Fully Secure and Efficient Data Sharing with Attribute Revocation for Multi-Owner Cloud Storage

Hari Baskar Sampath
Dept. of Information Technology
SRM University
Chennai, INDIA
baskarsampath@hotmail.com

J.Jeysree
Asst. Professor (Sr.G)
Dept. of Information Technology
SRM University
Chennai, INDIA
jey.sree@gmail.com

Abstract-Now a days, a lot of users are storing their data's in cloud, because it provides storage flexibility. But the main problem in cloud is data security. Cipher text-Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners more direct control on access policies. In this work to propose a data access control for multi-authority for verifying the integrity of an un-trusted and outsourced storage by third party auditor. In addition, this project propose method based on probabilistic query and periodic verification for improving the performance of audit services. It ensures efficiency of security by protecting from unauthorized users. These experimental results not only validate the effectiveness of these approaches, but also show our audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata.

Keywords- *Access control, multi-authority, audit, attribute revocation, cloud storage.*

*****

I. INTRODUCTION

In recent years, the cloud computing technologies has developing technology in IT world. The cloud computing has many features like access anywhere from anywhere and at any time. The cloud computing has large data storage or data centers and also uses large servers for web application and services. Access control and authentication methods ensure the authorized users to access the data. But, its main concern is data security. Because, the cloud server cannot be fully trustworthy by data owners, they cannot believe on servers to do access control. Ciphertext-policy Attribute based encryption (CP-ABE) is one of the recent technologies for data access control in cloud storage, because it provides the data owner more direct control on access policies. In this scheme, the attribute authority is responsible for the maintaining the attribute and also responsible for key distribution for the attribute. The certificate authority is activates the user and attribute authority registration. The CA can be the Human resource department in an organization, registration office in a university, etc. The data owner encrypts depending on the access policies and attribute. The access policies prevents the unauthorized person to access the data.

Multi-Authority CP-ABE is suitable for data access control of cloud data storage. The user may be hold n number of attributes from any attribute authority. The data owners can share the data with attribute based encrypted method along with the access policy. For Example, A Human resource department, the data owners share the data by using the access policy "Project Manager AND Team Leader" or "Project Manager OR Team Leader", where the attribute "Project Manager" have different access rules and the attribute "Team Leader" have different access rules. It is very difficult to apply directly on multi-authority CP-ABE method to cloud storage because the attribute revocation issues for users. This issue happens when the revoked user cannot decrypt any ciphertext that requires the revoked attribute to decrypt (Backward security) and the newly entered users can also decrypt the previous published ciphertext if its public key and sufficient attributes (Forward security).

CP-ABE:

One of the most suitable technologies for data access control in cloud storage systems is Cipher text-Policy Attribute-based Encryption (CP-ABE). It provides the data owner to direct control on access policies. The Authority in this scheme is responsible for key distribution and attribute management. The authority may be the university Administration office, Staff maintenance (Human resource-HR) department in a company, etc. The data owner in CP-ABE scheme defines the access policies and encrypts data depending on the policies.

CP-ABE TYPES:

In CP-ABE scheme for every user will be issued a secret key reflecting its attributes. A user can decrypt the data only when its attributes to satisfy the access policies.
There are two types of CP-ABE systems:

- Single-authority CP-ABE
- Multi-authority CP-ABE

In Single-authority CP-ABE method, where all the attributes are managed by only one a single authority. In a Multi-authority CP-ABE scheme where attributes are from different attribute authorities. This method is more suitable for data access control of cloud storage systems. Data users contain attributes should be issued by multiple authorities and data owners. Data users may also share the data using access policy defined over attributes from different authorities.

In our scheme, the data owner does not required to trust the server. Because, the key is based on attribute and maintained by the attribute authority. We designed new revocation method for multi-authority CP-ABE. Then, we apply them to design a fully secure and efficient data sharing for multi-authority scheme.

The important advantages of this work can be summarized as follows,

i. We proposed third party auditor (TPA) which used for auditing the data.
ii. We develop a new revocation method for user attribute revocation.

