International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

Authorizing Identity based Integrity Auditing and Data Assigning with Responsive Information Hiding for Cloud

Siva Priya, (M.E)[1], S.Vidhya M.E.,(Ph.D)[2]
Ponjesly College of Engineering, India
Assistant Professor, Department of CSE, Ponjesly College of Engineering, India

Abstract: Authorizing based on Identity-based cloud storage auditing schemes can check the security of cloud data. It contains the confidential information that must be protected from unauthorized access. To provide the privacy for an individual or organization, the sensitive information's are need to be hided. Hiding the sensitive information in the cloud makes the data in the cloud more secure. In the proposed system the Private Key Generator (PKG) generate the authentication key to the data owner and the data user. By using the authentication key, the data owner and the data user enter into the cloud to upload and to download the data from the cloud. The data are encrypted and the sensitive information's are hided before uploading in the cloud. Third Party Auditor (TPA) performs the auditing process in the cloud. This 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, which makes the auditing system to work in efficient manner.
Index Terms: Cloud storage; Data integrity auditing; Data sharing; Sensitive information hiding

I. INTRODUCTION

Cloud computing is the on demand availability of the computer system resources, especially data storage and computing power, without the direct active management by the user. 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. In order 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 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) 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.

II. 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. Solomon 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. [13–15] proposed key-exposure resilient remote data integrity auditing schemes based on key update technique [16].
Recently, cloud computing rapidly expands as an alternative to conventional computing due to it can provide a flexible, dynamic and resilient infrastructure for both academic and business environments. In public cloud environment, the client moves its data to public cloud server (PCS) and cannot control its remote data. Thus, information security is an important problem in public cloud storage, such as data confidentiality, integrity, and availability. In some cases, the client has no ability to check its remote data possession, such as the client is in prison because of committing crime, on the ocean-going vessel, in the battlefield because of the war, *et al*. It has to delegate the remote data possession checking task to some proxy. In this paper, Huaqun Wang et al. [33] study proxy provable data possession (PPDP). In public clouds, PPDP is a matter of crucial importance when the client cannot perform the remote data possession checking. We study the PPDP system model, security model and design method. Based on the bilinear pairing technique, we design an efficient PPDP protocol. Through security analysis and performance analysis, our protocol is provable secure and efficient.
In this work, we aim to make attribute-based encryption (ABE) more suitable for access control to data stored in the cloud. Mate Horvath et al. For this purpose, we concentrate on giving to the encrypt or full control over the access rights, providing feasible key management even in case of multiple independent authorities, and enabling viable user revocation, which is essential in practice. Our main result is an extension of the decentralized CP-ABE scheme of Lewko and Waters [6] with identity-based user revocation. Our revocation system is made feasible by removing the computational burden of a revocation event from the cloud service provider, at the expense of some permanent, yet acceptable overhead of the encryption and decryption algorithms run by the users. Thus, the computation overhead is distributed over a potentially large number of users, instead of putting it on a single party (e.g., a proxy server), which would easily lead to a performance bottleneck. The formal security proof of our scheme is given in the generic bilinear group and random oracle models.
Attribute-Based Encryption (ABE) with outsourced decryption not only enables fine-grained sharing of encrypted data, but also overcomes the efficiency drawback (in terms of ciphertext size and decryption cost) of the standard ABE schemes Baodong Qin, Robert H. Deng, et al.[37] Specifically, an ABE scheme with outsourced decryption allows a third party (e.g., a cloud server) to transform an ABE ciphertext into a (short) El Gamal-type ciphertext using a public transformation key provided by a user so that the latter can be decrypted much more efficiently than the former by the user. However, a shortcoming of the original outsourced ABE scheme is that the correctness of not be verified by the user. That is, an end user could be cheated into accepting a wrong or maliciously transformed output. In this paper we first formalize a security model of ABE with verifiable outsourced decryption by introducing a verification key in the output of the encryption algorithm. Then, we present an approach to convert any ABE scheme with outsourced decryption into an ABE scheme with verifiable outsourced decryption. The new approach is simple, general and almost optimal. Compared with the original outsourced ABE, our verifiable outsourced ABE neither increases the user's and the cloud server's computation costs except some non-dominant operations (e.g., hash computations), nor expands the ciphertext size except adding a hash value (which is less than 20 byte for 80-bit security level). We show a concrete construction based on Green et al.'s ciphertext-policy ABE scheme with outsourced decryption, and provide a detailed performance evaluation to demonstrate the advantages of our approach.
Ciphertext-policy attribute-based encryption (CP-ABE) has been a preferred encryption technology to solve the challenging problem of secure data sharing in cloud computing Sphurti Atram, N. R. Borkar et al The shared data files generally have the characteristic of multilevel hierarchy, particularly in the area of healthcare and the military. However, the hierarchy structure of shared files has not been explored in CP-ABE. In this paper, an efficient file hierarchy attribute-based encryption scheme is proposed in cloud computing. The layered access structures are integrated into a single access structure, and then, the hierarchical files are encrypted with the integrated access structure. The ciphertext components related to attributes could be shared by the files. Therefore, both ciphertext storage and time cost of encryption are saved. Moreover, the proposed scheme is proved to be secure under the standard assumption. Experimental simulation shows that the proposed scheme is highly efficient in terms of encryption and decryption. With the number of the files increasing, the advantages of our scheme become more and more conspicuous.

