ICT-6541 Applied Cryptography

Privacy Preserving Auditing for Shared Data in the Cloud

Mohammad Nazrul Islam
ID. 1018312016

10.04.2019

Privacy Preserving Auditing for Shared Data in the Cloud

Mohammad Nazrul Islam
ID. 1018312016

Abstract- Due to the non-sporadic advancement of technology now users can remotely store their data remotely and enjoy the on-demand high quality applications and services economically in clouds. But the integrity of cloud data is subject to skepticism to the users. Several mechanisms have been designed to allow both the data owners and the public verifiers to efficiently audit the integrity of the cloud data without retrieving the entire data from the cloud server also without being pernicious to the privacy of the data and data users. Also several mechanisms have been designed and introduced to support the dynamicity of the shared data group where users can be added or revoked by maintaining the integrity and privacy of the data and data users. To support large number of users in a group and to audit the most recent data on the shared storage, multiple techniques have also been introduced. In this paper, these several techniques of ensuring privacy preserving for shared data are discussed and a comparison among them is done in brief.

Keywords- Cloud, public auditing, shared data, privacy-preserving.

I. INTRODUCTION

The term "cloud" originated from the telecommunications sector, when virtual private network (VPN) services for data communications. Cloud computing refers to the computation, software, data access and storage services of a system that, not necessarily requires end-user having knowledge of the physical location or the configuration of the system. Cloud computing can be defined as a recent trend in IT that moves data and computing away from stationary devices i.e. desktop and portable PCs into large data centers. The main goal of cloud computing is to make a better use of distributed resources by combining them to achieve a higher throughput and to be able to solve large scale computation or problems.

Cloud storage is a model of computer data storage which refers to the storing of digital data into logical pools. The physical storage can comprises of multiple servers locating in same or different locations. The cloud storage provider also known as hosting company is responsible for the availability and accessibility the stored data.

Having both cloud computing and cloud storage facility available, users now at a lower marginal cost can access and share resources offered by different cloud service providers like- Dropbox, Google Drive, iCloud. And now in these platforms users can share data with other user or other groups easily and economically. So, data sharing became standard feature in most cloud storage offerings systems. However, the integrity of the stored or shared data is a skeptic subject to the users. Adverse effects of hardware and software failures of the service providing systems buttressed the skepticism of the users.

Though the traditional approach for checking data correctness, which is- retrieving the entire data and then checking the correctness of the hash values (MD5) or signatures (RSA) of the entire data could bring a successful check of entire data of the cloud, but it would be achieved by a higher cost of computation and communication time and space due to its large size. And if this is done on erroneous data then there will be a full waste of resources.

So, to increase the efficiency of the data correctness check, several mechanisms have been introduced. Where not only the data owner but also a public verifier, be it a data user or a third-party auditor can perform the integrity

check of the data from the cloud. These mechanisms are referred as public auditing where the entire data is partitioned into several small blocks, each block is independently signed by a user of that data and integrity checking of the entire data is done on a random combination of these blocks rather than on the whole data. Now, if a third-party auditor is used then it is termed as TPA. However, in public auditing using this TPA introduced a new security issue regarding protecting the identity privacy of the data users to these third-party auditors. To solve the emerged privacy issue, several mechanisms have also been introduced where the identity of the signer or the most valued block is kept private. There has been another issue of the data users, the complexity increases if a user is added or expelled from the group of shared data, which is termed as dynamic grouping. Then, several techniques have been introduced, where the system protects the shared data from the expelled users or to allow the added users to share the resources. And for large number of users in a group, clustering techniques has been introduced to efficiently do the auditing. And for doing this audition on recent data i.e. fresh data a new technique is also introduced to check for data freshness and then do the audition on the fresh data.

In this paper, these several techniques of ensuring privacy preserving for shared data are discussed and a comparison among them is done in brief.

II. PROBLEM STATEMENT

A. The System Model

The system comprises of three entities: the users, the cloud server and the third party auditor (TPA). Those who shares data as a group are considered as users. Typically, in a group there are users of two categories. One is the original user and other is the group of users. The original user is the one who originally owns the data in the group. He selects other users and forms a group with them to share his owned data. In case of a dynamic group, the original user also acts as a group manager who holds the power of adding any new user and revoking any former user. But when shared data are created, not only the original user but also the other users of the group can view and edit the data i.e. the group members can access and modify the shared data.

