Identity-based integrity auditing in Cloud Storage

Henry Ding
Tandon School of Engineering
New York University
hd2340@nyu.edu

***Abstract*—With cloud storage services, users can remotely store their data to the cloud and share their data to others easily and conveniently. To ensure the data stored in the cloud are accurate, remote data integrity auditing solutions are proposed to protect the integrity of the data stored in the cloud. In common cloud storage systems such as the electronic health records system, some data are sensitive because they contain the patient's personal information. Such information cannot be exposed to others when they are shared through the cloud. Encrypting the data can protect the sensitive information, but it will make the data unusable by authorized personnel. Implementing a secure key exchange is plausible; however, it can be implemented poorly due to human error.**

I. Introduction

One of the biggest concerns of data sharing in the cloud is some data contain sensitive information. For example, the Electronic Health Records (EHRs) contain patients' personal information and their hospital's sensitive information (hospital's name, etc.). If EHRs are uploaded to the cloud for medical research purposes, then the patients' and hospitals' sensitive information will have high chances of being exposed to the public. Therefore, it is at upmost importance that remote data integrity auditing needs to be accomplished on the condition that the sensitive information is protected.

One method to protect sensitive information in shared data in the cloud is to encrypt the entire shared file and generate the signatures used to verify the integrity of the file before uploading them publicly to the cloud. This can protect sensitive information from the share data because only the owner can decrypt the file. However, it will make the shared file unusable by other researchers. It is possible to distribute the decryption keys to researchers in a shared channel, but this will be hard to be used securely because they are not well-trained to secure their decryption keys. Thus, creating risks of leaking the decryption keys to the public due to human error. Finding a secure and simple auditing system is crucial to protect sensitive information, but also to reduce human errors. However, many cloud storage security solutions overlooked the problem of exposing sensitive information after encryption in cloud data. Therefore, this paper will try to address the issue of protecting sensitive information from being leaked in the cloud. This article will brief the previously cloud storage security solutions in Section 2. Section 3 will discuss the motivation and inspiration behind this paper's proposed solution. Section 4 introduces the hypothesis and Section 5 shows the outcome of the performance analysis of the proposed solution. Finally, the conclusion and future work are discussed in Sections 6 and 7 respectively.

II. Related Research

There were many proposed remote data integrity auditing schemes that verify the integrity of the data stored in the cloud. Ateniese et al. [2] first proposed the concept of Provable Data Possession (PDP) to ensure the data possession on the untrusted cloud by using random sampling strategies are used to achieve blockless verification and reduce I/O costs.

In order to protect the data privacy in the cloud, Wang et al. [3] proposed a privacy-preserving remote data integrity auditing scheme by using a random masking technique. Worku et al. [4] achieves better efficiency than the proposed scheme in [3] by using a different random masking technique to further construct a remote data integrity auditing scheme supporting data privacy protection.

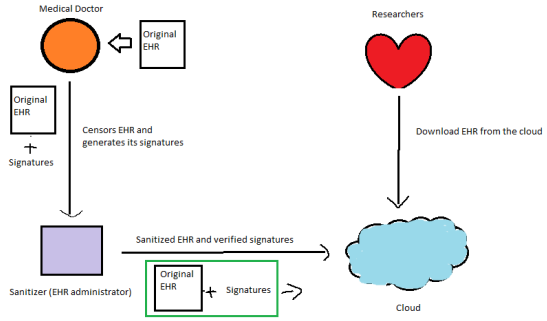
All mentioned solutions cannot support data sharing with sensitive information hiding. In this paper, we explore how to achieve data sharing while hiding sensitive information in identity-based integrity auditing for secure cloud storage.

III. Motivating Example

The motivation behind this paper was the experience of helping customers troubleshooting AWS services issues as a Cloud Support engineer at AWS. Although the AWS services I troubleshoot does not directly involve cloud storage services such as S3 buckets, the customers I have worked with use S3 buckets along with other AWS services such as Lambda and SQS (Simple Queue Service). If the customers did not have proper measures implemented for storing their S3 buckets, then they are at risk of massive financial loss. According to the Shared Responsibility model of AWS, the customer manages the security in the cloud, meaning the customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall. Although AWS is not directly responsible when the customer's cloud data got compromised because of poor configuration, I want to create a solution for the customers for them to implement within the AWS services such as S3 to prevent any future data breaches.

