International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

# Enhanced Technique for Privacy Preserving Public Auditing for Shared Data in Cloud

Jyothi Kannan
Computer Science
Toc H Institute of Science and Technology
Ernakulam, Kerala, India

Leda Kamal
Computer Science
Toc H Institute of Science and Technology
Ernakulam, Kerala, India

***Abstract*— Cloud computing, often referred to as simply "the cloud," is the delivery of on-demand computing resources everything from applications to data centers over the Internet on a pay-for-use basis. Cloud storage is a very important service of cloud computing. It provides owners to maneuver information from their native computing systems to the cloud. More and more owners begin to store the data in the cloud. By utilizing Cloud storage, users can access applications, services, software every time they requires within the internet. Users can put their data remotely to cloud storage and get advantageous asset of on-demand services and application from the resources. The cloud will need to have to make certain data integrity and security of data of user. In Cloud Computing Public auditing on the integrity of shared data is extremely major problem, because the public auditing scheme will predictably reveal confidential information, and identity privacy to public verifiers. This survey, propose a fresh scheme called a novel privacy-preserving mechanism which supports public auditing on shared data stored in the cloud. Specifically, this scheme takes benefit of ring signatures to compute verification metadata. The ring signature scheme is necessary to audit the correctness of shared data. With this particular mechanism, the identity of the signer on each block in shared data is kept secret or private from public verifiers. Public verifiers are person who are able the whole file. Furthermore, this mechanism has the capacity to perform multiple auditing tasks simultaneously as opposed to single auditing task using Batch Auditing. This survey shows the effectiveness and efficiency of our mechanism when auditing shared data integrity.**

***Key Words: Cloud Computing, Public Auditing, Privacy Preserving, Ring Signature, Batch Auditing***

## I. INTRODUCTION

Cloud computing may be a novel computing model that permits convenient and on-demand access to a shared pool of configurable computing resources. Auditing services are extremely essential to make sure that the info is properly hosted within the cloud. Cloud storage, a crucial service of cloud computing, permits users to maneuver knowledge of their native storage systems to the cloud and enjoy the on-demand top quality cloud services. It offers nice convenience to users since they do not have to be compelled to care regarding the complexities of direct hardware and software package managements. The integrity of information in cloud storage, after all, is susceptible to agnosticism and scrutiny, as knowledge hold on in the cloud will simply be lost or corrupted because of the unavoidable human errors and software failures. To create this matter even worse, cloud service suppliers could also be reluctant to inform users concerning these knowledge errors so as to keep up the name of their services and avoid losing profits. Therefore, the integrity of cloud knowledge ought to be verified before any knowledge utilization, like search or computation over cloud knowledge. The conventional approach for checking data correctness is to retrieve the whole data from the cloud, so verify data integrity by checking the correctness of signatures (e.g., RSA) or hash values (e.g., MD5) of the whole data. Certainly, this conventional approach is able to successfully check the correctness of cloud data. However, the efficiency of applying this ancient approach on cloud data is unsure. The most reason is that the dimensions of cloud knowledge are giant generally. Downloading the whole cloud knowledge to verify knowledge, integrity can value or perhaps waste user's amounts of computation and communication resources, particularly once knowledge are corrupted in the cloud. Besides, several uses of cloud data (e.g., machine learning and data mining) do not essentially would like users to download the entire cloud data to native devices. It's as a result of cloud providers, like Amazon, can give users computation services directly on huge data that already existed with the cloud [1].

Today's world depends on cloud computing to store completely different information like their public additionally as some personal info that is required by the user itself or several other persons. A cloud service is any service wanted to, its users by cloud. As cloud computing is available in service you can find some drawbacks such as for example privacy of the user's data, security of user data is vital aspects. Cloud computing is demand on shared computing resources. With the continual development of cloud computing technology, its appliance is additional and additional wide. Now at times, cloud computing is usually used in completely different synonymous like cluster computing, grid computing, distributed computing, involuntary computing. Privacy is a crucial issue in cloud computing, at any time, user needs to form use of information that involves individual sensitive data. With the fast development of web technology, privacy conserving information publication has become one in every of the foremost necessary analysis topics and become a heavy concern in publication of private information in recent years. For all that, for data owners who are becoming progressively involved regarding their privacy of the info that contains some personal data regarding people [2].

