ISSN:0975-9646

Delegation of Auditing Operation to Auditor by Data Owner with Multiple Authorities for Secure Cloud Storage

Sowmya M N, Girish

Department of Computer Networking Engineering,

The National Institute of Engineering,
Manadavady Road, Mysore-570008, INDIA

***Abstract*—In cloud storage service, users upload their data together with authentication information to cloud storage server .Using this services user can enjoy the on-demand high-quality applications and services, without burden of local data storage and maintenance .Those advantages are causes of security and privacy problem. So the integrity protection is very challenging task in cloud computing. Hence developer introduce a Third Party Auditor (TPA) to check the integrity of outsource data .In this paper we proposed a method that allow the data owner to delegate the auditing task to untrusted TPA in secure manner, that 1)the data owner can verify whether the TPA has indeed performed the specified audit task, and 2) whether the TPA did the audit task at the right time specified by the data owner. And extended our result to using multiple authorities achieves anonymous cloud data access control and provide a more security for user data. .Our security analysis shows the proposed schema are more secure and highly efficient in cloud computing.**

***Keywords*— Secure Delegation of Auditing, public auditability, Multiple -Authorities, privileges tree.**

I. INTRODUCTION

Cloud computing may be defined as delivery of product rather than service. It is a internet based computing which enables sharing of services and it is a concept of computing technique, by which computer resources are provided dynamically via Internet. It attracts considerable attention and interest from both academia and industry. However, it also has at least two challenges that must be handled before applied to our real life. First of all, data confidentiality should be guaranteed. Secondly, personal information (defined by a user's attributes) is at risk because one's identity is authenticated according to his information. As people are becoming more concerned about their privacy these days, the privacy-preservability is very important. 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[1],[4]. 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. Furthermore, several methods even support to verify multiple users' data together in batch. However, without proper enforcement, public verifiability, would give users a false impressions that their data were safe in the cloud storage. There are many ways that public verifications can be misused, for example:

Too many volunteers help to verify a CSP's(Cloud service provider) storage remotely, bringing in too much unnecessary burden on the CSP and possibly resulting in Denial of Service attack to the CSP, The distribution of verification time is uneven, too many verifications for some periods and too few verifications for other periods;

In this paper, we attempt to provide a solution which enables the data owner to securely delegate the auditing task to a potentially untrusted third party auditor (TPA). Extending our result to introduced a multiple-authority system[3], where each user has an ID and they can interact with each key generator (authority) using different pseudonyms. One user's different pseudonyms are tied to his private key[2], but key generators never know about the private keys, and thus they are not able to link multiple pseudonyms belonging to the same user. In fact they are even not able to distinguish the same user in different transactions. Also, the whole attributes set is divided into N disjoint sets and managed by N attributes authorities [9][10]. That is, an attribute authority will only issue key components which it is in charge of. In this setting, even if an authority successfully guesses a user's ID, it knows only parts of the user's attributes, which are not enough to figure out the user's identity. In this schema CP-ABE (Ciphertext-PolicyAttribute-BasedEncryption) [5] technique used the private key is distributed to users by a trusted central issuer only once. The keys are identified with a set of descriptive attributes, and the encrypter specifies an encryption policy using an access tree so that those with private keys which satisfy it can decrypt the cipher text. And privilege Tree [8] contain the data file has several operations executable on itself, but some of them should be restricted only to authorized users. For example, {*Read mine, Read all, Delete, Modify, Create*} is a privileges set of students' grades. Then, reading student grades is allowed to her and her professors, but all other privileges should be authorized only to the professors, so we need to grant the "*Read mine*" to student and all other to the professors. Every operation is associated with one privilege p, which is described by a privilege tree Tp. If a user's attributes satisfy Tp, he is granted the privilege p. By doing so, we not only control the file access but also control other executable operations, which makes the file controlling fine-grained and thus suitable for cloud storage service.

II. PRELIMINARY AND BACKGROUND

A Proof of Retrievability (POR) scheme consists of four algorithms KeyGen, DEnc, Prove, Verify:

(pk; sk) ← KeyGen. Given security parameter the randomized key generating algorithm outputs a public-private key pair (pk; sk)

M← DEnc(M; sk). Given a data file M and the private key sk, the encoding algorithm DEnc produces the encoded file M.

r ← Prove(M ; c; pk): Given an encoded file M , a challenge c and the public key pk, the prover algorithm Prove produce a response/proof r.

(accept; reject) ← Verify(c; r; pk): Given a challenge c, a response/proof r and the public key, the verifying algorithm Verify will output either accept or reject.

A. Blind Technique

Wang et al[4]. Proposed a blind technique in addition to Shacham and Water's scheme attempting to achieve privacy-preserving third party auditing. With their blind technique, the prover masks the proof for a challenge with some randomness and the verifier is still able to verify the validity of the masked proof.