II. SYSTEM MODEL

We designed a data access control for Multi-Authority cloud storage as fig (1) shows, there are six types of entities in system: The cloud server(server), the data owner, the attribute authority (AA), the Certificate authority (CA), the data users (User) and the third party auditor (TPA).

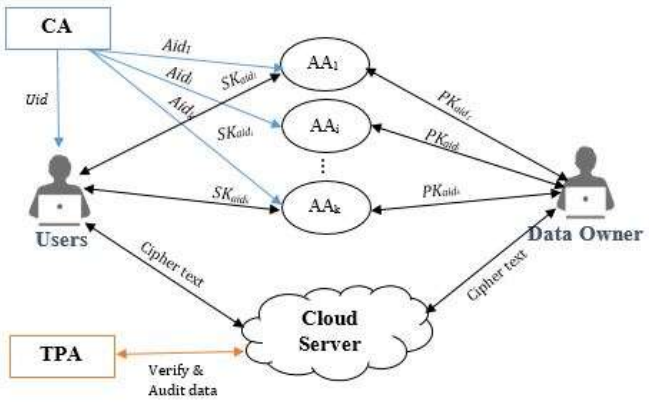

Fig. (1)

The CA is a global trusted certificate authority, which accepts the user and AA registration. The CA is distributes the global public key and global secret key for each legal user. But it is not involved in any attribute management and also creation of secret keys that are associated with attributes. For example, CA is like a Unique Identification Authority of India (UIDAI), for Indian government. Each user will be issued a Unique Identification Number (AADHAAR Number) as its Identity.

Every AA is a separate attribute authority. AA is responsible for create an attribute and revoke the attributes for user. The attribute is created by the role or identity of user. Each AA has maintaining the n number of attributes. AA generates the public key and private key for the each attribute it manages.

The user has a global identity in the system. They may be create a set of attributes which comes for multiple attribute authority and also receives a secret key for their attributes.

The data owners encrypts the data along with the access policies with the set of public key of the attributes. The data owner updates the ciphertext into the cloud server. The user can decrypt when the attributes satisfy the access policy along with the ciphertext, the user can decrypt the ciphertext.

The cloud server maintains the data owner's ciphertext. The server does not edit or updates any contents in the ciphertext.

Third party auditor (TPA) is used to audit the files on the cloud server. It increases more security for the data, because it prevents data from the attackers and hackers.

A. CP-ABE Alogirthm

A CP-ABE scheme have four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup (λ; U): The setup algorithm takes input as security parameter and attribute universe description. It outputs the global public parameters PK and a global master key MK.

Encrypt (PK; M; A): The encryption algorithm takes as input the public parameters PK of attributes, a message M, and an access structure A over the involved attributes. The algorithm will encrypt M and produce a ciphertext (CT) that only a user having a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains A.

Key Generation (MK; S): The key generation algorithm takes as input the global master key MK and a set of attributes S that clarify the key. It outputs a private key SK.

Decrypt (PK; CT; SK): The decryption algorithm takes as input the public parameters PK, a ciphertext (CT), which contains an access policy A, and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the ciphertext and return a message M.

B. Security model

Security Model for CP-ABE:

Setup: The Challenger runs the Setup algorithm and gives the public parameters, PK to the attacker.

Phase 1: The attacker makes repeated private keys corresponding to sets of attributes $S_1 \ldots\ldots S_{q1}$.

Challenge: The attacker submits two equal length messages M_0 *and* M_1. In addition the attacker gives a challenge access structure such that none of the sets $S_1 \ldots S_{q1}$ from Phase 1 satisfy the access structure. The attacker flips a random coin p, and encrypts M_b under. The ciphertext CT_* is given to the attacker.

Phase 2: Phase 1 is repeated with the restriction that none of sets of attributes $S_{q1+1} \ldots\ldots S_q$ satisfy the access structure corresponding to the challenge.

Guess: The attacker outputs a guess p′ of p.