Attribute-Based Encryption (ABE) is a public key encryption scheme that allows users to encrypt and decrypt messages based on user attributes Tianyu Zhao et al. In this paper, we consider the problem of constructing a ciphertext-policy attribute-based encryption (CP-ABE) scheme in a setting where the attributes distributor is also the owner of messages that are to be encrypted and shared. The CP-ABE scheme we propose bases on the Sibling Intractable Function Family (SIFF) scheme. Compared to the existing ABE schemes, the decryption of our scheme in this setting is quite fast and the ciphertext size is rather small. Our ABE system also provides a high degree of compatibility with the messages that are already encrypted when our system is set up, namely, encrypted messages can be used directly in our scheme without being re-encrypted. We compare the efficiency of our scheme with Bethencourt's work in this paper.

III. PROBLEM IDENTIFICATION

In the proposed system, the Remote Data Integrity Auditing Scheme with Refiner Method is used. In this Scheme, the sensitive information can be protected and the other information can be published. The ID was generated by the user it verifies the private key of user ID. If user ID want to retrieve the file F, it send request to the user. A refiner is used to refine the data blocks corresponding to the sensitive information of the file .It also transforms the corresponding signatures into valid ones for the refined file. Finally the refined file was verified and stored in the cloud storage which the sensitive information cannot be accessed. In this project, three packages (i.e : Data owner, Cloud server, Third Party Auditing) were used. 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 I Cloud. 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) stored and shared in the cloud usually contain patients' sensitive information (patient's name, telephone number and ID number, etc.). 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 I Cloud. 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) stored and shared in the cloud usually contain patients' sensitive information (patient's name, telephone number and ID number, 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. 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 upload 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 patient and hospital, but these encrypted EHRs cannot be effectively utilized by researchers any more. Distributing the decryption key to the researchers seems to be a possible solution to the above problem. It is infeasible to adopt this method in real scenarios due to the following reasons. Firstly, distributing decryption key 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 in the near future 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.

A. Disadvantage

1) It achieves less security and efficiency.
2) High chance of blocking authorized users.
3) Difficult to share data.

IV. PROPOSED MODELING

In the proposed system, the sensitive information is protected and encrypted by using the Attribute based Encryption (ABE) techniques. The files that are stored in the clouds are based on their attributes that is include in the authentication permission provided by the TPA. TPA provide the authentication permission to the data owner for uploading data in the cloud. 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 refiner.

