A Review on Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation

Madhurima Sharma[1], Dr. Vineet Richhariya[2]

[1]M-Tech Research Scholar, [2]Research Guide & HOD, Department of Computer Science & Engineering

Lakshmi Narain College of Technology Bhopal

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 numbers 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. The new public auditing scheme for shared data with efficient user revocation in the cloud is proposed so that the semi-trusted cloud can re-sign the blocks that were previously signed by the revoked user with the valid proxy re-signatures, when a user in the group is revoked.

Keywords - Cloud Data, Public Integrity, Group User Revocation, Shared Dynamic Data.

I. INTRODUCTION

Cloud concept is nothing but the storage service, but it can also share across multiple users. we firstly prioritizes privacy preserving mechanism because while auditing data from cloud services it's not a secured while that private information is publicly protected by cloud service. Specifically, the group signature scheme enables users to anonymously use the cloud resources, and the dynamic broadcast encryption technique allows data owners to securely share their data files with others including new joining users which protects the confidentiality from the revoked users in the dynamic broadcast encryption scheme. We propose that while any user is accessing the data from cloud it must be secured by unauthorized person or hacker. Cloud is un-trusted file storage, so utilize encryption based access control for sharing document in the cloud storage service. User's data is encrypted by using cryptographic technique because unauthorized person can hack the user's private data. In this cryptographic technique we uses different algorithms like signature algorithm, key generation algorithm, ring verify algorithm, etc. these algorithms are used in the cryptographic technique. Users can enjoy high-quality services by migrating local data management systems into cloud servers.

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..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 .

Cloud service providers manage an enterprise-class infrastructure that offers a scalable, secure and reliable environment for users, at a much lower marginal cost due to the sharing nature of resources.

It is routine for users to use cloud storage services to share data with others in a team, as data sharing becomes a standard feature in most cloud storage offerings, including Dropbox and Google Docs. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in an untrusted cloud can easily be lost or corrupted, due to hardware failures and human errors.

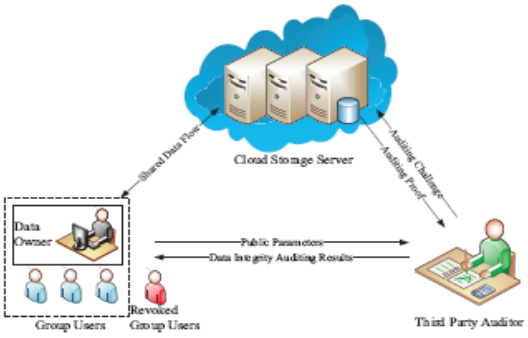

Fig:1.The cloud storage model

To protect the integrity of cloud data, it is best to perform public auditing by introducing a third party auditor (TPA), who offers its auditing service with more powerful computation and communication abilities than regular users.

II. SYSTEM MODEL

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.

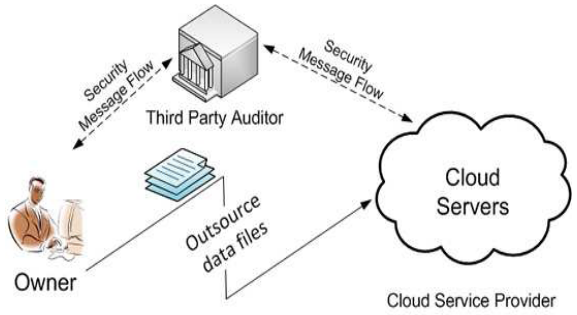

Fig 2: System Model Includes The Cloud Server, The Third Party Auditor And Users.

III. LITERATURE SURVEY

Tao Jiang, Xiaofeng Chen, and Jianfeng Ma [1], proposed a scheme to realize efficient and secure data integrity auditing for share dynamic data with multi-user modification. The scheme vector commitment, Asymmetric Group Key Agreement (AGKA) and group signatures with user revocation are adopt to achieve the data integrity auditing of remote data. Beside the public data auditing, the combining of the three primitive enable their scheme to outsource ciphertext database to remote cloud and support secure group users revocation to shared dynamic data.

Provable data possession (PDP), first proposed by Ateniese et al. [2], 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 verifiability or public auditing. Unfortunately, their mechanism is only suitable for auditing the integrity of static data.

Juels and Kaliski [3] 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.

To support dynamic operations on data, Ateniese et al. [4] presented an efficient PDP mechanism based on symmetric keys. This mechanism can support update and delete operations on data, however, insert operations are not available in this mechanism. Because it exploits symmetric keys to verify the integrity of data, it is not public verifiable and only provides a user with a limited number of verification requests.

Shacham and Waters [6] designed two improved POR schemes. The first scheme is built from BLS signatures, and the second one is based on pseudorandom functions. Wang et al. [3] is able to preserve users' confidential data from the TPA by using random maskings. In addition, to operate multiple auditing tasks from different users efficiently, they extended their mechanism to enable batch auditing by leveraging aggregate signatures [5].