*A. Justification*

Many public auditing mechanisms are used to preserve privacy of shared data in the cloud. These mechanisms are Provable Data Possession, Compact Proof of Retrievability, Dynamic Provable Data Possession, and Remote Data Checking and so on. These mechanisms are proposed to permit not solely a knowledge owner it, however additionally a public supporter to with efficiency, perform integrity checking while not downloading the whole knowledge from the cloud, that is noted as public auditing. With these mechanisms, knowledge is split into several tiny blocks, wherever every block is severally signed by the owner; and a random combination of all the blocks rather than the whole knowledge is retrieved throughout integrity checking. A public supporter may well be a knowledge user (e.g., researcher) who would love to utilize the owner's knowledge via the cloud or a third party auditor (TPA) who will offer professional integrity checking services. Moving a discovery, Wang et al. [1] designed an advanced auditing mechanism (WWRL), so throughout public auditing on cloud knowledge, the content of personal knowledge, happiness to a private user isn't disclosed to any public verifiers. Unfortunately, current public auditing solutions mentioned higher than solely concentrate on personal knowledge in the cloud. The foremost partaking options that Inspire cloud storage is sharing of information among multiple users. Therefore, it's additionally necessary to confirm the integrity of shared knowledge within the cloud is correct. Existing public auditing mechanisms will truly be extended to verify shared knowledge integrity. However, a replacement vital privacy issue introduced within the case of shared knowledge with the use of existing mechanisms is that the leak of identity privacy to public verifiers.

The integrity of shared knowledge with these existing mechanisms can inevitably reveal confidential information of identity privacy to public verifiers. Public auditing mechanisms will really be extended to verify shared knowledge integrity. However, a fresh important privacy issue introduced in the case of shared knowledge with the utilization of existing mechanisms is that the outflow of identity privacy to public verifiers. Once a block during this shared file is changed by a user, this user must sign the new block mistreatment his/her non- public key. Eventually, completely different blocks are signed by different users because of the modification introduced by these 2 totally different users. The most drawbacks with this approach is that it needs all the users mistreatment designed hardware, and wishes the cloud provider to maneuver all the present cloud services to the trusty computing the supporter doesn't need to transfer all the blocks to examine the integrity of knowledge. Non-malleability indicates that an somebody cannot generate valid signatures on capricious blocks by linearly combining existing signatures. Failing to preserve identity privacy on shared knowledge throughout public auditing can reveal significant guidance to public verifiers. In order to guard this guidance, it is essential and demanding to preserve identity privacy from public verifiers throughout the public auditing. The projected System to unravel the on top of privacy issue on shared knowledge. This is often a completely unique privacy protective public auditing mechanism.

The shared file is split into a several small individual blocks, where each block is independently signed by one of many two users with Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing scheme. Once a block in this shared file is modified by an individual, that specific user must sign the newest block using his/her secret private key. Finally, different blocks are signed by various users as a result of modification introduced by these different users. Then, to be able to correctly audit the integrity or correctness of the whole data, a public verifier needs to find the suitable public key for every single block. Specifically, as shown in Fig. 1, after performing several auditing tasks, this public verifier can first learn that Alice might be a more important role in the group because a lot of the blocks in the shared file are usually signed by Alice; on one other hand, this public verifier also can easily deduce that the eighth block may contain data of an increased value (e.g., one last bid in an auction), since this block is often modified by both different users. To be able to protect this confidential information, it is important and critical to preserve identity privacy from public verifiers during public auditing. Consequently, this public verifier will inevitably learn the identity of the signer on each block because of the unique binding between an identity and a public key via digital certificates under public key infrastructure (PKI) [1].

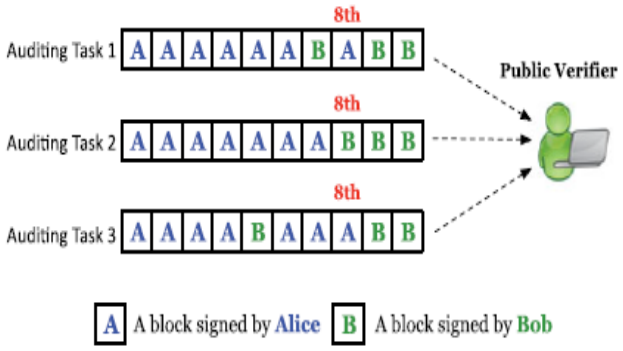


Fig -1: Alice and Bob share a data file in the cloud, and a public verifier audits shared data integrity with existing mechanisms.