After receiving the message from the user, the refiner refines these blinded data blocks and the data blocks corresponding to the organization's sensitive information, and then transforms the signatures of refined data blocks into valid ones for the refined file. Finally, the refiner sends this refined file and its corresponding signatures to the cloud. These signatures are used to verify the integrity of the refined file in the phase of integrity auditing. When the TPA wants to verify the integrity of the refined 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. The TPA verifies the integrity of the refined file by checking whether this auditing poof is correct or not.

A. Proposed Architecture Modeling

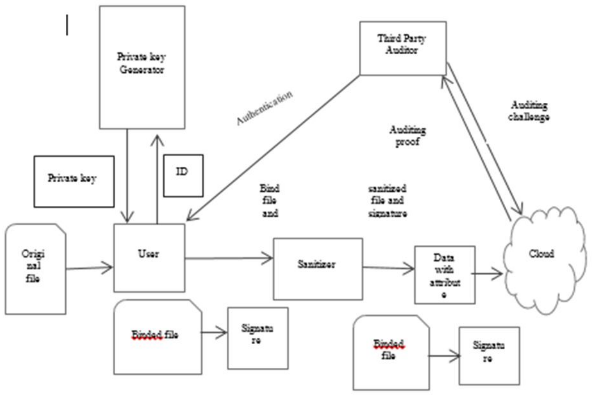


Figure 2: Architecture of Proposed Modeling

B. Implementation

Implementation is the stage of the project when the theoretical design id turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constrains on implementation, designing of methods to achieve changeover.

C. Private key Generation Module

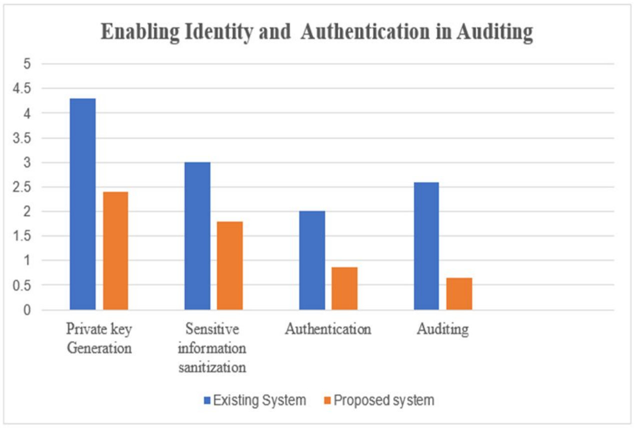
1) *User Module:* In this module, it takes input as a security parameter k. It outputs the master secret key msk and the system public parameters pp. Extract (pp;msk; ID) is an extraction method run by the PKG. It takes input as the system public parameters pp, the master secret key msk, and a user's identity ID. After receiving the user's identity ID the PKG randomly pick the value and compute the private key of the user ID. It outputs the user's private key skID. The user can verify the correctness of skID and accept it as his private key only if it passes the verification. Then the signature was generated, it takes the input as the original file F, the user's private key skID, the user's signing private key ssk and the file identifier name. The user ID randomly chooses a value and calculates a verification value . Then the user ID randomly chooses a seed as the input secret key of pseudo-random function f. The user ID employs the secret seed to calculate the blinding factor which is used to blind the data blocks corresponding to the personal sensitive information. To preserve the personal sensitive information from the refiner, the user ID should blind the data blocks corresponding to the personal sensitive information of the original file F before sending it to the refiner. The indexes of these data blocks are in set. The user ID computes the blinded data block m of the blinded file F, the user ID calculates the signature i on block m and calculate file tag. The user ID calculates a transformation value which is used to transform the signature in refination algorithm. The user ID can recover the original file F using the blinding factor. It outputs a blinded file F, its corresponding signature set and a file tag .

