A Survey on Integrity Verification in Cloud Computing

K.Sujatha[1] K.Sundar[2]
[1]M.E. Student [2]Assistant Professor
[1,2]Department of Computer Science & Engineering
[1,2]Velammal Engineering College, Chennai

Abstract—Cloud computing has been popular as the IT architectures. Cloud service providers offers many services based on cloud computing. Cloud storage service is one of the cloud services which can provide a huge storage space to solve the bottleneck of the storage space of local end users. Here the Cloud server allows the users to store their data on cloud without worrying about correctness and integrity of data. Cloud data storage has many advantages over local data storage. Users can upload their data on cloud and can access those data anytime anywhere without any additional burden. The Users need not to worry about storage and maintenance of cloud data. However, cloud storage service may have data security because the user's data is stored at the untrusted servers instead of their own storage. In this paper, we will focus on data integrity which is very important cloud storage service. Public auditability is a model of outsourcing data integrity verification, which can achieve efficiency and security. Therefore, we survey the previous researches of data integrity based on public auditability to analyze security and efficiency. In this paper we propose a new idea for privacy preserving public auditing for data storage security in cloud computing using AES algorithm and Secure Hash Algorithm. It will also support data dynamics where the users can insert, delete and update their data.

Key words: AES, Secure Hash Algorithm, TPA

I. INTRODUCTION

Cloud computing is a computing technology which provides different types of services through Internet. It shares the software and hardware resources, and provide resources to user's computer or mobile device. Cloud service provider's offers three important services. They are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud computing has four models. They are

- Public cloud: Public clouds are prepared to be available to the general public. Generally, public cloud providers like Microsoft and offer is access over the internet. In this model, customers doesn't have any visibility or control over where the infrastructure is.
- Private cloud: It is a cloud infrastructure which is designed to a particular organization. It doesn't share with other organizations, whether managed internally or externally by third-party, and it can be hosted externally or internally.

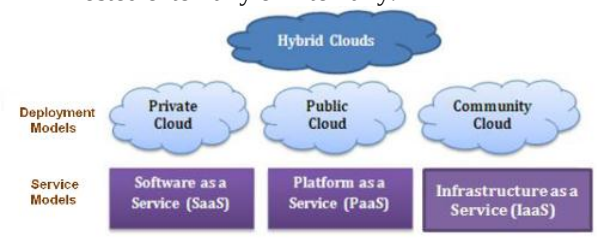

Fig. 1: Cloud Deployment Models

- Community Cloud: here the cloud is a multi-tenant cloud service model which is shared among several organization. It is controlled and managed commonly by all the participating organizations.
- Hybrid cloud: It is a composition of two or more clouds (private, public or community cloud).

Cloud storage service is the most common and popular service among many cloud services (e.g. Google Drive, Dropbox, etc...) for general users. They have a bottleneck in local storage space because there are more and more users save their data in cloud storage, so cloud storage service has high capacity which solves user's difficulty in storing and retrieving data. Besides, cloud storage service provides high capacity space in order to achieve ubiquitous service, it also provides access to cloud services from web services or applications which utilize the application programming interface by mobile devices (e.g. laptop, table computer and smart phones). Even though the cloud storage service has many advantages, it brings a lot of challenging issues which include efficiency and security. One of the biggest challenge in cloud computing is verifying the integrity of the data stored at the untrusted servers. Because users cannot know how the cloud storage service handles their data. Cloud storage services are provided by commercial enterprises, so that it cannot be fully trusted by users. Therefore, the cloud service provider may hide data loss and data errors in the service in order to maintain their reputation. It is very serious when a user stores data in untrusted cloud storage, Eg. a large size of the outsourced data and the client's limited resource capability, and the client how to find an efficient way to achieve integrity verifications without the local copy of data files.