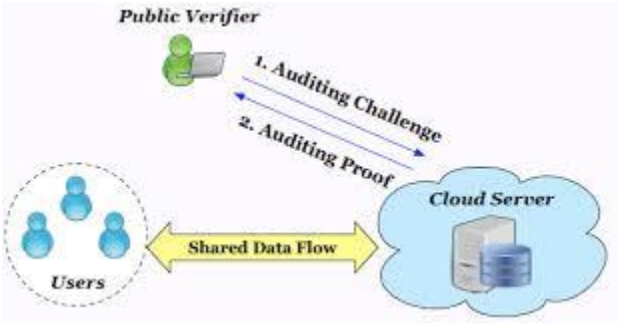


Figure 1. The public auditing system.

The cloud is the place where the original user created the shared data. And the cloud server refers to the place where the shared data and the verification information of those shared data are stored. The cloud server provides services to its users by the name- cloud service. The cloud service includes- software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). Being cheap and available as per demand, the cloud server has been very popular for the last few years.

Though people can store their data with less cost in the cloud, but the security of the stored data in the cloud has always been a bottleneck to the users. Due to the records of hardware and software failures, attacks on the data and the well-known economic behavior of the cloud server, which inhibits the server to notify the user about any unwanted hazard for the fear of losing market, the security issue has always been skeptical to the users. So, the users do not trust the cloud fully to maintain the integrity of the stored shared data. Thus, third party auditor (TPA) came in the scenario. TPA is a public verifier who does the work of the user by auditing the integrity of the shared data on the cloud. When a user wishes to check the integrity of the stored shared data, he

sends auditing request to the TPA. Then TPA requests auditing message from the cloud server where data is stored. And upon receiving the auditing message from the server, TPA checks the correctness of the auditing message and then sends the check result to the requesting user.

B. The Threat Model

The threat model comprises of possible threats that can happen in this system.

1) Integrity Threat: Integrity threat refers to the threats that have adverse effect on the integrity of the data. Till now, there can be of two type's integrity threats. One is due to an adversary, who tries to corrupt the shared data. This results in inhibiting the user to access correct data. And the other one is due to the cloud server. The cloud server may inadvertently corrupt or remove data from the stored shared data. But due to its economic feature, that is fear of losing trust or market of the user, the cloud may even be reluctant to share this information with the user.

2) Privacy Threat: Privacy thereat refers to the threats having adverse effect on the privacy of the data. A TPA is responsible only for checking the integrity of the data without downloading the data itself. But if somehow he gets hold of the data, and knows which user has signed which part or which block is signed the most by the users, then the TPA can get the idea or hint of the valuable users or valuable data among the stored data. This affects the system. Cause, this type of information was supposed to be secret, especially to a third party out of the group.

C. *Design Objectives*

To allow the TPA to function efficiently, the design of the system should be done to achieve following properties:

1. Public Auditing: The TPA is able to verify the integrity of the stored shared data of a group of users without accessing the entire data.

2. Correctness: The TPA is able to correctly detect the integrity of the data. That is to be able to detect any corrupt data in the stored shared data.

3. Non-forgerable: No one other than a user of a shared data group can generate a valid verification information i.e. signature on shared data.

4. Identity privacy: The TPA cannot distinguish between the signers of the block of the shared data.

III. EXISTING METHODS FOR PRIVACY PRESERVING AUDITING FOR SHARED DATA IN THE CLOUD

At first, B. Wang et. al.[1] introduced Oruta as a privacy preserving public auditing mechanism for shared data in the cloud. It is a public auditing mechanism with identity privacy which does not reveal the identity of the users. Oruta uses ring structure, specifically HARS (Homomorphic Autheticable Ring signature) scheme. Homomorphic authenticators are used to store the blocks of the data which has unique properties like, block-less verification and non-malleability.

Oruta, this technique works on four algorithms:

1. *KeyGen:* Users generate their own public/private key pairs. They randomly choose the private key. And based on that private, generate the public key. Also the users generate an aggregate key.
2. *SignGen:* This algorithm refers to the generation of signature on the data. Generally at first the original user computes the ring signatures on blocks in shared data. But later if there is any modification of data , then the modifier user computes ring signatures on blocks in shared data by using his own private key and public keys of all the group members.
3. *ProofGen:* This algorithm is used by both the TPA and the cloud server together interactively to generate a proof of the possession of the shared data.
4. *ProofVerify:* This algorithm is used by the TPA. The TPA uses this algorithm to audit the integrity of shared data by verifying the proof obtained from the cloud server.

