An Authorized Auditing Scheme for Identity-Based Integrity to Hide Confidential Information Data Sharing with Cloud Storage

D. Saritha Reddy, Y Vinay Kumar, V. Ravali

1Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering college, Gudur.
2Associate Professor, Dept. Of Computer Science and Engineering, Narayana Engineering College, AP, India.
3PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur

Abstract—With cloud storage services, users can remotely store their data to the cloud and realize the data sharing with others. Remote data integrity auditing is proposed to guarantee the integrity of the data stored in the cloud. In some common cloud storage systems such as the electronic health records system, the cloud file might contain some sensitive information. The sensitive information should not be exposed to others when the cloud file is shared. Encrypting the whole shared file can realize the sensitive information hiding, but will make this shared file unable to be used by others. How to realize data sharing with sensitive information hiding in remote data integrity auditing still has not been explored up to now. In order to address this problem, we propose a remote data integrity auditing scheme that realizes data sharing with sensitive information hiding in this paper. In this scheme, a sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file and transforms these data blocks’ signatures into valid ones for the sanitized file. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing. As a result, our scheme makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is hidden, while the remote data integrity auditing is still able to be efficiently executed. Meanwhile, the proposed scheme is based on identity-based cryptography, which simplifies the complicated certificate management. The security analysis and the performance evaluation show that the proposed scheme is secure and efficient. Keywords: Cloud Storage, Data Integrity Auditing, Data Sharing, Sensitive Information Hiding.

1. INTRODUCTION

With the explosive growth of data, it is a heavy burden for users to store the sheer amount of data locally. Therefore, more and more organizations and individuals would like to store their data in the cloud. However, the data stored in the cloud might be corrupted or lost due to the inevitable software bugs, hardware faults and human errors in the cloud [1]. In order to verify whether the data is stored correctly in the cloud, many remote data integrity auditing schemes have been proposed [2]–[8]. In remote data integrity auditing schemes, the data owner firstly needs to generate signatures for data blocks before uploading them to the cloud. These signatures are used to prove the cloud truly possesses these data blocks in the phase of integrity auditing. And then the data owner uploads these data blocks along with their corresponding signatures to the cloud. The data stored in the cloud is often shared across multiple users in many cloud storage applications, such as Google Drive, Dropbox and iCloud. Data sharing as one of the most common features in cloud storage, allows a number of users to share their data with others. However, these shared data stored in the cloud might contain some sensitive information. For instance, the Electronic Health Records (EHRs) [9] stored and shared in the cloud usually contain patients’ sensitive information (patient’s name, telephone number and ID number, etc.) and the hospital’s sensitive information (hospital’s name, etc.). If these EHRs are directly uploaded to the cloud to be shared for research purposes, the sensitive information of patient and hospital will be inevitably exposed to the cloud and the researchers. Besides, the integrity of the EHRs needs to be guaranteed due to the existence of human errors and software/hardware failures in the cloud. Therefore, it is important to accomplish remote data integrity auditing on the condition that the sensitive information of shared data is protected.

Contribution

The contribution of this paper can be summarized as follows:
(1) We investigate how to achieve data sharing with sensitive information hiding in remote data integrity auditing, and propose a new concept called identity-based shared data integrity auditing with sensitive information hiding for secure cloud storage. In such a scheme, the sensitive information can be protected and the other information can be published. It makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is protected, while the remote data integrity auditing is still able to be efficiently executed.

(2) We design a practical identity-based shared data integrity auditing scheme with sensitive information hiding for secure cloud storage. A sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file. In our detailed scheme, firstly, the user blinds the data blocks corresponding to the personal sensitive information of the original file and generates the corresponding signatures, and then sends them to a sanitizer. The sanitizer sanitizes these blinded data blocks into a uniform format and also sanitizes the data blocks corresponding to the organization’s sensitive information. It also transforms the corresponding signatures into valid ones for the sanitized file. This method not only realizes the remote data integrity auditing, but also supports the data sharing on the condition that sensitive information is protected in cloud storage. To the best of our knowledge, this is the first scheme with the above functions. Besides, our scheme is based on identity-based cryptography, which simplifies the complex certificate management.

(3) We give the security analysis of the proposed scheme, and also justify the performance by concrete implementations. The result shows that the proposed scheme achieves desirable security and efficiency.