A. *Data Integrity*

Integrity, is a one which is used to ensure that the data stored is as it is, there is no modification have been done in it. The data can modified and accessed only by the authorized users. Data Integrity is a main challenging issue in cloud computing. In cloud computing users store their data. Then the users relies on cloud servers for storage and maintenance. But such hope may also fail sometimes. Because the servers may also misuse or delete the rarely accessed data of user's in order to maintain their reputation. So integrity verification is main concern in cloud computing. For this purpose we have two integrity verification methods. They are private auditing and public auditing. In private auditing the authorized one only can audit the users data whereas in public auditing anyone can perform the auditing task. In my paper I am going to introduce public auditing with privacy preservation. So anyone can audit data correctness of my data without accessing the actual file content.

II. RELATED WORKS

There are many approaches has been proposed for data integrity verification in cloud computing [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. In 2007, the provable data possession (PDP) model is proposed by Ateniese et al. [1]. It uses public auditability and ensures the possession of user's data on untrusted storage. Here they were used RSA-based homomorphic verifiable tags in order to audit the outsourced data. Their scheme provides both block less verification and public verifiability at the same time. Even though, Ateniese et al.'s scheme cannot support verification of dynamic data because their scheme only considers the static data situation where the client stores outsourced data and will not modify it. In order to overcome this, Ateniese et al. [2] proposed a scalable PDP scheme in 2008 to improve dynamic data verification. However, their scheme cannot support fully dynamic data. Because their scheme cannot support block insertions and only allows simple block operation which performs partially dynamic data like block level modification and block level deletion. A challenge-response protocol is proposed by Wang et al. [9] which is used to determine data correctness and locate possible errors. But, their scheme supports only partially dynamic data operation. A dynamic provable data possession has been introduced by Erway et al. [3]. It extends the PDP model to support fully dynamic data operations. They used another authenticated data structure which is a rank-based authenticated skip lists to prove and update the remotely stored data. Even though, their scheme cannot support public verification why because they only considers the fully dynamic data as their main goal. Juels and Kaliski [4] introduced the proof of retrievability (POR) model, in which the spot-checking and error-correcting codes can make sure the possession of data files and retrievability of it on remote archive service systems. However, their scheme only suitable for static data storage because the number of queries can performed by a client is fixed a priori and embedding special blocks which prevent the development of dynamic data updates. Shacham and Waters [7] proposed an improved POR scheme. It uses BLS signature in order to replace RSA-based signature and also to reduce the proof size. They were used secure random oracle model with public verifiable homomorphic linear authenticators which are built from BLS signature. They proved that it is secure in a polynomial extraction algorithm to reveal messages. However, they were only considered static data operation. In order to satisfy public verification and dynamic data Wang et al. [10] proposed a new scheme. In this scheme they improves an index of data block which can support fully dynamic data. They also extended their scheme to support batch auditing which is used to improve efficiency. Wang et al. [8] pointed out that Wang et al.'s scheme has data privacy issues because the TPA can get the client's data information. So, they use a random mask technology in order to avoid TPA learning knowledge on every verification process. Li et al. [5] considered that the client's resource-constrained device which is simple and lightweight. Therefore, they propose a new scheme which delegate TPA to execute high computing process and solve the client's overhead

A. Basic Requirements And Evaluation Metrics

According to [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] studies, where they provide the basic requirements of security and performance.

1) Security Evaluation:

- Block less Verification: The auditor can verify data blocks, and need not to retrieve all audited data blocks in the cloud storage service.
- Stateless Verification: the auditor needs not to maintain and update data situation because data situation is maintained by the client and cloud storage service together.
- Batch Auditing: The auditor can verify the data of different clients at the same time because the auditor can be delegated by a lot of clients.
- Dynamic Data. The data owner can insert, modify and delete data blocks in the cloud storage service because their data can be continuously updated at any time.
- Privacy Preserving: The auditor cannot get knowledge which is the delegated data from the response of the cloud storage service.

2) Performance Evaluation:

- Computing Cost: In order to achieve an efficient public auditing, we will analyze the client, TPA and cloud storage service cost on the computing resources.
- Storage Cost: Because the client will upload data to the cloud storage service without the local copy of data files, we will analyze the client, TPA and cloud storage service cost on the storage spaces.
- Reduce storage cost.