## II. RELATED WORKS

Cloud computing is that the new term for the long-dreamed vision of computing as a utility. The cloud provides convenient, on-demand network access to a centralized pool of configurable computing resources which will be quickly deployed with nice potency and bottom management overhead. Cloud service suppliers provide users economical and climbable knowledge storage services with a far lower incremental cost than ancient approaches. The normal approach for checking data correctness is to retrieve the whole knowledge from the cloud, and so verify knowledge integrity by checking the correctness of the signatures or hash values of the whole knowledge. Certainly, this typical approach is ready to with success check the correctness of cloud knowledge. For all that, the efficiency of exploitation this ancient approach on cloud knowledge is unsure. The most reason is that the size of cloud knowledge is massive

generally. Downloading the whole cloud knowledge to verify knowledge integrity will price or maybe waste user's amounts of computation and communication resources, especially once knowledge are corrupted within the cloud. There is a unit many existing mechanisms are wont to describe the construct of privacy conserving in cloud shared knowledge. They are explained here,

Provable data possession (PDP), proposed by Ateniese et al. [3], allows a verifier to check the correctness of a clients data stored on 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 auditing. Unfortunately, their mechanism is only suitable for auditing the integrity of personal data. Provable Data Possession at untrusted Stores Some method to provide data availability and integrity that follow traditional cryptographic technologies based on signature scheme and a hash function. It cannot work on outsourced data and did not suitable for a large size file.

Juels and Kaliski [4] describe a Proofs of Retrievability (POR) model, 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. A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a large file (or bitstring) F. They explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F.

To support dynamic data, Ateniese et al. [5] 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. Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its clients (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form. In Scalable and Efficient Provable Data Possession, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e., it efficiently supports operations, such as block modification, deletion m, and append.

Wang et al. [6] utilized Merkle Hash Tree and BLS signatures [7] to support dynamic data in a public auditing mechanism. To achieve efficient data dynamics, we improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure. This Merkle tree based structure has two distinctive features compared to other authenticated file systems: (1) Support for existing file system operations: that maintains a balanced binary tree over the file system directory structure to efficiently support existing file system calls; and (2) Support for concurrent operation: the Merkle tree supports efficient updates from multiple clients operating on a file system in parallel. It also optimizes for sequential file block accesses: sequences of identical version counters are compacted into a single leaf.

Erway et al. [8] introduced dynamic provable data possession (DPDP) by using authenticated dictionaries, which are based on rank information. Dynamic provable data possession (DPDP), which extends the PDP model to support provable updates on the stored data. Given a file F consisting of n blocks, they define an update as either insertion of a new block (anywhere in the file, not only append), or modification of an existing block, or deletion of any block. Therefore our update operation describes the most general form of modifications a client may wish to perform on a file. DPDP solution is based on a new variant of authenticated dictionaries, where they use rank information to organize dictionary entries. Thus, they are able to support efficient authenticated operations on files at the block level, such as authenticated insert and delete. They prove the security of our constructions using standard assumptions. They also show how to extend our construction to support data possession guarantees of a hierarchical file system as well as file data itself.

Zhu et al. [9] 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 on data. Our audit service, constructed based on the techniques, fragment structure, random sampling and index- hash table, can support provable updates to outsourced data, and timely abnormal detection. In addition, they propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches, but also show our audit system has a lower computation overhead, as well as a shorter extra storage for audit mete-data.

The public mechanism proposed by Wang et al. [10] and its journal version [11] are able to preserve user's confidential data from a public verifier by using random masking. The main contributions are It motivate the public auditing system of data storage security in cloud computing and provide a

privacy-preserving auditing protocol. This scheme enables an external auditor to audit users cloud data without learning the data content, To the best of our knowledge, This scheme is the first to support scalable and efficient privacy-preserving public storage auditing in cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner and It prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state of the art.

Chen et al. [12] also introduced a mechanism for auditing the correctness of data under the multi-server scenario, where these data are encoded in network coding instead of using erasure codes. Remote Data Checking (RDC) is a technique by which clients can establish that data outsourced at untrusted servers remains intact over time. RDC is useful as a prevention tool, allowing clients to periodically check if data has been damaged, and as a repair tool whenever damage has been detected. Initially proposed in the context of a single server, RDC was later extended to verify data integrity in distributed storage systems that rely on replication and on erasure coding to store data redundantly at multiple servers. Recently, a technique was proposed to add redundancy based on network coding, which offers interesting tradeoffs because of its remarkably low communication overhead to repair corrupt servers. Unlike previous work on RDC which focused on minimizing the costs of the prevention phase, it takes a holistic look and initiate the investigation of RDC schemes for distributed systems that rely on network coding to minimize the combined costs of both the prevention and repair phases.

