A Secure Auditing Mechanism for Sharing Data on Cloud Storage

A. Mangalakshmi
Assistant Professor, Dept of CSE,
Christ College of Engg. & Tech,
Pondicherry, India.

M. Achudhan
Final Year M.Tech, Dept of CSE,
Christ College of Engg. & Tech,
Pondicherry, India.

Abstract: **In Cloud Storage, users can remotely store their data and enjoy the high quality applications and services on-demand. Existing Oruta mechanisms has been designed to allow both data owners and public verifiers (TPA) to audit cloud data integrity without retrieving the entire data from the cloud server. Since there is no central authority, Data Management and the Services are not Trust Worthy. In the proposed model, Encryption schemes came forward to protect the data functionality of the storage system over the encrypted data & the Data Owner Sends the Data and it is Stored by splitting the Data using Merkle Hash Tree Algorithm and Verification Process is achieved for Data Safety, so that the data leakage can be prevented, traceability and data freshness was efficiently achieved with identity of each block of data. Key escrow mechanism helps us to recover the lost key during system crash which makes key management effective Authorized users can change the data which is updated in a separate Copy as a replica , then it is updated once Owner Authenticates the modifications.**

Keywords: MHT- Merkle hash tree; EKA - Key escrow algorithm; TPA Third party auditor; BT –Batch auditing.

I. INTRODUCTION

Cloud computing is Internet ("cloud") based development and use of computer technology ("computing"). It is a style of computing in which various scalable and often virtualization resources are provided as a service over the internet. One of the most fundamental services offered by cloud providers is storage. Let us consider a practically a application. The concept of cloud computing has evolved from the concepts of cluster, grid, and utility computing and providing software as a service (SaaS). Cluster and grid computing leverage the use of many computers in parallel to solve complex problems. Utility and SaaS provides the computing resources as a service with a notion of pay per use. Cloud Computing provides us a means by which we can access the applications as utilities, over the Internet [9]. It allows us to create, edit and customize applications from the online. A company allows its staffs in the same group or department to store and share files in the cloud server. However, it also makes a significant risk to the confidentiality of the stored files. Specifically, the cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and confidential, such as business plans. To maintain data security, a basic solution is to perform cryptographic methodologies on data files, and then uploaded onto the cloud server.

A. Characteristics of Auditing Protocols:

Peaked Private: The TPA should not gain knowledge of the original user data during the auditing process.

Data Dynamism: The clients must be able to perform operations on data files like insert, alter and delete while maintaining data correctness.

Verifiability: Anyone, not just the clients, must be allowed to verify the integrity of data.

Block Free Validation: Challenged file blocks should not be retrieved by the verifier during verification process.

No Restriction of multi-query: The verifier may be allowed to use unlimited number of queries in the challenge-response protocol for data verification.

B. Attacks in the cloud data storage are:

Snooping: Snooping is to peep into others non-public data. It is a better way to send and retrieve the data over a secure transmission line.

Cloud Authentication: The clients can obtain others authorization and may misuse the files. Hence it is necessary to protect one’s unique identity. The unauthorized clients must not be enter into others account and delete the data.

Key Management: The cryptographic keys have to be managed in the cloud environment but this key management must be user friendly.

Data Inconsistency: Data inconsistency occurs when it is transmitted between the user and the cloud server. The best way is to encrypt the data from owner’s side.

Performance: A durable security approach is necessary for encrypting and decrypting the files to and from the cloud but it should keep the user’s performance integral [12].

C. Auditability Schemes

Auditing reduces risk for customers as well as give incentives to providers to improve their services. Audit ability falls under two categories as follows when we consider the available schemes in audit ability: private audit ability and public audit ability. Even though schemes with private auditability can attain higher scheme efficiency, public auditability permits anyone, not just the client (data owner), to deal with the cloud server for correctness of data storage while keeping no private information. Then, clients are able to pass on the evaluation of the service performance to an independent third party auditor (TPA), without giving their computation resources. So we can denote the types of auditing protocols as Data Owner Auditing and Third Party Auditing. According to the methods of data storage auditing methods can be categorized into three: Message Authentication Code (MAC) - based methods, AES- based Homomorphic methods and Boneh-Lynn-Shacham signature (BLS) – based Homomorphic methods. The challenging issues of data storage auditing include Dynamic Auditing, Collaborative Auditing and Batch Auditing. We need to meet the three performance criteria when comes to designing of auditing protocols as: low storage overhead, low communication cost and low computational complexity.

The schemes available in public auditability works related to public auditability in cloud. The provably not privacy preserving but they lead to the development of efficient privacy-preserving methodologies

D. *Third Party Auditing (TPA)*

Cloud consumers save data in cloud server so that security as well as data storage correctness is primary concern. The data owners having huge amount of outsourced data and the task of auditing the data correctness in a cloud environment can be difficult and expensive for data owners.

To support third party auditing where user safely delegate in integrity checking tasks to third party auditors(TPA)this scheme can almost guarantee the simultaneous localization of data error(i.e. the identification of misbehaving servers).

A novel and homogeneous structure is introduced to provide security to different cloud types. To achieve data storage security, BLS (Bonch-Lynn-Sachems) algorithm is used to signing the data blocks before outsourcing data into cloud. Reed Solomon technique is used for error correction and to ensure data storage correction.

