, building secure channel for transmission of data. For this procedure they use various Cryptographic as well as Security based algorithm such as AES (Advance Encryption Standard), DES (Data Encryption Algorithm), Triple DES, RSA algorithm with digital signature.

II. RELATED WORKS

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 [3]. Moving a step forward, Wang et al. designed an advanced auditing mechanism [2] (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 [1]. 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. Existing public auditing mechanisms can actually be extended to verify shared data integrity [1], [2], [4], [5]. 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 [1].

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. Specifically, as shown in Fig. 1, 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 these confidential information, it is essential and critical to preserve identity privacy from public verifiers during public auditing.

In this paper, to solve the above privacy issue on shared data, we propose Oruta,1 a novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures [6] to construct homomorphic authenticators [10] 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.

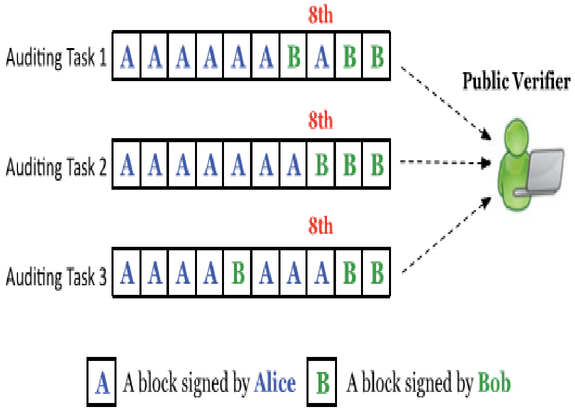

Fig. 1. Alice and Bob share a data file in the cloud, and a public verifier audits shared data integrity with existing mechanisms.

III. PROPOSED SYSTEM MODEL

As illustrated in Fig. 2, the system model in thispaper involves three parties: the cloud server, a groupof users and a public verifier. There are two types ofusers in a group: the original user and a number ofgroup users.

The original user initially creates shareddata in the cloud, and shares it with group users. Boththe original user and group users are members of thegroup. Every member of the group is allowed to accessand modify shared data. Shared data and its verificationmetadata (i.e. signatures) are both stored in the cloudserver. A public verifier, such as a third-party auditor(TPA) providing expert data auditing services or a datauser outside the group intending to utilize shared data, isable to publicly verify the integrity of shared data storedin the cloud server. When a public verifier wishes to check the integrityof shared data, it first sends an auditing challenge tothe cloud server. After receiving the auditing challenge,the cloud server responds to the public verifier with anauditing proof of the possession of shared data. Then,this public verifier checks the correctness of the entiredata by verifying the correctness of the auditing proof.Essentially, the process of public auditing is a challengeand-response protocol between a public verifier and thecloud server [9].

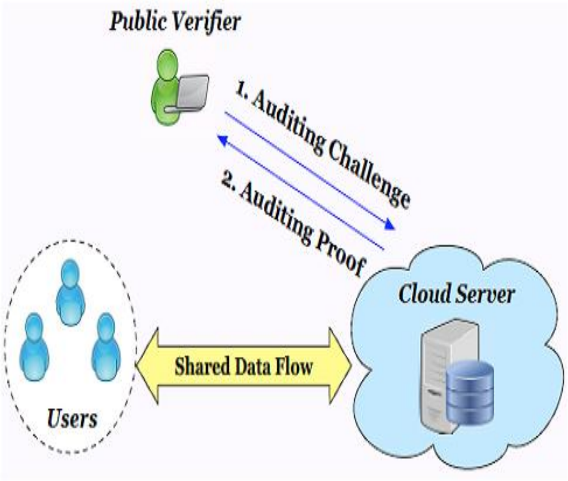

Fig.2 System model of three parties.

A. 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, thecloud 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 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 meta data. Once the public verifier reveals the identity ofthe 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.

B. DESIGN OBJECTIVES

Oruta should be designed to achievefollowing properties:

(1) Public Auditing: A public verifieris able to publicly verify the integrity of shared

datawithout retrieving the entire data from the cloud.

(2) Correctness: A public verifier is able to correctly verifyshared data integrity.

(3) Unforgeability: Only a user inthe group can generate valid verification metadata (i.e., signatures) on shared data.

(4) Identity Privacy: A publicverifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

(5) Data freshness: data freshness is essential to protect against mis-configuration errors or rollbacks caused intentionally.

We can develop an authenticated file system that supports the migration of an enterprise-class distributed file system into the cloud efficiently, transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations.

C. NEW RING SIGNATURE SCHEME

Overview

The design of new homomorphicauthenticable ring signature (HARS) scheme, which is extendedfrom a classic ring signature scheme [15]. The ring signatures generated by HARS are not only able to preserveidentity privacy but also able to support block less verifiability. We will show how to build the privacy preserving public auditing mechanism for shared datain the cloud based on this new ring signature scheme inthe next section.

Construction of HARS

HARS contains three algorithms: KeyGen, Ring Sign and RingVerify. In KeyGen, each user in the group generateshis/her public key and private key. In RingSign,a user in the group is able to generate a signature on ablock and its block identifier with his/her private keyand 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 agiven block is signed by a group member in Ring Verify.

IV. CONCLUSION

The security aspect in cloud is major concern thus we have proposed novel system which can process the request in grouping or batch manner which can enhance performance and efficiency of data transfer/system.The algorithm clearly shows improvements to its predecessor in various fashions like security, transfer of data, scalability and other perspective. To ensure freshness, it is necessary to authenticate not just data blocks, but also their versions. Each block has an associated version counter that is incremented every time the block is modified. This version number is bound to the file-block's MAC: To protect against cloud replay of stale file-blocks (rollback attacks), the counters themselves must be authenticated.

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] 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.

[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] Y. Zhu, H.Wang ―Dynamic Audit Services for Integrity Verification of Outsourced Storage in

Clouds, in Proceedings of ACM SAC 2011, 2011, pp. 1550–1557.

[8] J. Yuan and S. Yu, “Proofs of Retrievability with Public Verifiability and Constant Communication Cost in Cloud,” in Proceedings of ACM ASIACCS-SCC’13, 2013.

[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ―Provable Data Possession at Untrusted Stores,‖ in Proceedings of ACM CCS’07, 2007, pp. 598–610.

[10] 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.

BIODATA

Author

Alabazar Ramesh presently pursuing her M.Tech in CSE, Kottam College of Engineering, Chinnatekuru (V),Kurnool,Andhra Pradesh, India.

Co-Author

V.Sujatha received M.Tech. Presently working as Assistant Professor in Kottam College of Engineering,Chinnatekuru (V), Kurnool, Andhra Pradesh, India.