III. LITERATURE SURVEY

A. Provable Data Possession (PDP)

In paper [1] author proposed a Provable Data possession (PDP) scheme in order to assuring the data integrity over remote servers. In Provable Data Possession scheme a client can store their data at an unfaithful server and can verify whether the server possesses the original content of data file or not. This is the first technique to consider the public auditability in their defined "provable data possession" model in order to ensuring possession of user files on untrusted cloud storages. They were used RSA-based homomorphic verifiable tags to audit user's outsourced data in cloud. Their scheme provides both block less verification and public verifiability at the same time. Even though, Ateniese et al.'s scheme cannot support verification of dynamic data because their scheme only considers the static data condition in which the client stores outsourced data and cannot modify it.

1) Advantages:

- The server does not have to access the file blocks,
- Supporting Block less verification.
- Allows public verifiability.

2) Limitations:

- It has lack of privacy preservation.
- It doesn't support dynamic operations.
- It has unbound no. of queries.

B. Scalable PDP

Author in [2] proposed Scalable PDP which is an improved version of the original PDP. The main difference between these two is Scalable PDP uses the symmetric key encryption whereas the original PDP uses public key in order to reduce computation overhead. Scalable PDP have the dynamic operation on remote data. This scheme has all the pre-computed challenges and answers are and limited number of updates. It does not require a bulk encryption. Scalable PDP relies on the symmetric-Key which is more efficient compared to the public-Key encryption. But it does not offer public verifiability.

1) Limitations:

- A client can perform only limited number of updates and challenges.
- This scheme doesn't perform block insertions; it supports only modification and deletions of data blocks.
- This scheme is problematic for larger files. Because recreation of all the challenges are difficult.

C. Dynamic PDP

Dynamic PDP is a collection of seven polynomial-time algorithms it supports fully dynamic operations such as insert, update and delete. Here in this scheme [3] the author used an authenticated directories base on rank. It contains skip list for inserting and deleting operations. DPDP has some amount of computational complexity, but it is still efficient. For example, if we want to verify the proof for 600MB file, the DPDP only produces 208KB proof of data and has 15ms computational overhead. This technique provides fully dynamic operation support like deletion, insertion and modification etc. while it is support fully dynamic operation there is relatively higher communication, computational and storage overhead. Because all the challenges and proofs are dynamically generated.

1) Limitations:

- It has more computational complexity.
- It is not suitable for thin client.

D. Proof of Retrievability for large files

In [4] the authors uses a technique "Proof of Retrievability" for larger files by using "sentinels". In this method, only a single key has been used irrespective of the size of the original file. The small portion of the file F is in fact independent of the original length of File. Here they used a special sentinels blocks. They are hidden among other blocks in the data file and they are embeds randomly among the data blocks. In order to check the integrity of the data file F, the user need to send challenges to the CSP during the time of verification phase by providing the positions of the collection of sentinels and have to asks the CSP to return the related sentinel values. If the Cloud Server has modified or deleted the file, then there will be a possible that the position of sentinels may also be changed. So it is unable to respond correctly to the CSP. The encryption is performed on whole modified file in the CSP.

1) Limitations:

- This technique has the computational overhead for larger files while encryption is performed on whole file.
- This method provides storage overhead to the server.
- This scheme works only with static data.

E. Public Auditing of Privacy Preserving Data

Wang et al. [8] proposed a privacy protection scheme which is considered user's data privacy in the public auditability. Data privacy implies personally identifiable information or sensitive information whether they can be shared with third parties. As far as users are concerned what they depend on TPA just for the outsourced storage security of their data. However, most studies do not consider the protection of clients' private information in the auditing phase. This is a serious problem because an auditor may leak information without the client's authorization. Besides, there are legal regulations, such as the Health Insurance Portability and Accountability Act, it guarantees patient confidentiality for all healthcare-related data and demands the outsourced data not to be leaked to external parties. Because public auditing model allows third-party auditors to assist clients to verify their data integrity, TPA obtains partly data blocks and learns by each sample to collect information in the auditing phase. So Wang et al. proposed to integrate the homomorphic linear authenticator with random masking technique, and it achieves privacy-preserving public auditing. Because the random masking technique affects TPA learning knowledge, it can avoid TPA getting user's data.

