PRIVATE AND PUBLIC IDENTITY-BASED DATA SHARING FOR SECURE CLOUD STORAGE SYSTEM

M.MARIA SHEEBA*

[a] *Department of Computer Science, Ponjesly College of Engineering, Tamil Nadu, India*

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. 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 sanitiser is used to sanitize the data blocks corresponding to the sensitive information of the file and transform 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. The security analysis and the performance evaluation show that the proposed scheme is secure and efficient.

Keywords: *cloud storage, data integrity, sensitive information, data storage.*

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

* For Correspondence *er.mariasheeba@gmail.com*

software bugs, hardware faults and human errors in the cloud. To verify whether the data is stored correctly in the cloud, many remote data integrity auditing schemes have been proposed. 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 some 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) 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 patients 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. A potential method of solving this problem is to encrypt the whole shared file before sending it to the cloud, and then generate the signatures used to verify the integrity of this encrypted file, finally uploading this encrypted file and its corresponding signatures to the cloud. This method can realize the sensitive information hiding since only the data owner can decrypt this file. However, it will make the whole shared file unable to be used by others. For example, encrypting the EHRs of infectious disease patients can protect the privacy of patients and hospitals, but these encrypted EHRs cannot be effectively utilized by researchers anymore. Distributing the decryption key to the researchers seems to be a possible solution to the above problem. However, it is infeasible to adopt this method in real scenarios due to the following reasons. Firstly, distributing decryption keys needs secure channels, which is hard to be satisfied in some instances. Furthermore, it seems very difficult for a user to know which researchers will use his/her EHRs soon when he/she uploads the EHRs to the cloud. As a result, it is impractical to hide sensitive information by encrypting the whole shared file. Thus, how to realize data sharing with sensitive information hiding in remote data integrity auditing is very

important and valuable. Unfortunately, this problem has remained unexplored in previous researches.

2 EXISTING TECHNIQUES

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 the user. Ateniese *et al.* firstly proposed a notion of Provable Data Possession (PDP) to ensure the data possession on the untrusted cloud[1]. In their proposed scheme, homomorphic authenticators and random sampling strategies are used to achieve blockless verification and reduce I/O costs. Juels and Kaliski defined a model named Proof of Retrievability (POR) and proposed a practical scheme[2]. 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 proposed a private remote data integrity auditing scheme and a public remote data integrity auditing scheme[3,4]. To protect data privacy, Wang *et al.* proposed a privacy-preserving remote data integrity auditing scheme with the employment of a random masking technique[5].

Worku *et al.* utilized a different random masking technique to further construct a remote data integrity auditing scheme supporting data privacy protection[6]. This scheme achieves better efficiency compared with the scheme in. To reduce the computation burden of signature generation on the user side, Guan *et al.* designed a remote data integrity auditing scheme based on the indistinguishability obfuscation technique[7]. Shen *et al.* introduced a Third Party Medium (TPM) to design a lightweight remote data integrity auditing scheme[8]. In this scheme, the TPM helps the user generate signatures on the condition that data privacy can be protected. To support data dynamics, Ateniese *et al.* firstly proposed a partially dynamic PDP scheme[9]. Erway *et al.* used a skip list to construct a fully data dynamic auditing scheme[10]. Wang *et al.* 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.* and, and Yu and Wang proposed key-exposure resilient remote data integrity auditing schemes based on key update techniques. Data sharing is an important application in cloud storage scenarios. To protect the identity privacy of the 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 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. 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. The aforementioned schemes all rely on Public Key Infrastructure (PKI), which incurs considerable overheads from complicated certificate management. To simplify certificate management, Wang proposed an identity-based remote data integrity auditing scheme in multi-cloud storage. This scheme used the user's identity information such as the user's name or e-mail address to replace the public key. Wang *et al* designed a novel identity-based pro-oriented remote data integrity auditing scheme by introducing a proxy to process data for users.

Yu *et al.* constructed a remote data integrity auditing scheme with perfect data privacy preservation in identity-based cryptosystems. Wang *et al.* proposed an identity-based data integrity auditing scheme satisfying unconditional anonymity and incentive. Zhang *et al.* proposed an identity-based remote data integrity auditing scheme for shared data supporting real efficient user revocation. Other aspects, such as privacy-preserving authenticators and data deduplication in remote data integrity auditing have also been explored. However, all of the 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 TECHNIQUES

To achieve data sharing with sensitive information hiding, we consider making use of the idea in the sanitizable signature to sanitize the sensitive information of the file by introducing an authorized sanitiser. Nonetheless, it is infeasible if this sanitizable signature is directly used in remote data integrity auditing. Firstly, this signature is constructed based on chameleon hashes. However, a lot of chameleon hashes exhibit the key exposure problem. To avoid this security problem, the signature used requires strongly unforgeable chameleon hashes, which will inevitably incur huge computation overhead. Secondly, the signature used does not support blockless

verifiability. It means that the verifier has to download the entire data from the cloud to verify the integrity of data, which will incur huge, communication overhead and excessive verification time in big data storage scenarios. Thirdly, the signature used is based on the PKI, which suffers from complicated certificate management. To address the above problems, we design a new efficient signature algorithm in the phase of signature generation. The designed signature scheme supports blockless verifiability, which allows the verifier to check the integrity of data without downloading the entire data from the cloud. In addition, it is based on identity-based cryptography, which simplifies the complicated certificate management. In our proposed scheme, the PKG generates the private key for the user according to his identity *ID*. The user can check the correctness of the received private key.