IV. Hypothesis

The paper proposes an identity-based shared data integrity auditing solution inspired by Shen et al.[1] that tackles the tasks of hiding sensitive information in secure cloud storage. The doctor first replaces the sensitive information on the file using wildcards "*", then generates signatures for the censored EHR and sends them to the "sanitizer" (the administrator of the EHR information systems in a hospital). The administrator then sanitizes the censored information into a uniform format such as PDF and sanitizes the hospital's sensitive information by removing unwelcomed characters such as line breaks, extra white spaces, tabs, ampersend, and tags. The admin also validates the corresponding signatures. After that, the admin stores the sanitized files into the EHR information system. If this EHR needs to be uploaded to the cloud, then the admin will simply upload the sanitized EHRs and their corresponding signatures to the cloud. To measure the proposed solution's effectiveness, I have decided to measure how much time it takes to perform certain tasks (computation overhead) in the proposed solution. In this research paper, I have decided to measure the computation overhead of the signature generation and signature verification. The smaller the computation overhead (the less time it takes to perform a task), the more effective the solution. This research paper hypothesizes the computation overhead will be linear for both performance metrics, meaning the performance of the proposed solution is dependent on the input size.



V. Empirical Evidence

All testing methods are performed using a C program with cryptographic libraries such as the Pairing-Based Cryptography (PBC) Library from Stanford [4] and the GNU Multiple Precision Arithmetic library [5]. I created a Word document that has a size of 20 mb for the C program to consume and split its data to about a total of 1000 data blocks. The data blocks are represented as blocks in the array. I then used the libraries mentioned previously to generate signatures and verify them for different numbers of blocks from 0 to 1000 increased by an interval of 100.

A. Computation Overhead in the process of signature generation and signature verification

In this experiment, we evaluate the performance of these processes: signature generation, and signature verification. To evaluate them, we set the number of data blocks to be 100, then we run the proposed solution to process these blocks.

To evaluate the computation overhead in the process of signature generation and signature verification, we generate the signatures for different number of blocks from 0 to 1000 increased by an internal of 100. In Fig. 1, the time consumed by signature generation and signature verification both increased linearly with the number of data blocks.

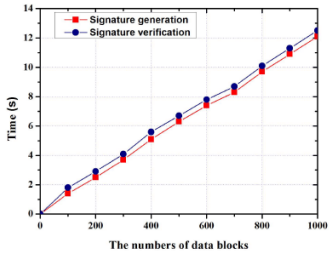


Fig. 1. The computation overhead in the process of signature generation and signature verification.

VI. Conclusion

This paper proposes an identity-based data integrity auditing scheme for secure cloud storage that supports data sharing without exposing sensitive information. In this proposed solution, the file stored in the cloud can be safely shared and used by others while the sensitive information is protected. Meanwhile, the performance experiments showed that the proposed solution is both secure and efficient.

VII. Future work

One of the things that the proposed solution can improve on is how to reduce the steps of the administrator sanitizing the data file after the censor process. If the administrator is busy or unavailable, then the doctors who urgently needs to upload the data to the cloud will face production issue because they don't have access to sanitize them. If the doctors can sanitize the data file with a click of the button, then they can quickly upload the data files to the cloud at the times of emergency.

References

[1] Wenting Shen, Q. Jing , Y. Jia , H. Rong and H. Jiankun , "Enabling Identity-Based Integrity Auditing and Data," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, vol. 14, no. 2, 2019.

[2] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[4] B. Lynn. (2015). The Pairing-Based Cryptographic Library. [Online]. Available: https://crypto.stanford.edu/pbc

[5] The GNU Multiple Precision Arithmetic Library (GMP). Accessed: Nov. 2022. [Online]. Available: http://gmplib.org