Whenever a user of the group modifies any block of the shared data, a ring signature is computed by using his private key and all members public key by using SigGen algorithm. These signatures are verified by using the ProofVerify algorithm by the cloud server. So, this mechanism has achieved unforgeability, i.e. except the group user, none of the user can generate the signature on the block. Hence, this mechanism provides authentication security to the shared data.

However, Oruta runs on static groups only. So, Oruta is not capable of functioning efficiently on a dynamic group, where a user can be added or revoked from the group. Also, Oruta fails in ensuring the authentication on fresh data.

Another technique of B.Wang et. at. [2] is Knox, which also introduced privacy preserving auditing technique for shared data in the cloud for large groups. This technique aimed to robust the precious technique by solving the space complexity due to storing the resigned data. This technique is based on homomorphic MAC. This reduces the space required to store the verification data along with group signature using pseudo-random function. Knox uses Homomorphic authenticable Group Signature scheme, which achieves block-less verifiability and non-malleability.

Knox works on six algorithms:

1) *KeyGen:* This algorithm is used by the original user to compute a public key and a group manager private key.
2) *Join:* This algorithm is used by the original user to add another user in the group and issue private keys to the users of the group.
3) *Sign:* This algorithm is used by the members of the group to compute the signature of the data using his private key and the group key.
4) *ProofGen:* This algorithm is used by the public verifier and the cloud server together to interactively generate a proof of the possession of the shared data.
5) *ProofVerify:* This algorithm is used by the TPA to check the integrity of shared data by verifying the proof obtained from the server.
6) *Open:* This algorithm is used by the original user to reveal the identity of the signer of the shared data.

Knox can be performed on a large group of users. And it also privileges to have a group manager who can add or revoke any user on his misbehavior using the group manager's private key. That is unlike Oruta, in Knox, the group manager can add a user without re-computing any signature. Moreover, unlike Oruta, Knox can reveal the identity of the signer of the data.

However, though Knox support dynamic group with a greater number of users, but it doesn't support multiple TPA. So, it takes much time in auditing. And, Knox doesn't ensure authentication on fresh data.

A detailed comparative analysis of Oruta and Knox is given below-

TABLE 1
COMPARATIVE ANALYSIS OF ORUTA AND KNOX

	Oruta	Knox
Data Storage Usage (GB)	2	2
Signature Storage Usage (GB)	2	0.33
Communication Cost (KB)	18	106.4
Auditing Time (seconds)	11.49	3.44

B. Wang et. at.[3] proposed another technique of privacy preserving auditing for shared data. But this technique differs from the previous ones by ensuring efficient management for dynamic groups. In previous, for a

dynamic group they had to generate signatures with different private keys, but here they did it with a single group private key. And all other algorithms and processes are just like the previous ones.

Though this technique claimed to solve the signing hurdle of the previous ones, but still it possesses the problem of sharing this group private key with the users. Cause, after a user revocation, the user manager has to distribute secretly the new group private key with the existing users to ensure that the revoked user can't modify the shared data any longer and also doesn't get hold of the new private key.

A comparative analysis of Oruta, Knox and Dynamic grouping is given below, based on joining and revoking of user-

TABLE 2
COMPARATIVE ANALYSIS OF ORUTA, KNOX AND DYNAMIC GROUPING

	Oruta	Knox	Dynamic grouping
User joining	331.54	0.11	0.13
User revocation	330.27	43.93	2.91

Another technique for privacy preserving auditing for shared data by P. Maheswari et. al. [4] is AFS-Authenticated File System. It is almost same as Oruta of B. Wang et. al. [1] except data freshness. It works on authenticated file system. While performing the file operations it verifies the data. They guarantee data freshness with two layers: Lower layer stores a MAC along with a version number for each block that enables random access. And version number is incremented by each update. The upper layer consists of a Markle tree. Its leaves store the block versions while the internal nodes store the hashes of the children. Thus, the freshness of the file data block can be verified by the Mac block and the freshness of the block version.

Like Oruta, this technique works on four algorithms:

1. *KeyGen:* Users generate their own public/private key pairs. They randomly choose the private key. And based on that private, generate the public key. Also the users generate an aggregate key.
2. *SignGen:* This algorithm refers to the generation of signature on the data. Generally at first the original user computes the ring signatures on blocks in shared data. But later if there is any modification of data , then the modifier user computes ring signatures on blocks in shared data by using his own private key and public keys of all the group members.
3. *ProofGen:* This algorithm is used by both the TPA and the cloud server together interactively to generate a proof of the possession of the shared data.
4. *ProofVerify:* This algorithm is used by the TPA. The TPA uses this algorithm to audit the integrity of shared data by verifying the proof obtained from the cloud server.

