Dynamic Auditing Protocol for Data Storage on Cloud Environment

Supragya G\textsuperscript{1}, Sushmapriya MP\textsuperscript{2} & Rajesh N\textsuperscript{3}
\textsuperscript{1,2}B.E, Dept of ISE, NIE, Mysuru, Karnataka, India
\textsuperscript{3}Assistant Professor, Dept of ISE, NIE, Mysuru, Karnataka, India

Abstract - Auditing is an important service to verify the data in the cloud. In most of the current auditing protocols would inevitably become unable to work when a secret key for auditing is exposed. In this paper a new mechanism of cloud auditing is implemented. It is investigated on how to reduce the damage of the client’s key revelation in cloud storage auditing, and provide the first handy elucidation for this new problem setting. We formalize the definition and the security model of auditing protocol with key-exposure resilience and propose such a protocol. In the proposed design, the binary tree structure and the pre-order traversal technique is used to update the secret keys of the client. The security proof and performance analysis show that the projected protocol is protected and well organized.

1. Introduction

Cloud computing can help enterprises improve the creation and delivery of IT solutions by providing them with access to services in a cost-effective and flexible manner. Clouds can be classified into three categories, depending on their accessibility restrictions and deployment model. They are:

- Public Cloud,
- Private Cloud, and
- Hybrid Cloud.

A public Cloud is made available in a pay-as-you-go manner to the general public users irrespective of their origin or affiliation. A private Cloud’s usage is restricted to members, employees, and trusted partners of the organization. A hybrid Cloud enables the use of private and public Cloud in a seamless manner. Cloud computing applications span many domains, including business, government, health care, smart grids, intelligent transportation networks, life sciences, disaster management, automation, data analytics, and consumer and social networks. Various models for the creation, deployment, and delivery of these applications as Cloud services have emerged.

Cloud storage auditing is used to verify the integrity of the data stored in public cloud, which is one of the important security techniques in cloud storage. In recent years, auditing protocols for cloud storage have attracted much attention and have been researched intensively[1]. These protocols focus on numerous different characteristics of auditing, achieving high bandwidth and computation efficiency is one of the essential concerns. For that perseverance, the Homomorphic Linear Authenticator (HLA) technique that maintains block less verification is explored to diminish the overheads of computation and communication in auditing protocols, which allows the auditor to verify the integrity of the data in cloud without retrieving the whole data. Many cloud storage auditing protocols have been proposed based on this technique. In order to reduce the computational burden of the client, a third-party auditor (TPA) is introduced to help the client to periodically check the integrity of the data in cloud.

2. Literature survey

2.1 An efficient and secure dynamic auditing protocol

In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges[2], which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. In this paper, we first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, we extend our auditing protocol to support the data dynamic operations, which is
efficient and provably secure in the random oracle model. We further extend our auditing protocol to support batch auditing for both multiple owners and multiple clouds, without using any trusted organizer. The analysis and simulation results show that our proposed auditing protocols are secure and efficient, especially it reduce the computation cost of the auditor.

### 2.2 Privacy preserving public auditing for secure cloud storage

Using cloud storage, users can remotely store their data and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free.

To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

### 3. System design

We show an auditing system for secure cloud storage in Fig 1. The system involves two parties: the client (files owner) and the cloud. The client produces files and uploads these files along with corresponding authenticators to the cloud. The cloud stores these files for the client and provides download service if the client requires. For the simplicity of description, we assume that the client also plays the role of auditor in our system. The client can periodically audit whether his files in cloud are correct. In our model, the client will update his secret keys for cloud storage auditing in the end of each time period, but the public key is always unchanged. The cloud is allowed to get the client’s secret key for cloud storage auditing in one certain time period. It means the secret key exposure can happen in this system model.

An auditing protocol with key-exposure resilience[3] is composed by five algorithms as shown below:

1. **SysSetup**: It is the first algorithm that is first setup the input parameter k and the total time period T. Here the parameters that used in this algorithms is K and T. and finally it will generate an output as an public key PK. This was generated by the client.

2. **KeyUpdate**: It is a probabilistic algorithm. It will take the input as public key pk. For denoting the current period where the data to be position is find out by the parameter j. For the first period the current data that is denoted by the client secret key is SKj. And the next time period the current time is denoted as SKj+1. This algorithm is also run by the client side.

3. **AuthGen**: It is also termed as Authentication generated Algorithm. This algorithm is used to authenticate the file that should be used for process. This algorithm is also generated in client side.

4. **ProofGen**: This algorithm is used to verify the sign value of the system. This value is issued by the auditor. This algorithm is also generated by the client side.

5. **ProofVerify**: Proof verification is done by the client side was the proof should be used to find the required authority or not.

![Figure 1 System model of cloud storage auditing](image-url)
4. Conclusion

In the proposed paradigm, it is deliberated on how to deal with the client’s key exposure in cloud storage auditing. A new standard called auditing protocol with key-exposure resilience. The integrity of the data formerly stored in cloud can still be substantiated even if the client’s current secret key for cloud storage auditing is bare in these kinds of protocols. It is enacted in the definition and the security model of auditing protocol with key-exposure resilience, and has given the practical solution. The security proof and the asymptotic presentation assessment depicted that the protocol is secure and efficient. The efficient comparison between current protocol and earlier protocol based on BLS signature also has been provided.

5. References