When there is a desire for the user to upload data to the cloud, to preserve the personal sensitive information of the original file from the sanitiser, this user needs to use a blinding factor to blind the data blocks corresponding to the personal sensitive information of the original file. When necessary, the user can recover the original file from the blinded one by using this blinding factor. And then this user employs the designed signature algorithm to generate signatures for the blinded file. These signatures will be used to verify the integrity of this blinded file.

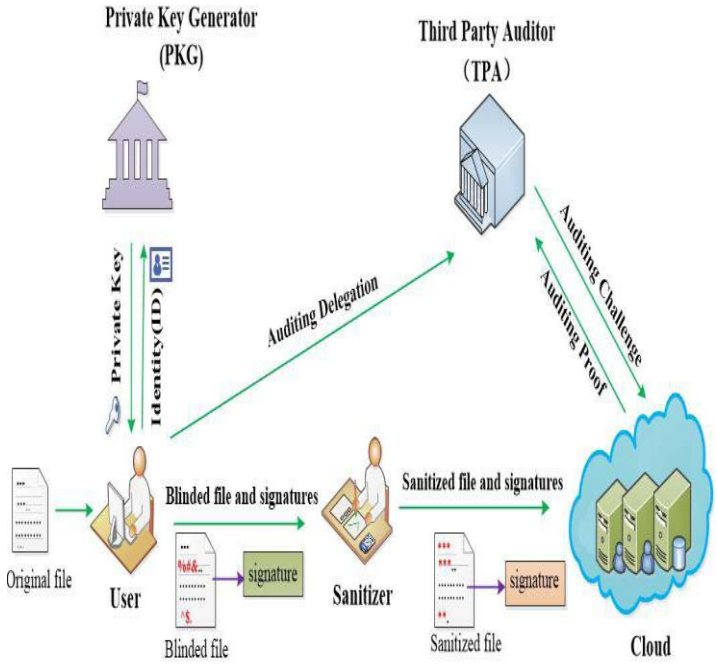

Fig 1 System model

In addition, the user generates a file tag, which is used to ensure the correctness of the file identifier name and some verification values. The user also computes a transformation value that is used to transform signatures for sanitiser. Finally, the user sends the blinded file, its corresponding signatures, and the file tag along with the transformation value to the sanitiser. When the above messages from the user are valid, the sanitiser firstly sanitizes the blinded data blocks into a uniform format and also sanitizes the data blocks corresponding to the organization's sensitive information to protect the privacy of the organization.

4. IMPLEMENTATION

The system model involves five kinds of different entities: the cloud, the user, the sanitiser, the Private Key Generator (PKG) and the Third Party Auditor (TPA),

A.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.

B.User

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

C.Sanitizer

The sanitiser 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.

D.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*.

E.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 user firstly blinds the data blocks corresponding to the personal sensitive

information of the file and generates the corresponding signatures. These signatures are used to guarantee the authenticity of the file and verify the integrity of the file. Then the user sends this blinded file and its corresponding signatures to the sanitiser. After receiving the message from the user, the sanitiser sanitizes these blinded data blocks and the data blocks corresponding to the organization's sensitive information and then transforms the signatures of sanitized data blocks into valid ones for the sanitized file. Finally, the sanitiser sends this sanitized file and its corresponding signatures to the cloud. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing. When the TPA wants to verify the integrity of the sanitized file stored in the cloud, he sends an auditing challenge to the cloud. And then, the cloud responds to the TPA with an auditing proof of data possession. Finally, the TPA verifies the integrity of the sanitized file by checking whether this auditing proof is correct or not.

5. CONCLUSION

In this paper, 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.

REFERENCE

[1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan. 2012.

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

[3] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secure., 2007,
pp. 584–597.

[4] H. Shacham and B. Waters, "Compact proofs of retrievability," J. Cryptol., vol. 26, no. 3, pp. 442–483, Jul. 2013.

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

[6] S. G. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," Comput. Electr. Eng., vol. 40, no. 5, pp. 1703–1713, 2014.

[7] C. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, "Symmetrickey based proofs of retrievability supporting public verification," in
Computer Security—ESORICS. Cham, Switzerland: Springer, 2015, pp. 203–223.

[8] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," J. Netw. Comput. Appl., vol. 82, pp. 56–64, Mar. 2017.

[9] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 6, pp. 754–764, Jun. 2010.

[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secure. Privacy Commun. Netw., 2008, Art. no. 9.

[11] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secure., 2009, pp. 213–222.

[12] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, May 2011.

[13] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," IEEE Trans. Inf. Forensics Security, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.

[14] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," IEEE Trans. Inf. Forensics Security, vol. 11, no. 6, pp. 1362–1375, Jun. 2016.

[15] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," IEEE Trans. Inf. Forensics Security, vol. 12, no. 8, pp. 1931–1940, Aug. 2017.

Received 4 Jan 2021
Revised 28 Feb 2021
Accepted 2 April 2021
Published Online 27 April 2021