Cao et al. [13] constructed an LT codes-based secure and reliable cloud storage mechanism. With the increasing adoption of cloud computing for data storage, assuring data service reliability, in terms of data correctness and availability, has been outstanding. While redundancy can be added into the data for reliability, the problem becomes challenging in the pay- as-you-use cloud paradigm where it always want to efficiently resolve it for both corruption detection and data repair. Prior distributed storage systems based on erasure codes or network coding techniques have either high decoding computational cost for data users, or too much burden of data repair and being online for data owners. In this paper, it designs a secure cloud storage service which addresses the reliability issue with near-optimal overall performance.

## III. PROBLEM STATEMENT

### A. System Model

The cloud data storage majorly containing three entities, as given in Fig.2: the cloud server, the third party auditor (TPA) and users. Two kinds of users are within the group. They're the Original user and the members of the group. Group members are allowed to get into and modify shared data produced by the original user on the basis of the access control policies. Shared data and its verification information (i.e. Signatures) are generally stored in the cloud server. The third party auditor has the capacity to verify the integrity of shared data in the cloud server with respect to the group members [1].

In this paper, we simply how exactly to audit the integrity of shared data in the cloud with static groups and how to the integrity of shared data in the cloud with dynamic groups. In Static group, it is pre-defined before shared data is established in the cloud and the membership of users in the group is not changed during data sharing. The original user is accountable for deciding who has the capacity to share her data before outsourcing data to the cloud. In dynamic group, a new user could be added to the group and a preexisting group member may be revoked during data sharing while still preserving identity privacy [1] [2].

When a user (either the original user or a group user) wishes to check the integrity of shared data, she first sends an auditing request to the TPA. After receiving the auditing request, the TPA generates an auditing message to the cloud server, and retrieves an auditing proof of shared data from the cloud server. Then the TPA verifies the correctness of the auditing proof. Finally, the TPA sends an auditing report to the user based on the result of the verification [1][2].

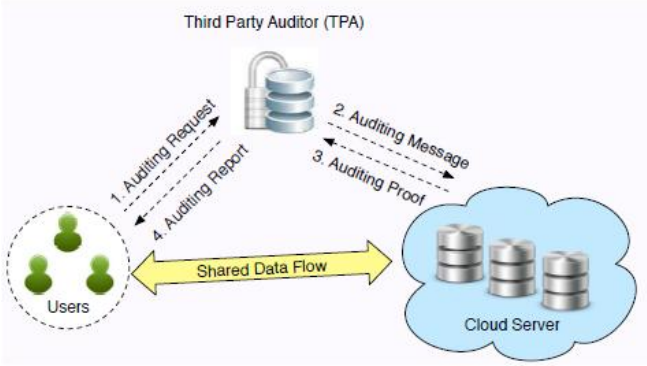


Fig-2: System model includes the cloud server, the third party auditor and users.

### B. Threat Model

- Integrity Threats: Two forms of threats associated with the integrity of shared information area unit potential. First, associate degree person might attempt to corrupt the integrity of shared information and forestall users from victimization information properly. Second, the cloud service supplier might unknowingly corrupt (or even re- move) information in its storage attributable to hardware failures and human errors. Creating matters worse, so as to avoid jeopardizing its name, the cloud server supplier is also reluctant to tell users regarding such corruption of knowledge.
- Privacy Threats: The identity of the signer on every block in shared information is non-public and confidential to the cluster. Throughout the method of auditing, a semi-trusted TPA, who is just accountable for auditing the integrity of shared information, might try and reveal the identity of the signer on every block in shared information supported verification info. Once the TPA reveals the identity of the signer on every block, it will simply distinguish a high-value target (a explicit user within the cluster or a special block in shared data) [1].

### C. Design Objectives

This mechanism should be designed to achieve following properties: (1) Public Auditing: The third party auditor is able to verify the integrity of shared data for a group of users without retrieving the entire data. (2) Correctness: The third party auditor is able to correctly detect whether there is any corrupted block in shared data. (3) Unforgeability: Only a user in the group can generate valid verification information on shared data. (4) Identity Privacy: During auditing, the TPA cannot distinguish the identity of the signer on each block in shared data [1].