F. Public Auditing of Resource constrained Devices

It says how to perform public auditing in Resource constrained devices. In [5] propose a public auditability scheme in resource-constrained devices. Resource constrained device is a simple and lightweight composition. Thus, these devices have low computation and storage capacity. However, these devices can achieve high mobility which allows users to carry and easily to use. Because the client may require repeatedly modified data in cloud storage service, this operation needs to compute in every update. Therefore, in the public audit model, the client needs a high burden of computing resources to operate dynamic data which is required to perform exponentiation and multiplication operation. In order to reduce the client's computation Li et al. propose a scheme which delegates trusted TPA to generate key, signature and delete file tag function. The clients can effectively reduce the computing resources because they only upload data to TPA. Therefore, the client will not have to compute signature on data update every time. Li et al.'s scheme is best on the client's point of view because the client delegates the whole operation process to TPA. Their scheme needs TPA to assist the client's data file, because it does not satisfy stateless verification. The client needs to store some information on the TPA because their scheme make the client delegate TPA to perform signature.

G. Authorized Public auditing of Fine Grained Update

These schemes can support public auditing and dynamic data update. However, these schemes [9, 8, 5] support to insert, delete and modify operation in a fixed-size block which is later termed as coarse-grained updates. For instance, when a data block is partially modified, the block will be completely modified in coarse-grained updates. Therefore, Liu et al. [6] propose a scheme which can support variable-size blocks in dynamic data update. they

propose a variable-size block scheme which is later termed as fine-grained updates in the public auditing. Their scheme can reduce an additional operation in partially modified block update. They also consider an authentication process to improve between the client and TPA. In the dynamic data update phase, Liu et al.'s scheme is better because their scheme can support partially modified data update which can reduce computing. However, Liu et al.'s scheme requires costly computing, but their scheme is the only way to achieve between TPA and CSS authentications.

IV. ANALYSIS

In the section, we will analyze these schemes [5, 6, 8, 10] which contain functional requirement, security and performance.

A. Functional Requirement

In order to raise efficiency in verification, every scheme can support blockless verification. Li et al.'s scheme [5] needs TPA to assist the client's data file, their scheme does not satisfy stateless verification. Although Li et al. [5] and Liu et al. [6] did not explain whether their scheme support batch audit, we analyze whether their scheme can be extended to achieve it. In the dynamic data, because these scheme [5, 8, 10] do not consider partially modified data update, Liu et al. [6] only considered to update variable-size blocks. Wang et al. [8] only considered privacy presenting using random mask technology because other schemes assume that TPA can be fully trusted.

B. Performance Evaluation

We will analyze three phases: setup phase, auditing phase and dynamic data update phase.we analyze the computation cost in setup, auditing, and dynamic data phases, respectively. In the setup phase, Wang et al.'s scheme [10] is better than these schemes [5, 6, 8] because their scheme does not compute the number of sectors of a block. However, Li et al's scheme [5] is best on the client's point of view because the client delegates the whole operation process to TPA. In the auditing phase, Wang et al.'s scheme [8] is better because the auditor reduces computation which cannot construct the root in the auditing phase. However, Liu et al.'s scheme [6] requires costly computing, but their scheme is the only way to achieve between TPA and CSS authentications. In the dynamic data update phase, Liu et al.'s scheme [6] is better because their scheme can support partially modified data update which can reduce computing. We also analyze storage cost in public auditing. Liu et al.'s scheme [7] requires a large storage space because their scheme can support partially modified data update and authentication. Li et al.'s scheme [5] needs to store some information on the TPA because their scheme make the client delegate TPA to perform signature.

V. PROPOSED SCHEME

A. Problem Statement

Security of the data stored in the cloud storage is a serious issue now a days. For the purpose of saving money and storage space the service provider might deliberately discard rarely accessed data files which belong to an ordinary client. So auditing of data is necessary to assure safety of client's data. In oreder to overcome this problem we introduce an secure storage system using an AES to verify users data and keep it safe.