P. Raghavan et. al.[5] proposed a privacy preserving public auditing technique that claims to handle user revocation efficiently and they did it by forming clusters of users. In their technique instead of having a large group of users they formed clusters of the users and each cluster acted as a group. And each cluster has a separate TPA. So, the time required for auditing has decreased for a great extent.

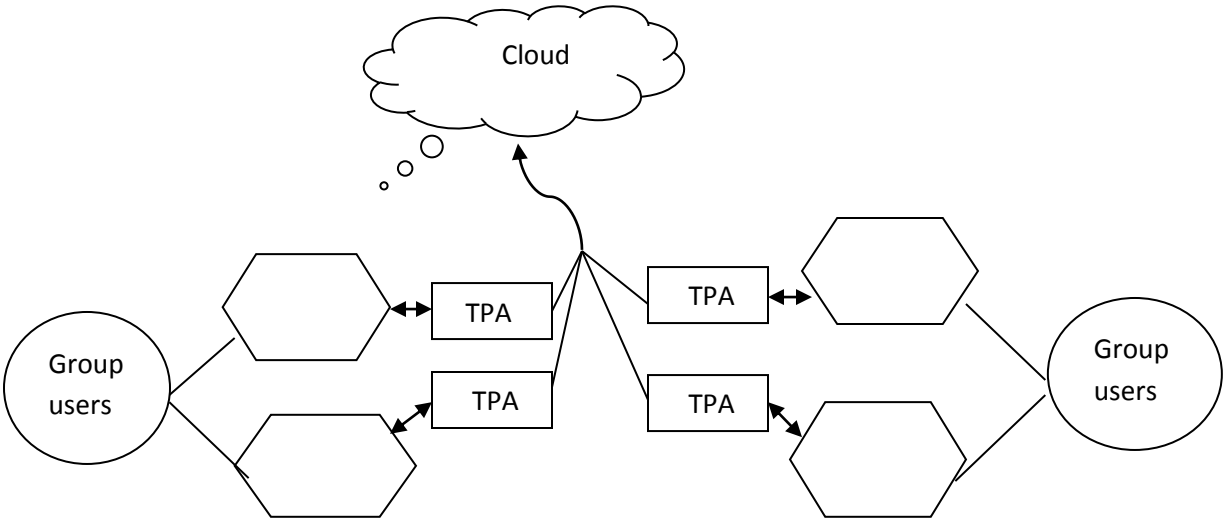
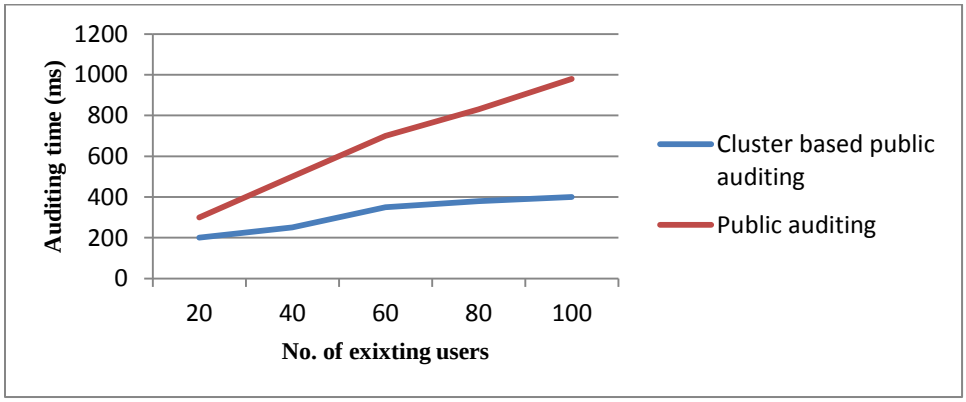
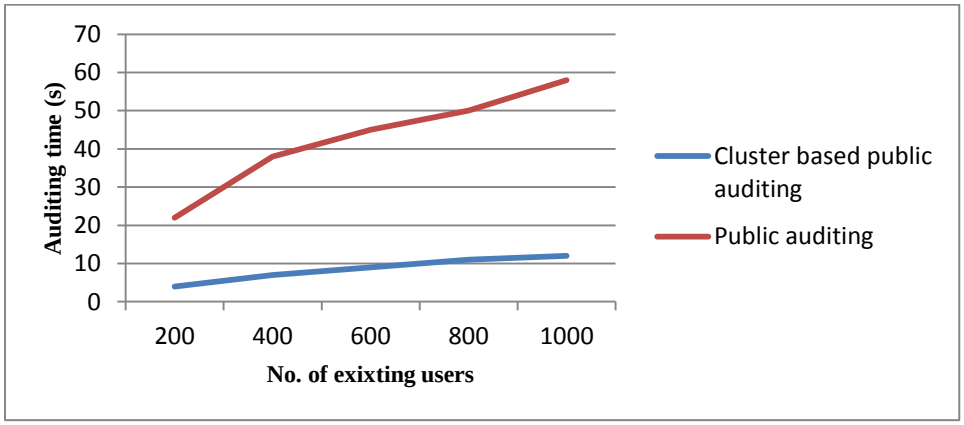