2. RELATED WORK

In order to verify the integrity of the data stored in the cloud, many remote data integrity auditing schemes have been proposed. To reduce the computation burden on the user side, a Third-Party Auditor (TPA) is introduced to periodically verify the integrity of the cloud data on behalf of user. Ateniese et al. [2] firstly proposed a notion of Provable Data Possession (PDP) to ensure the data possession on the untrusted cloud. In their proposed scheme, homomorphic authenticators and random sampling strategies are used to achieve block less verification and reduce I/O costs. Juels and Kaliski [3] defined a model named as Proof of Retrievability (PoR) and proposed a practical scheme.

In this scheme, the data stored in the cloud can be retrieved and the integrity of these data can be ensured. Based on pseudorandom function and BLS signature, Shacham and Waters [4] proposed a private remote data integrity auditing scheme and a public remote data integrity auditing scheme. In order to protect the data privacy, Wang et al. [5] proposed a privacy-preserving remote data integrity auditing scheme with the employment of a random masking technique. Worku et al. [6] utilized a different random masking technique to further construct a remote data integrity auditing scheme supporting data privacy protection. This scheme achieves better efficiency compared with the scheme in [5]. To reduce the computation burden of signature generation on the user side, Guan et al. [7] designed a remote data integrity auditing scheme based on the indistinguishability obfuscation technique. Shen et al. [8] introduced a Third-Party Medium (TPM) to design a light-weight remote data integrity auditing scheme. In this scheme, the TPM helps user generate signatures on the condition that data privacy can be protected.

In order to support data dynamics, Ateniese et al. [10] firstly proposed a partially dynamic PDP scheme. Erway et al. [11] used a skip list to construct a fully data dynamic auditing scheme. Wang et al. [12] proposed another remote data integrity auditing scheme supporting full data dynamics by utilizing Merkle Hash Tree. To reduce the damage of users’ key exposure, Yu et al. [13] and [14], and Yu and Wang [15] proposed key-exposure resilient remote data integrity auditing schemes based on key update technique.

The data sharing is an important application in cloud storage scenarios. To protect the identity privacy of user, Wang et al. designed a privacy-preserving shared data integrity auditing scheme by modifying the ring signature for secure cloud storage. Yang et al. constructed an efficient shared data integrity auditing scheme, which not only supports the identity privacy but only achieves the identity traceability of users. Fu et al. designed a privacy-aware shared data integrity auditing scheme by exploiting a homomorphic verifiable group signature. In order to support efficient user revocation, Wang et al. proposed a shared data integrity auditing scheme with user revocation by using the proxy re-signature.
With the employment of the Shamir secret sharing technique, Luo et al. constructed a shared data integrity auditing scheme supporting user revocation. Other aspects, such as privacy-preserving authenticators and data deduplication in remote data integrity auditing have also been explored. However, all of existing remote data integrity auditing schemes cannot support data sharing with sensitive information hiding. In this paper, we explore how to achieve data sharing with sensitive information hiding in identity-based integrity auditing for secure cloud storage.

3. PROPOSED WORK

System Model
The system model involves five kinds of different entities: the cloud, the user, the sanitizer, the Private Key Generator (PKG) and the Third-Party Auditor (TPA), as shown in Fig. 1.

![Fig. 1. The System Model](image)

This system model consists and implements the following modules:

**Cloud**: The cloud provides enormous data storage space to the user. Through the cloud storage service, users can upload their data to the cloud and share their data with others.

**User**: The user is a member of an organization, which has a large number of files to be stored in the cloud.

**Sanitizer**: The sanitizer is in charge of sanitizing the data blocks corresponding to the sensitive information (personal sensitive information and the organization’s sensitive information) in the file, transforming these data blocks’ signatures into valid ones for the sanitized file, and uploading the sanitized file and its corresponding signatures to the cloud.

**PKG**: The PKG is trusted by other entities. It is responsible for generating system public parameters and the private key for the user according to his identity ID.

**TPA**: The TPA is a public verifier. It is in charge of verifying the integrity of the data stored in the cloud on behalf of users.

The proposed scheme has following algorithms:
Algorithm-1: **Private Key Generation**

1. **identity** $ID = (ID_1, ID_2, \ldots, ID_i)$

2. randomly picks $r_D \in Z_p^*$ and computes user $ID$’s private key $sk_D = (sk_D^{x_1}, sk_D^{x_2}) = (g_z^{x_1} \cdot (\mu \prod_{j=1}^{i} \mu_j^{ID_j}) y_z, g_z^{z_2})$