B. Design

The system has three network entities. They are the client, CSS and TPA.

Client: an individual consumer or organization has a lot of data files and needs to store in the cloud. It depends on the cloud to manage data and computation, so it can reduce storage cost.

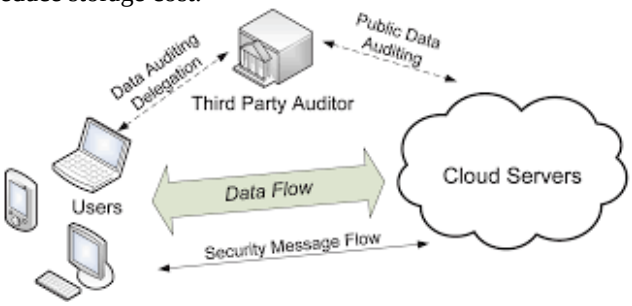

Fig. 2: System Architecture

- Cloud Storage Service (CSS): A cloud service provider has huge storage space and computation resource to provide the clients' data.
- Third Party Auditor (TPA): A trusted organization has expertise and capabilities that the clients do not have. It is responsible for assessing the clients data on cloud storage service.

C. General Idea

In proposed system, we are consider the cloud storage server is an untrusted storage server. While the server is consider to be untrusted we encrypt the data using AES algorithm before storing it to the cloud storage server. So the server cannot read the actual content in the file. If we uses AES-192 it will increase the power and time consumption. so that we are going to use AES-128 bit key here. we use the Secure Hash Algorithm for authentication and integrity verification of files.

D. Data Uploading

Here the user data is encrypted using AES before it is uploaded to the cloud server. At the time of uploading the cloud server generate the integrity proof of stored data. That is, the digital signature which is generated using Secure Hash Algorithm and send it to the client for later integrity verification.

E. Integrity Verification

At the time of downloading the user has to send request to TPA which includes digital signature generated during upload phase. Then TPA will send the challenge to the Cloud Storage Server for that corresponding file. Now the Cloud Storage server generates digital signature of the remote file and send it to the TPA. TPA verifies whether the stored proof and the proof which is generated now are same or not. If it is matches the integrity is maintaining otherwise not.

F. Data Downloading

After all the integrity verification has been done the user can download their data without any confusion about the storage

correctness of the data. The user data will be decrypted by using AES algorithm at the time of downloading.

VI. CONCLUSION

Because users' data is stored in the cloud storage service, it brings users' data security issues. In the public auditability model, users can delegate the third party auditor to verify their data is efficient. For future development, with big data generation, data verification will be more and more difficult. Therefore, it will be a major challenge how to efficiently verify data integrity in Cloud Storage. In this paper we proposes an efficient and secure system for auditing data stored at untrusted storage servers. This system supports the public auditing by using the TPA to verify the user's data and supports the privacy preserving by making the TPA to verify the users data without seeing the actual content. It will also support the dynamic operations on user's data. Therefore this proposed scheme will remove the burden of users and help them to keep their data safe.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, , D. Song and Z. Peterson "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, 2007.
[2] G. Ateniese, R., L. V. Mancini, D. Pietro and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, 2008.
[3] C. Erway, A. K, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession" in Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 213–222, Illinois, USA, 2009
[4] A. Juels, J. Burton and ,S. Kaliski, "Proofs of retrievability for large files," in Proceedings of he 14th ACM Conference on Computer and Communications Security, pp. 584–597, 2007.
[5] J. Li, X. Tan, D. Wong, X. Chen and F. Xhafa, "OPoR: Enabling proof of retrievability in cloud Computing with resource-constrained devices," accepted and to be published in IEEE Transactions on Cloud Computing, Oct. 2014.
[6] C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Ramamohanarao, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2234–2244, 2014.
[7] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security, pp 90–107, Melbourne, Australia, 2008.
[8] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Transactions on Computers, vol. 62, no. 2, pp. 362–375, 2013.
[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proceedings of the 17th International Workshop on Quality of Service, pp. 1–9, South Carolina, USA, 2009.
[10] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Paral lel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