Figure 2. Cluster based public auditing system.



Graph 1. Verificatioin time between cluster basesd and public auditing and public auditing (<100 users)



Graph 2. Verificatioin time between cluster basesd and public auditing and public auditing (upto 1000 users)

IV. COMPARATIVE ANALYSIS

So, different techniques are introduced for privacy preserving public auditing. Among the above mentioned mechanisms, Oruta and AFS work on ring signature which ensures identity privacy. But both of them are unable to trace the identity of the user on the misbehavior i.e. can't cope up with dynamic group facility. Both Knox and system in [2] are based on group signature, which is able to trace the identity of misbehaved user and can revoke by respectively by using group manager's private key and a group private key. And this can be carried out by the group manager only. But there remains the question of secured distribution of the group key. All of the above techniques use the homomorphic authenticators with different schemes. And only AFS technique ensures auditing on fresh data, which is absent on all other techniques. And all other techniques except P. Raghavan et.al. [5] technique uses a single TPA no matter how many users are in the group. Only technique proposed by P. Raghavan et. al. [5] supports multiple TPA based on the number of clusters formed by users.

A comparative analysis among the discussed techniques of privacy preserving auditing for shared data is given below-

TABLE 3
COMPARATIVE ANALYSIS AMONG DISCUSSED TECHNIQUES OF PRIVACY PRESERVING AUDITING FOR SHARED DATA

	Oruta	Knox	Dynamic grouping	AFS	Cluster based
Public auditing	✓	×	✓	✓	✓
Data privacy	✓	✓	✓	✓	✓
Identity privacy	✓	✓	✓	✓	✓
Dynamic grouping	×	✓	✓	×	×
Data freshness	×	×	×	✓	×
Multiple TPA	×	×	×	×	✓
Parallel auditing	×	×	×	×	✓

V. CONCLUSIONS

Multiple privacy preserving techniques for public auditing on shared data and a comparative study of them is done. It cannot be said that this particular technique is best. Every technique has its unique features that differentiate it from the rest of the techniques. But it can be said that none of the techniques resolves all the hurdles for privacy preserving auditing for shared data. They need to do tread-off. From that, we can come to a conclusion that all the techniques yet not achieved perfection. They can be evolved or expanded further in terms of the lacking behavior to attain perfection. It is shown that different techniques use different schemes to authenticate and to verify the correctness of the blocks containing shared data. Though most of the techniques are protected from the attackers, with the advancement of technology, the form of attacks are also evolving. So, this can be a field of future venture and a subject of further exploration.

REFERENCES

[1] B. Wang, B. Li, H. Li, "Oruta: Privacy-Protecting Public Auditing for Shared Data in the Cloud", IEEE 5th International Conference on Cloud Computing, 2012.

[2] B. Wang, B. Li, H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud", 10th International Conference, ACNS 2012, pp. 507-525, June 2012.

[3] B. Wang, H. Li, M. Li, "Privacy-Protecting Public Auditing for Shared Cloud Data Supporting Group Dynamics", IEEE ICC 2013.

[4] P. Maheswari, B. Sindhumathi, "AFS: Privacy-Protecting Auditing With Data Freshness in the Cloud", IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, pp. 56-63.

[5] P. Raghavan, S. Sakthivel, "Cluster Based Public Auditing for Shared Data with Efficient Group User Revocation in the Cloud", ISSN: 0976-3104, August 2016.