2) *Authentication:* The authentication permission was provided by TPA. TPA provide the permission for the data owner to upload the file based on their files attribute. The attribute based encryption is performed. By using the attribute the data user can files from the cloud.
3) *Sensitive Information Sanitization:* The refiner check the validity of file tag by verifying the signature. If the signature is valid the refiner parses to obtain file identifier name and verification values. The refiner refine the blinded data blocks and the data blocks corresponding to the organization's sensitive information. The data blocks whose indexes are in set 1have been blinded by the user ID, which will make the contents of these data blocks become messy code. The refiner can use wildcards to replace the contents of these data blocks. The refiner sends F' to the cloud, and then sends the file tag to the TPA. Finally, he deletes these messages from local storage.
4) *Integrity Auditing:* The TPA verifies the validity of file tag.The TPA will not execute auditing task if the file tag is invalid. Otherwise, the TPA parses to obtain file identifier name and verification values, and then generates an auditing challenge. After receiving an auditing challenge from the TPA, the cloud generates a proof of data possession. The TPA verifies the correctness of auditing proof.

V. RESULT AND DISCUSSION

A. *Performance Analysis*

To evaluate the identity encryption and auditing was performed. Then the authentication is provide by the TPA for uploading and downloading the data from the cloud.



B. *Experiment Setup*

Before sending the original file to the refiner, the user needs to blind the data blocks corresponding to the personal sensitive information of the file. The computation overhead of data blinding. And then the user generates the signatures for the blinded file. The computation overhead of computing signatures and the data blocks corresponding to the organization's sensitive information, and transforms the signatures of these data blocks into valid ones for the refined file. In the phase of refination, the computation overhead on the refiner side. In the phase of integrity auditing, the TPA firstly needs to generates the authentication permission and send an auditing challenge to the cloud, which only costs negligible computation overhead. And then, the cloud outputs an auditing proof to reply the TPA. The construction of the scheme is generally regarded as one of the most efficient one among all existing remote data integrity auditing schemes. In this scheme, one data block is divided into multiple sectors, which can reduce storage overhead. Actually, our scheme also can support multiple sectors. But for simplification, we only consider the situation that the file F is divided into n data blocks, and do not consider the situation that each data block is divided into s sectors. For fairness comparison, we set s=1 in scheme. It means that our scheme have the same efficiency. when processing the same file. In the phase of proof generation, the cloud's computation overhead in our scheme are the same. For the TPA in the phase of proof verification, our scheme costs more computation overhead than the existing scheme. The communication overhead incurred in the remote data integrity auditing. In the phase of integrity auditing, the TPA sends an auditing challenge to the cloud. The size of an auditing challenge bits. After receiving the auditing challenge from the TPA, the cloud generates an auditing proof to reply the TPA. The size of an auditing proof bits. Therefore, for one auditing task, the whole communication overhead

VI. CONCLUSION

In this project, an identity-based data integrity auditing scheme was proposed for secure cloud storage, which supports data sharing with sensitive information hiding. In our scheme the authentication is provide by the TPA for the data owner and the data user for uploading and downloading the files from the cloud. 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 Computing, vol. 16,no. 1, pp. 69–73, Jan 2012.
[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07, 2007, pp. 598–609.
[3] Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser.CCS '07, 2007, pp. 584–597.
[4] H. Shacham and B. Waters, "Compact proofs of retrievability," J. Cryptology, 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 Transactions on Computers, vol. 62, no. 2, pp. 362–375, 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, Jul. 2014.
[7] C. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, "Symmetric-key based proofs of retrievability supporting public verification," in Computer Security – ESORICS 2015. Cham: Springer International Publishing, 2015, pp. 203223.
[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," Journal of Network and Computer Applications, vol. 82, pp. 56–64, 2017.
[9] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 6, pp. 754–764, June 2010.
[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th international conference on Security and privacy in communication netowrks, 2008, pp. 1–10.
[11] C. Erway, C. Papamanthou, and R. Tamassia, "Dynamicprovable data possession," in ACM Conference on Computer and Communications Security, 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 Transactions on Parallel and Distributed Systems, 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 Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1167–1179, 2015.