The public mechanism proposed by Wang et al. [6] 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 operations on data, but also to identify misbehaved servers. To minimize communication overhead in the phase of data repair, Chen et al. [7] also introduced a mechanism for auditing the correctness of data with the multi-server scenario, where these data are encoded by network coding instead of using erasure codes. More recently, Cao et al. [8] constructed an LT codes-based secure and reliable cloud storage mechanism. Compare to previous work [6], [7], this mechanism can avoid high decoding computation cost for data users and save computation resource for online data owners during data repair.

Wang et al. utilized Merkle Hash Tree and BLS signatures [9] to support fully dynamic operations in a public auditing

mechanism. Erway et al. [8] introduced dynamic provable data possession (DPDP) by using authenticated dictionaries, which are based on rank information. Zhu et al. exploited the fragment structure to reduce the storage of signatures in their public auditing mechanism. In addition, they also used index hash tables to provide dynamic operations for users.

IV. PROBLEM IDENTIFICATION

For providing the integrity and availability of remote cloud store, some solutions and their variants have been proposed. In these solutions, when a scheme supports data modification, we call it *dynamic* scheme, otherwise *static* one (or limited dynamic scheme, if a scheme could only efficiently support some specified operation, such as append). A scheme is *publicly verifiable* means that the data integrity check can be performed not only by data owners, but also by any third-party auditor. However, the dynamic schemes above focus on the cases where there is a data owner and only the data owner could modify the data.

Disadvantages of existing system:

In the Wang et al. scheme, the user revocation problem is not considered and the auditing cost is linear to the group size and data size.

However, the scheme assumed that the private and authenticated channels exist between each pair of entities and there is no collusion among them. Also, the auditing cost of the scheme is linear to the group size.

However, in Yuan and Yu scheme, the authors do not consider the data secrecy of group users. It means that, their scheme could efficiently support plaintext data update and integrity auditing, while not ciphertext data. In their scheme, if the data owner trivially shares a group key among the group users, the defection or revocation any group user will force the group users to update their shared key. Also, the data owner does not take part in the user revocation phase, where the cloud itself could conduct the user revocation phase. In this case, the collusion of revoked user and the cloud server will give chance to malicious cloud server where the cloud server could update the data as many time as designed and provide a legal data finally.

V. PROPOSED METHODOLOGY

As shown in Fig. below system model consists of 3 entities: the cloud, the public verifier and users who share data in a group. By using data storage and sharing services of cloud user can share data in group, they not only access and modify data but also share the latest version with group. Second entity public verifier is nothing but the group admin or a third party auditor (TPA) who can provide verification services to maintain integrity of data on cloud. During data is uploading one user act as a original user who is original owner of the data and others are group members. Owner of data initially upload data with his signature. Shared data is always divided into small data blocks. When group user performs modification operation on any data block he needs to compute a new signature for the modified block. As data is shared in a group, different blocks may be signed by different users because of modifications by different users. Due to the some reason when any user from group leaves the group or misbehaves, the group needs to revoke this user. The original user acts as the group manager

Proposed System

Proposed system is based on proxy re-signature concepts. Blaze et al first proposed the concept of proxy re-signature in. Proxy signature is a digital signature scheme where original user delegates his signing capability to a proxy signer, and then the proxy signer performs message signing on behalf of the original signer. In simple word it allows semi-trusted proxy to work as a converter of signatures between two users belonging to same group. This concept is the heart of our system which includes below mentioned algorithms.

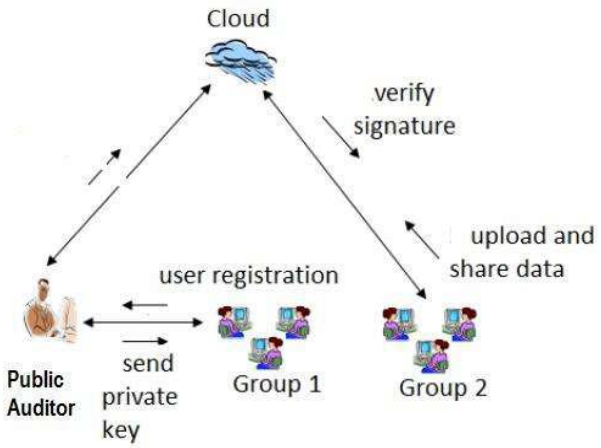

Fig 3 :System Model

1. *KeyGen* - It is a key generation algorithm, it is the one of the important step where public key and private key for each and every group user is created.

2. *ReKey* - In this step cloud computes a re-signing key for all group users.

3. *Sign* - When original user upload his data on cloud and shared it within group, he attach his signature on each data block as Sign. Also when other member from a group modified data shared by original user he computes his signature on modified data block.