III. MOTIVATIONS AND OBSERVATIONS

A. *Integrity verification and data accessing cannot be completely separated.*

Before accessing data, the user has to verify the downloaded data locally, even if he/she has performed the auditing task periodically or delegated the audit task to a third party. Without local checking, a malicious CSP can inevitably cheat and provide the user altered data with non-negligible probability, no matter what remote integrity check schemes are deployed. Suppose the data owner/verifier will initial poly number of interactions with the CSP, and each interaction is either for verification or data retrieval. Furthermore, if CSP is able to distinguish verification from retrieval, he/she will win with even higher probability.

B. Timing is essential in remote integrity check of cloud storage.

If data are corrupt and no longer retrievable in the cloud server, the data owner know this at soon as possible. So that he/she can take counter-measure in time, to minimize the loss. Without timing concern, users can always download the (partial) data from CSP, verify the integrity of data locally, before using the data. If data owner eventually does not retrieval the data, it does not matter whether the data in the cloud is intact or not. A dishonest TPA may have incentive to perform all audit tasks specified by the data owner within a short period, to decrease his/her Internet connection time.

C. Auditors themselves should be audited.

Auditors may have incentive to execute a partial audit task specified by the data owner, or execute the audit task at time which it their own interests. Furthermore, the CSP may collude with the auditors or just create some Sybil identities, who are volunteer to be the auditors. In order to ensure they faithfully accomplish their promise on auditing the cloud storage at right time, auditors themselves should be audited.

IV. LIMITATION OF PREVIOUS WORK

Wang et al.[4] [6] proposed to a method to protect data confidentiality against the TPA. However, their security model is weak:

- In their model, both CSP and TPA are semi-trusted. Precisely, they trust TPA in auditing and trust CSP in data confidentiality; they do not trust TPA in data confidentiality and do not trust CSP in maintaining data integrity.
- Their privacy protection is also weak. Although they showed that their blind technique prevents TPA from recovering the original data through auditing process they did not analyze whether their blind technique reveals any partial information about the original data. We found that, in their scheme, the TPA is able to verify that whether the original data equal to any particular value, with only information that he/she is allowed to access. Such ability to evaluate equality predicate over blinded data could be a serious vulnerability when the entropy of some data blocks is very low. Although this issue can be mitigated by compressing the whole data file before outsourcing, this potential weakness may suggest that a stronger privacy requirement is desired in such applications.

V. DEFINITIONS OF OUR SCHEME

A. *System Overview*

In our system, there are five types of entities: N Attribute Authorities , Cloud Server, Data Owners and Data Consumers, TPA .A user can be a Data Owner and a Data Consumer simultaneously.

Authorities assumed to have powerful computation abilities, which are supervised by government offices since keys act as IDs and partially contain users' PII (Personally Identifiable Information). The whole attribute set is divided into N disjoint sets and controlled by each authority. One practical method to divide the attributes set is to divide them by category (e.g., {Sex: Male, Female}, {Nationality: Indian, Chinese, Japanese}, {University: mysore university, Peking University}, {Position: Professor, Ph.D Student, Master Student}). In this way, since each authority is aware of only one type of attribute, no useful information is leaked. The authorities jointly compute a system-wide public key, and individually compute their master keys at the initialization phase. The public key is used for all operations within the system, and the master keys are used by each attribute authority when he generates private keys for Data Consumers.

Data Owner Who has a large amount of data to backup **,** owner achieves public key from any one of the authorities, and he uses the public key to encrypt the data file before

outsourcing it to the Cloud Servers. The third-party auditor is audit the user outsource data and who has expertise and capabilities that cloud users do not have and is to assess the cloud storage service reliability on behalf of the user upon request. The Cloud Server, who is assumed to have adequate

storage capacity, does nothing but store them. Newly joined Data Consumers request private keys from all of the authorities, and they do not know which attributes are controlled by the authorities. On the other hand, authorities do not know which Data Consumers are interacting with them because each of them knows only a part of Data Consumers attributes. When the Data Consumers request their private keys from the authorities, authorities jointly create corresponding private key and send it to them .All *Data Consumers* are able to download any of those data files, but only those whose private keys satisfy the privilege tree Tp can execute the operation associated with privilege p. When a user wants to execute a specific operation upon a data, he should satisfy the relevant privilege tree Tp and gets verified by the *Cloud Server. The* server is delegated to execute an operation p if and only if the user's privilege is verified through the privilege tree.

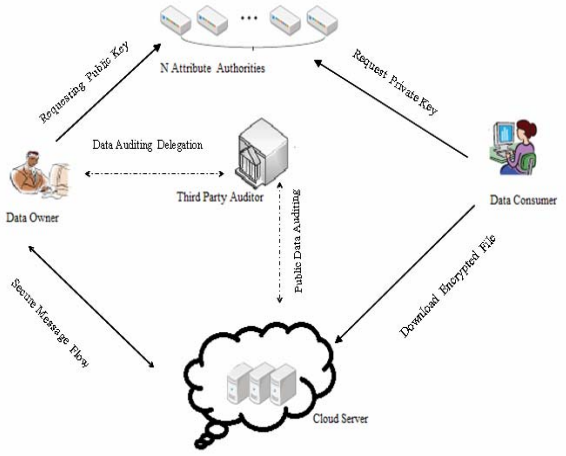

Figure 1: Our System Model

D. Our Scheme.

Our scheme contain Release Plan, Execute Plan, Review Plan Time Server, Receive Server .Setup, Key generate .Encrypt Algorithm.