II. SYSTEM DESIGN

In the existing system the data shared in the cloud is based on the ring signature methodology which randomly chooses the identity of the user present in that particular group while uploading the files

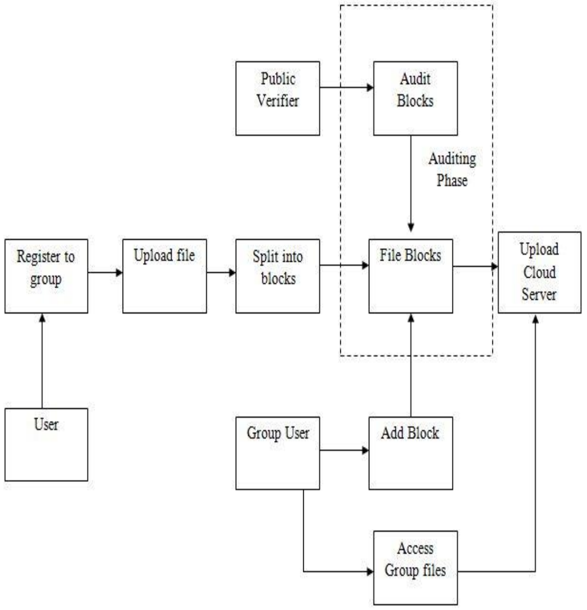

Fig. 1. System Architecture

But this system is meant for a small group (i.e. group is consist of less number of user).Whenever the number of members in the group gets increased then the system fails to generate the ring signature efficiently due to some complexities. The freshness of the data shared in the cloud is not verified here which makes a big drawback in this system. Existing advantages are being implemented in our proposed work so that we can improve performance efficiency in sharing information in the cloud.

III. METHODOLOGY

A. *User Module*

Registration: In this module each user registers his user details for using files. Only registered user can able to login in cloud server.

File Upload**:** In this module user upload a block of files in the cloud with encryption by using his secret key. This ensures the files to be protected from unauthorized user.

Download**:** This module allows the user to download the file using his secret key to decrypt the downloaded data of blocked user and verify the data and Re-upload the block of file into cloud server with encryption .This ensure the files to be protected from unauthorized user.

Re-upload**:** This module allow the user to Re-upload the downloaded files of blocked user into cloud server with resign the files (i.e.) the files are uploaded with new signature for security.

B. *Data Dynamics*

Data dynamics means after clients store their data at the remote server, they can dynamically update their data at later times. At the block level, the main operations are block insertion, block modification and block deletion.

Block Insertion: The Server can insert anything on the client’s file.

Block Deletion: The Server can delete anything on the client’s file.

Block Modification: The Server can modify anything on the client’s file. The above data modifications are updated on the original files only after authenticating the user identity.

C. Metadata Key Generation

Let the verifier V wishes to the store the file F. Let this file F consist of n file blocks. Initially preprocess the file and create metadata to be appended to the file. Let each of the n data blocks have m bits in them. A typical data file F which the client wishes to store in the cloud.

Each of the Meta data from the data blocks mi is encrypted by using a RSA algorithm to give a new modified Meta data Mi. Without loss of generality Show this process. The encryption method can be improvised to provide still stronger protection for Client’s data. All the Meta data bit blocks that are generated using the procedure are to be concatenated together. This concatenated Meta data should be appended to the file F before storing it at the cloud server. The file F along with the appended meta data with the cloud.

D. File Verification module

The public verifier is able to correctly check the integrity of shared data. The public verifier can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have been re-signed by the cloud.

E. Data Freshness Verification

Extend the construction of oruta with data freshness in the authenticated file system that verify the freshness of any data retrieved from the file system while performing typical file system operations. Freshness ensures that the latest version of the data is always retrieved (and thus prevents rollback attacks reverting the file system state to a previous version. Another challenge is efficient management and caching of the authenticating information. Freshness verification should be extremely efficient for existing file system operations and induce minimal latency. 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.

IV. CONCLUSION

In Cloud Storage, day by day user level was increased and the on-demand high quality services. Encryption schemes came forward to protect the data functionality of the storage system over the encrypted data & the Data Owner Sends the Data. To avoid the data duplication and data lost we utilized Merkle hash tree for data splitting in an advanced manner. Homomorphic ring signature algorithm is implemented in order to generate signature for larger number of users in the group Here the third party auditor is utilized as an interface between the CSS and Client so that the Data can’t be viewed by the TPA and Verification Process is achieved for Data Safety, so that the data leakage can be prevented, traceability and data freshness was efficiently achieved with identity of each block of data. Using key escrow mechanism recovery of lost key can be obtained properly. Key management is maintained efficiently. Privacy of data is achieved highly since the auditor is unaware of the contents stored in cloud.

REFERENCES

[1] Boyang Wang, Baochun Li , Hui Li ,” Oruta: Privacy-Preserving Public Auditing For Shared Data in The Cloud,” IEEE Trans on Cloud Computing, Vol.2, 2014.

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

[3] 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,” in Proceedings of the 26th Annual ACM Symposium on Applied Computing (SAC '11), March 2011, pp. 1550–1557.

[4] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, “Dynamic Secure Cloud Storage with Provenance,” in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[5] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, “SPICE - Simple Privacy-Preserving Identity-Management for Cloud Environment,” in Applied Cryptography and Network Security - ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

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

[7] B. Wang, S. S. M. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[8] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, “Dynamic Secure Cloud Storage with Provenance,” in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[9] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in Proceedings of Advances in Cryptology - EUROCRYPT ’03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[10] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and Efficient Key Management for Access Hierarchies,” ACM Transactions on Information and System Security (TISSEC), vol. 12, no.3,2009.