3. $sk_D$

4. verifies the correctness of the private key $sk_D$

Algorithm-2: **Signature Generation & Sanitization**

1. randomly chooses a value $r \in Z_p^*$ and calculates $\hat{r}$
2. randomly chooses a seed $k_1 \in Z_p^*$ and calculates the blinding factor $b_1 = \hat{r} \cdot (\mu \prod_{j=1}^{i} \mu_j^{ID_j}) y_z$
3. computes the blinded data blocks $\sigma_i = m_i + \tau_0 (0 \in K_i)$
4. calculates the signature set $\Phi = \{\sigma_i\}_{i=0}^{|I|}$
5. sets $\tau_0 = name \ || \ b_1 \ || \ \hat{r}$
6. calculates the file tag $\tau = \tau_0 \ || \ Sig_{\sigma_i}(\tau_0)$
7. calculates transformation value $\beta = \hat{r}$
8. $(F, \Phi, \tau, K_i)$ and $\beta$

9. checks the validity of the file tag $\tau$ and the file tag $\tau$
10. verifies the correctness of the signature set $\Phi$
11. verifies the correctness of $\beta$
12. sanitizes the blinded data blocks and the data blocks whose indexes are in set $K_i$
13. computes the signature set $\Phi' = \{\sigma_i\}_{\tau_{\text{new}}}$ of sanitized file
14. $(F', \Phi')$
Algorithm-3: Integrity Auditing

The proposed method implements a new efficient signature algorithm in the phase of signature generation. The designed signature scheme supports block less verifiability, which allows the verifier to check the integrity of data without downloading the entire data from the cloud.

4. EXPERIMENTAL RESULTS & DISCUSSION

In this subsection, we evaluate the performance of the proposed scheme by several experiments.

Fig. 2. Performance of different processes

Performance of Different Processes: To effectively evaluate the performance in different processes, we set the number of data blocks to be 100 and the number of sanitized data blocks to be 5 in our experiment. As shown in Fig. 2, private key generation and private key verification spend nearly the same time, which are nearly 0.31s. The time consumed by the signature generation is 1.476s. The time of signature verification and that of sensitive information sanitization respectively are 2.318s and 0.041s. So, we can conclude that in these processes, the signature verification spends the longest time and the sensitive information sanitization spends the shortest time.
To evaluate the performance of signature generation and signature verification, we generate the signatures for different number of blocks from 0 to 1000 increased by an interval of 100 in our experiment. As shown in Fig. 3, the time cost of the signature generation and the signature verification both linearly increases with the number of the data blocks. The time of signature generation ranges from 0.121s to 12.132s. The time of signature verification ranges from 0.128s to 12.513s.

**Performance of Auditing:** With the different number of challenged data blocks, we respectively show the computation overhead of the TPA and that of the cloud in integrity auditing phase in Fig. 4 and Fig. 5. In our experiment, the number of challenged data blocks varies from 0 to 1,000.

As shown in Fig. 4, we see the that the computation overheads of challenge generation and proof verification on the TPA side linearly increase with the number of challenged data blocks. The computation overhead of proof verification varies from 0.317s to 11.505s. Compared with the time of proof verification, the time of challenge generation increases slowly, just varying from 0.013s to 0461s.
From Fig. 5, we have the observation that the computation overhead of proof generation on the cloud side varies from 0.021s to 3.981s. So we can conclude that, with the more challenged data blocks, both the TPA and the cloud will spend the more computation overheads.

5. CONCLUSION

In this Scheme, we proposed an identity-based data integrity auditing scheme for secure cloud storage, which supports data sharing with sensitive information hiding. In our scheme, the file stored in the cloud can be shared and used by others on the condition that the sensitive information of the file is protected. Besides, the remote data integrity auditing is still able to be efficiently executed. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency.

REFERENCES


Author’s Profile:

**D. Saritha Reddy** has received her M.Tech degree in Computer Science from Acharya Nagarjuna University, Guntur in 2010 and pursuing Ph.D with area Computer Networks. At present she is working as Assistant Professor in Narayana Engineering College, Gudur, Andhra Pradesh, India.

**Y. Vinay Kumar** has received his M. Tech degree in Computer Science and Engineering from Andhra University, Vizag in the year 2006. He completed B. Tech in Computer Science and Engineering at Narayana Engineering College, Nellore affiliated from JNTUH, Hyderabad in the year 2003. At present he is pursuing Ph. D at K. L. University and working as an Associate Professor in Narayana Engineering College, Andhra Pradesh, India.

**V. Ravali** has Received her B.Sc Degree in Computer Science from Sri VSSC Govt. Degree College, Sullurpeta affiliated to VikramaSimhapuri University, Nellore in 2017 and pursuing PG Degree in Master of Computer Applications(M.C.A)from Narayana Engineering College ,Gudur affiliated to JNTU Anantapur, Andhra Pradesh, India.