The advantage of an attacker A in this game is defined as

Pr[p′ = p] – 12. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition: A multi-authority attribute revocable ciphertext-policy attribute-based encryption scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above game.

III. FRAME WORK

The data access control for Multi-Authority cloud storage system consists following methods.

1) System Initialization

- **CA Setup** (1λ): (GMK, GPP, (GPK'_{uid}, GPK'_{uid}), (GSK_{uid};GSK'_{uid}), Certificate(uid)).

  The CA setup algorithm is run by the CA. It takes no input other than the implicit security parameter λ. It generates the global master key GMK of the system and the global public parameters GPP. For each user uid, it generates the user's global public keys (GPK_{uid}, GPK'_{uid}), the user's global secret keys (GSK_{uid} , GSK'_{uid}) and a certificate Certificate (uid) of the user.

- **AASetup** (U_{aid}):(SK_{aid}, PK_{aid}, {VK_{xaid}, PK_{xaid} }$_{xaid \in Uaid}$).

  The attribute authority setup algorithm is run by each attribute authority. It takes the attribute universe U_{aid} managed by the AA_{aid} as input. It outputs a secret and public key pair (SK_{aid}, PK_{aid}) of the AA_{aid} and a set of version keys and public attribute keys {VK_{xaid}, PK_{xaid} }$_{xaid \in Uaid}$ for all the attributes managed by the AA_{aid}.

2) Attribute Authority's key generation and management

- **Secret Key Distribution** A randomized algorithm takes as input the authority's secret key SK, a user u's UID, and a set of attributes A_{ku} in the authority AA_k's domain (We will assume that the user's claim of these attributes has been verified before this algorithm is run, $A_u = \{A_{ku}, k = 1, \ldots, n\}$). Output a secret key Du for the user u.

- **Access issue id Distribution** The collected attributes from all attribute authorities (A_a) will be sent to the users for the encryption purpose.

3) Data Encryption:

The data owner runs the encryption algorithm to encrypt the content keys. By using symmetric encryption method the data is encrypted with content keys. A randomized algorithm takes as input a set of public key of attributes involved in encryption, a message M, the global public parameters GPP and outputs the ciphertext C.

4) Data Decryption:

The users first run the decryption algorithm and use them to decrypt data's from the ciphertext C. It takes input the ciphertext C, it have access policy with itself for verifying the access rules of the users. If the access policy is satisfied with the users attribute, the decryption algorithm will decrypt the ciphertext C.

5) Attribute revocation:

The attribute revocation has been solved by assigning new version key VK for non-revoked attribute. It takes as inputs the secret key of Attribute authority, revoked attribute id and current version key. Its outputs as new version key and new attribute key.

IV. OUR DATA ACCESS CONTROL SCHEME

A. System initialization

System Initialization have two following steps:

Step 1: CA Setup
Taking input as security parameter, the CA sets up the system using the CAsetup Algorithm. The CA registers both user and AA.

- **User Registration:** During system initialization each and every user should register to CA. The global unique user id *uid* is assigned to user by the CA, if the user is a legal user.
- **AA Registration:** During system initialization the AA should register to CA. The CA assigns a global attribute authority identity *aid* if the AA is the legal authority.

Step 2: AA Setup

In this algorithm, the set of user attributes and data owner attributes are stored in data set, which provides the secret key obtained by matching the public key pair AA_{aid} as input.
$SkeyGen(GPP,GPK_{uid},GPk_{uid},GSK_{uid},SK_{aid},S_{uid,aid}\ldots)$
$=\{GPK,(PK_{aid1..n})$ With $_{uidK}\}$
$=SKuid_naid_n$

B. Secret key generation

When data owners outsource their data with some attributes and is encrypted by attributes identity (aid) then it authenticates with user identity (uid), which is issued by CA.

$GPK \rightarrow (PKuid1,aid1$
$= g_ir_1uid, aid,\ldots g_nr_nuid_naid_n)$
$=GPK_{uid_1\ldots n,\ aid1..n}.$

The secret key $SK_{uid,aid}$ only contains the first component $K_{uid,aid}$, if the user uid does not hold any attribute from AA_{aid}.