4. *ReSign* - In this step, when user revoked from the system, cloud re-signs the data blocks which were signed by revoked user by using his resigning key.

5. *ProofGen* - Data integrity is verified by using challenge-and-response protocol between the cloud and a public verifier. In ProofGen step cloud can generate a proof of possession for shared data.

6. *ProofVerify* - In this step a public verifier can check the correctness of a proof responded by the cloud. It verifies data without retrieving it. Here it uses HAPS (a homomorphic authenticable proxy re-signature scheme) with Blockless Verifiability.

System Architecture

Architecture of proposed cloud public auditor is shown in Fig. It consists of different modules which are responsible for different process which are required for efficient user revocation and to check correctness of data.

1) *User Module:*

User module is consisting of small sub-modules like Registration, File Upload, Download, Re-upload and Unblock. When user join group he can register by using web server. After successful registration user can able to upload his data on cloud. He can select data file and upload it on cloud. RSA algorithm converts plain text into cipher text and stored it into cloud database .After successful data uploading generated private key is provided to the user. If user or group member want to access that data they need to download it using their public key.

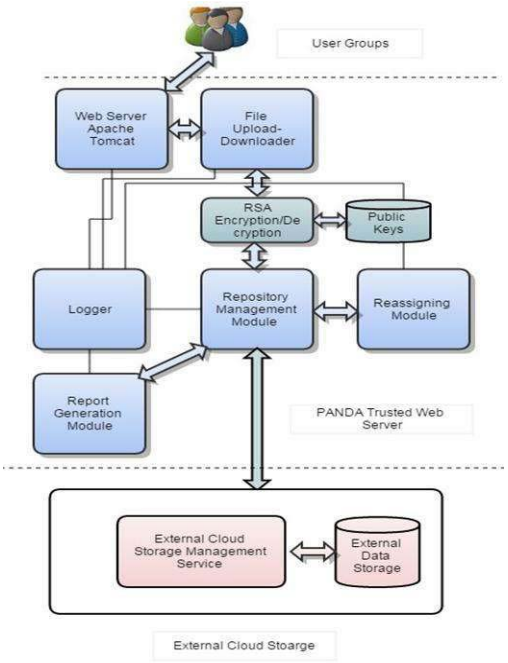

Fig 4 :Architecture of Cloud Public Auditor

2) *Repository Management Module:*

Repository management module is responsible for storing user information and their public key. During verification whatever information required related to data blocks is stored in the repository module.

3) *Reassigning Module:*

This module is work on concept of proxy resignature. New signature is attaché to data blocks which were previously signed by revoked user.

4) *Report Generation Module:*

Report generation is attached to logger which maintaining logs of daily activities. These logs are used by report generation module for generation of scheduled reports. Generated reports contain information about users (revoked users and active users).They also produced graphical reports representing performance of system against time used and scalability.

5) *External Cloud Storage Management:*

For security reasons, it is required to store data and keys separately on different servers by cloud service providers. Therefore, in our mechanism, we assume that the cloud has a server to store shared data, and has another server to manage resigning keys.

VI. CONCLUSION

In this review paper, the first privacy-preserving public auditing mechanism for shared data in the cloud. With Oruta, the TPA is responsible for to audit the integrity of shared data, till cannot distinguish who is the signer on each block which can preserve identity privacy for users. We utilizing, privacy preserving who shared the data in the cloud storage service with the help of the ring signature and Homomorphic authentication ring signature. And we utilize ring signature to construct the HARS, so TPA will protected from unauthorized user. It will easily audit the integrity of shared data. Our future work is how to audit shared data with dynamic members while users sharing the data it will be safe from the TPA. An interesting problem in our future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

REFERENCES

[1] Tao Jiang, Xiaofeng Chen, and Jianfeng Ma, “Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation” 10.1109/TC.2015.2389955, IEEE.

[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 Conference on Computer and Communications Security (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 International Conference on Computer Communications (INFOCOM)*, 2010, pp. 525–533.

[4] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security (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. In- ternational Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer-Verlag, 2003, pp. 416–432.

[6] H. Shacham and B. Waters, "Compact Proofs of Retrievability," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*. Springer- Verlag, 2008, pp. 90–107.

[7] 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 (SAC)*, 2011, pp. 1550–1557.

[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 534–542.

[9] D. Boneh, B. Lynn, and H. Shacham, "Short Signature from the Weil Pairing," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*. Springer-Verlag, 2001, pp. 514–532.

[10] D. Boneh and D. M. Freeman, "Homomorphic Signatures for Polynomial Functions," in *Proc. International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer-Verlag, 2011, pp. 149–168.

[11] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practical Short Signature Batch Verification," in *Proc. RSA Con- ference, the Cryptographers' Track (CT-RSA)*. Springer-Verlag, 2009, pp. 309–324.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," in *Proc. ACM Conference on Computer and Communications Security (CCS)*, 2006, pp. 89–98.