1) Time Server

A time-server is associated with a domain T of timestamps and an CP-ABE public-private key pair (tpk; tsk), where tpk is publicly available and tsk is kept secret by the time server. At each time point the time server broadcasts the decryption key w.r.t. the attribute . The time server does nothing else.

2) Receive Server

A receive-server has a large storage. Once receiving a message Msg designating for receiver Rev from a sender Snd at time t, the receive-server will record (t; Rev; Snd; Msg) in his/her storage. The receiver-server also allows the designated receiver to retrieve their message. In real world application, we may adopt a reliable email server to play the role of receive-server.

3) Setup

At the system initialization phase, any one of the authorities chooses a bilinear group of prime order with generator and publishes it. Then, all authorities independently and randomly pick and send to all other authorities who individually compute

Then, every authority randomly picks N − 1 integers and computes secrete key. Each secrete key is shared with each other authority . An authority after receiving N −1 pieces computes its secret parameter.

4) Key Generate

When a new user wants to join the system, he requests the private key from all of the authorities by following this process which is composed of two phases.

1) Attribute Key Generation: For any attribute randomly picks attribute key and individually compute the partial private key .Then, all of the authorities randomly picks and compute secrete parameter and share it with others. Then, authorities merge the partial private keys and sent to the user

2) Key Aggregation: User, after receiving key from authorities then aggregates the components as his private key.

5) Encrypt

Before uploading the data to cloud server user can divide the data file into blocks and encrypt each block using a semantic secure public key encryption scheme (or a symmetric encryption scheme, e.g. AES). And send it into TPA.

6) Release Plan

Data owner choose m time points tm from T, and chooses m -challenges (c1; : : : ; cm) at random, then data owner encrypts challenges using the Timed-Release Encryption scheme and owner sends the coded audit plan to TPA .

7) Execute Plan

At time t, TPA receives the decryption key for time t, and decrypt challenges .TPA who is taking the role of verifier, interacts with the CSP , who is taking the role of prover, to execute the interactive algorithm. TPA obtains reply for the challenge from CSP. Then TPA sends Msg to Receive Server.

An authorized verifier may interact with CSP by running algorithm to audit the integrity owner's data in CSP storage.

8) Review Plan

Data owner retrieves msg from the Receive Server.

VI. SECURITY ANALYSIS

A. . *User's Identity Information Confidentiality*

The attributes, which contain a user's identity information, are separately controlled by different attribute authorities. Therefore, a user's attributes information is securely protected.

B. *Data Confidentiality against Collusion Attack*

In order to access a plaintext, attackers must recover which can be recovered only if the
attackers have enough attributes to satisfy the tree .When two different keys' components are combined, the combined key cannot go through the polynomial interpolation in the decryption algorithm due to the randomization. Therefore, at least one key should be valid to satisfy the privilege tree.

VII. FUTURE ENHANCEMENT

The full-fledged implementation of the mechanism on commercial public cloud as an important future extension, which is expected to robustly cope with very large scale data and thus encourage users to adopt cloud storage services more confidently.

VIII. CONCLUSIONS

We proposed a solution that allows the owner of data stored in a cloud storage to delegate the auditing task to a potentially untrusted third party verification in a secure way. That is, the data owner can verify whether the TPA did perform the audit task at the right time as specified by the data owner. In another words, we provide a method allowing the data owner to audit to the auditor. In another words, we provide a method allowing the data owner to audit to the auditor. And using multiple authorities we provide more data security for user.

ACKNOWLEDGMENT

We thank Girish, the Associate professor for his support of this effort and assistance throughout this project.

REFERENCES

[1] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, Senior , and Wenjing Lou," Privacy-Preserving Public Auditing for Secure Cloud Storage" IEEE Transactions on Computers, vol. 62, no. 2, February 2013

[2] Taeho Jung§, Xiang-Yang Li§, Zhiguo Wan† and Meng Wan "Privacy Preserving Cloud Data Access With Multi-Authorities" inarXiv:1206.2657v6[cs.CS] 11 April 2013

[3] M. Chase and S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 121– 130.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[5] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," Advances in Cryptology–EUROCRYPT 2011, pp. 568–588, 2011.

[6] T. Jung, X. Mao, X. Li, S. Tang, W. Gong, and L. Zhang, "Privacy preserving data aggregation without secure channel: multivariate polynomial evaluation," in IEEE INFOCOM, 2013.

[7] 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 and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in IEEE INFOCOM, 2010, pp. 1–9.

[9] J. Liu, Z. Wan, and M. Gu, "Hierarchical attribute-set based encryption for scalable, flexible and fine-grained access control in cloud computing," Information Security Practice and Experience, pp. 98–107, 2011.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, 2010, pp. 261– 270.