## IV. PROPOSED SYSTEM

The mechanism to solve the privacy issue on shared data, a novel privacy-preserving public auditing mechanism. Cloud Service suppliers transferring the information to cloud user from cloud server. Currently the TPA has to be compelled to check the integrity of transferred information. The method are going to be like this, The TPA can collect the receive information and send information to verify. If each information same, then there's no violation within the information integrity. Much this is often uphill for the big information. Additionally TPA additionally a external entity once more if we have a tendency to offer full set of information once more data integrity question can rise in TPA finish. For the multiple cloud and multiple users we'd like multiple auditing known as batch auditing. We'd like to implement the new technique with Homomorphic authenticator and therefore the additive combination signature methodology. To construct the public auditing mechanism we will extend the ring signature scheme.The concept of ring signatures is first proposed by Rivest et al. in 2001. With ring signatures, a verifier is convinced that a signature is computed using one of group member's private keys, but the verifier is not able to determine which one. 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 [14] [15].

Mainly four components are included in this mechanism. They are Client, Owner, Cloud server, and Third Party Auditor. But, here Third Party auditor will carrying important role in this mechanism. The owner provides the data's to the consumer through the Cloud server. The integrity of the info within the Cloud server is in punctuation mark. The owner will verify the integrity of knowledge in Cloud server by auditing. However having auditing within the owner is price effective and it results in headache to the owner. The answer is to possess a 3rd Party Auditor to verify the integrity of knowledge within the Cloud server. The Third Party Auditor could be a neutral entity to the Cloud Server and also the Owner. On behalf of owner the TPA can verify the owner's knowledge storage and security method. TPA ought to be a trusty entity. However trusting a 3rd party isn't sensible. to make sure the privacy of the info, the info content isn't out there to the Third Party Auditor. The TPA verifies the encrypted knowledge in order that the privacy of the info is ensured. this will be done exploitation Homomorphic authentication. The information is generated exploitation Homomorphic authentication. The TPA disputes the Cloud server for the proof of knowledge integrity. The Cloud server provides the proof that is verified against the owner's information.

The privacy-preserving public auditing using signatures include three algorithms as previously mentioned here:

- KeyGen: Each user in the group generates their public key and private key.
- Ring Sign: User in the group relates to sign a block with her private key and all group members‟ public keys.
- Ring verify: The verifier can be used to test if the given block is signed by the group member.

The ring signatures for public auditing include following steps for auditing

- Each user generates its public and private key.
- A user in the group signs a block with her private key and all group member‟s public key. Pk1 is public key of the user; Sk1 is private key of the user; (Pk1 ……Pkd ) is „d‟ number of users of data block m Є ZP.
- User randomly selects data block m Let id is identifier of data block m.
- User ui encrypts with all user‟s public key, so only private key of the group user‟s i Є [1,d] would be able to decrypt it. This ensures privacy of data.
- To ensure auditing by third-party user (ui), where i Є [1,d] signs the data block using his private key.
- TPA (Third-party auditor), using a Pk1 …….Pkd Where d is number of users in the group.

TPA calculates signature of data blocks but unaware of who sign it .Therefore calculates signature using each given public key (Pk1 ……Pkd ) from this set. Gsign = signature set for (Pk1 ……Pkd ) If Gsign = {sign1,sign2,……signd} matches with original sign then data block is intact. By using this scheme user can also do the data dynamic operation. As there is group of users which share their data to each other , they can do modification on data of CS. [16] [17].

### A. Scheme construction

- Data Owner: The Owner sends the query to the server. Based on the query the server sends the corresponding file to the Owner. Before this process, the client authorization step is involved.
- Third Party Auditor : Third Party Auditor has capabilities to manage or monitor outsourced data under the delegation of data owner.
- User: Users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.
- Data Sharing: Here only consider how to audit the integrity of shared data in the cloud with static groups.It means the group is pre-defined before shared data is created in the cloud and the membership of users in the group is not changed during data sharing. The original user is responsible for deciding who is able to share her data before outsourcing data to the cloud. Another interesting problem is how to audit the integrity of shared data in the cloud with dynamic groups.a new user can be added into the group and an existing group member can be revoked during data sharing while still preserving identity privacy.