C. *Data encryption*

Before outsourcing the data's to cloud, the data owner first partitions the data into several components according to logical granularities as $m=\{m_1,\ldots m_n\}$. For example, data can be partitioned into {name, address, employee, salary, contact number}, next the data components is encrypted with different content keys$\{k_1,\ldots..,k_n\}$ using symmetric encryption method, last the access structure mechanism M_i is defined for each content key $k_i(i=1,\ldots,n)$. The encryption algorithm takes GPP as input, a collection of public keys for all AAs and outputs the ciphertext

$CT= GPP,\{PK_{aidk}\}\ _{aidk}=k(\Pi_{aid\in AAs}PK\ aid_k=PK_{aid1..n}$

D. *Data decryption*

The user can obtain the content key only when it satisfies the access structure defined in the ciphertext CT. The decryption is as follows

Decrypt(CT,GPKuid,GSKuid{SKuid,aid}→K
$= (\Pi_{aid\in AAs}K'aid_kuid_k)$
$=(\Pi_{aid\in AAs}g_{uid,r_{uid..n}})$
$=CT,GPK_{uid},GSK_{uid}$
$=K_{uid.}$

E. *Third Party Auditor(TPA)*

We consider a TPA for auditing the data in cloud storage, as described in Fig. 2. The users shared the data to the third party auditor for verifying the data. TPA checks the modification of data in server. It compares the original data with the modified data by attacker or misbehave by unauthorized person and it maintains the log file of user's accessing data.

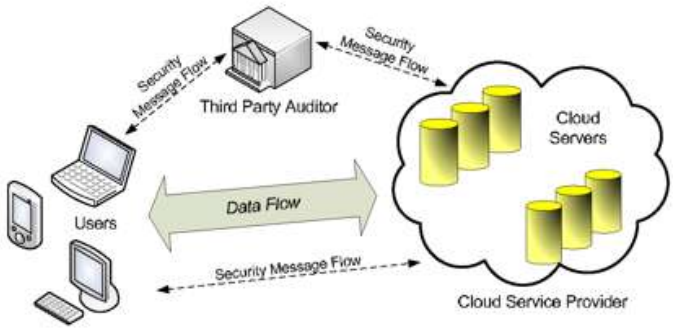

Fig. (2)

V. CONCLUSION

In this paper, we proposed an effective attribute revocation method for the Multi-authority CP-ABE method. Also, we proposed third party auditor can audit the data for data loss and attack in the multi-authority CP-ABE method. We construct the effective data access method for multi-authority cloud storage. This technique, which can be applied in any social networks and cloud data center's etc.

REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, 2007.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., 2009.

[3] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B.Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," in Proc. Advances in cryptology-EUROCRYPT'10, 2010, pp. 62-91.

[4] M. Li, S. Yu, Y. Zheng, K. Ren, andW. Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," IEEE Trans. Parallel Distributed Systems, vol. 24, no. 1, pp. 131-143, Jan. 2013.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS '10), 2010.

[6] J. Hur and D. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Transactions on Parallel and Distributed Systems, 2010.

[7] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," in Proc. 10th IEEE Int'l Conf. TrustCom, 2011, pp. 91-98

[8] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in ACM Conference on Computer and Communications Security, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM, 2010, pp. 735–737.

[9] C. Dong, G. Russello, and N. Dulay, "Shared and Searchable Encrypted Data for Untrusted Servers," J. Computer Security, vol. 19, pp. 367-397, 2010.

[10] S. Vishnupriya, C. swathi and Lina Dinesh, "Improved Privacy of Cloud Storage Data users by Using Enhanced Data Access Control Scheme for Multi-Authority Cloud Storage," in International Journal of Computer Science & Communication Networks, vol 4,2014, pp 165-168.

[11] Kan Yang and Xiaohua Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," in IEEE transactions on parallel and distributed systems, vol. 25, no. 7, july 2014, pp 1735-1744.