### *B. Supports for data dynamics*

As information in Cloud is dynamic, static auditing isn't enough. A dynamic auditing is required to verify the information integrity of the dynamic data. However as information square measure dynamic in cloud, it's dangerous to own associate degree auditing with efficiency. Server will enforce Replay attack and forge attack to fail the auditing method. The dynamic operations embrace modification, insertion and deletion. Whenever dynamic operation is performed, the owner sends the update message to the auditor representing the indicator of that message. The Auditor updates the table. The message m and therefore the tag square measure replaced by the new message and tag in message modification. The new message m and new tag square measure inserted in insertion operation. The message m and tag square measure deleted from the index table and every one the entries below the deleted message move upwards. After performing updates in the table, the auditor conducts the data integrity test for the updated data. Auditor sends the result to the owner and he deletes the local copy of updated data [18] [19].

### *C. Batch Auditing*

In Batch Auditing, Owner can use more than one Cloud server to store the data. Similarly the more than one owner can use the same Cloud server to store their data. If the owner user more than one cloud the auditor select the cloud severs to check the data integrity [20].

## V. CONCLUSION

The proposed system is really a solution for privacy dilemma of the shared data in the cloud. The perfect solution is really a privacy-preserving public auditing mechanism for shared data in the cloud. Here utilize ring signatures to make homomorphic authenticators, so that the public verifier has the capacity to audit shared data integrity without retrieving the whole data, yet it cannot distinguish who's the signer on each block. To enhance the efficiency of verifying multiple auditing tasks, it further extends this mechanism to guide batch auditing. This mechanism is provide reduced signature storage and also supports dynamic operations on shared data.

## REFERENCES

1. B. Wang, B. Li, and H. Li, Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,Proc. IEEE Fifth Intl Conf.Cloud Computing, pp. 295-302, 2012.
2. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A.Konwinski, G. Lee,D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, A View of Cloud Computing ,Comm. ACM,vol. 53, no. 4, pp. 50-58, Apr. 2010.
3. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, Provable Data Possession at Untrusted Stores ,Proc. 14th ACM Conf. Computer and Comm.Security (CCS 07), pp. 598-610, 2007.
4. A. Juels and B.S. Kaliski, PORs: Proofs of Retrievability for Large Files ,Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), pp. 584-597, 2007.
5. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, Scalable and Efficient Provable Data Possession ,Proc. Fourth Intl Conf. Security and Privacy in Comm. Networks (SecureComm08), 2008.
6. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing ,Proc. 14th European Conf. Research in Computer Security (ESORICS09), pp. 355-370, 2009.
7. D. Boneh, B. Lynn, and H. Shacham, Short Signatures from the Weil Pairing, Proc.Seventh Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology(ASIACRYPT01), pp. 514-532, 2001.
8. C. Erway,A. Kupcu,C. Papamanthou, and R. Tamassia, Dynamic Provable Data Possession ,Proc. 16th ACM Conf. Computer and Comm. Security (CCS09), pp. 213-222, 2009.
9. 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 (SAC11), pp. 1550-1557, 2011.
10. 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.
11. 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.
12. B. Chen, R. Curtmola, G. Ateniese, and R. Burns, Remote Data Checking for Network Coding-Based Distributed Storage Systems,Proc. ACM Workshop Cloud Computing Security Workshop (CCSW10), pp. 31-42, 2010.
13. N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, LT Codes-Based Secure and Reliable Cloud Storage Service,Proc. IEEE INFO COM, 2012.
14. R.L. Rivest, A. Shamir, and Y. Tauman, How to Leak a Secret,Proc. Seventh Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT01), pp. 552-565, 2001.
15. R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems,Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.
16. B. Wang, B. Li, and H. Li, Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud,Proc. 10th Intl Conf. Applied Cryptography and Network Security (ACNS12),pp. 507-525, June 2012.
17. X. Liu, Y. Zhang, B. Wang, and J. Yan, Mona: Secure Multi Owner Data Sharing for Dynamic Groups in the Cloud,IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191,June 2013.
18. B. Wang, H. Li, and M. Li, Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics,Proc. IEEE Intl Conf. Comm. (ICC13), pp. 539-543, 2013.
19. B. Wang, S.S. Chow, M. Li, and H. Li, Storing Shared Data on the Cloud via Security-Mediator,Proc. IEEE 33rd Intl Conf. Distributed Computing Systems (ICDCS13),pp. 124-133, 2013.
20. 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.