International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

Distributed data access control into multi authority with a central certified authority in cloud storage systems

Eeresh M.S
Dept. of Computer Science and Engg SIT Mangalore, Karnataka
Visvesvaraya Technological University
email:vmsagar99@gmail.com

Mohan K
Dept. of Computer Science and Engg SIT Mangalore, Karnataka
Visvesvaraya Technological University
e-mail: hamohax@gmail.com

***Abstract* - Data access control is an effective way to ensure data security in the cloud. However, due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Ciphertext-policy attribute-based encryption (CP-ABE) is a promising technique for access control of encrypted data. However, due to the inefficiency of decryption and revocation, existing CP-ABE schemes cannot be directly applied to construct a data access control scheme for multiauthority cloud storage systems. In this paper, we propose Data access control is distributed in to multi authority with a central certified authority in cloud storage systems, an effective and secure distributed data access control scheme with efficient decryption and an efficient attribute revocation method that can achieve both forward security and backward security.**

***Index Terms* - *Access control, CP-ABE, outsourcing decryption token, attribute revocation, multiauthority cloud*.**

I.INTRODUCTION

One of the most fundamental services offered by cloud providers is data storage. It allows data owners to host their data in the cloud and rely on cloud servers to provide "24/7/365" data access to users (data consumers). Data access control is an effective way to ensure the data security in the cloud. However, due to the data outsourcing, the cloud server cannot be fully trusted to provide data access control service. To achieve data access control on untrusted servers, traditional methods usually encrypt the data and only users holding valid keys are able to decrypt. Although these methods can provide secure data access control, the key management is very complicated when more users are in the system. Data owners also have to stay online all the time to deliver keys to new users.

To achieves a fine-grained data access control. Ciphertext-Policy Attribute-based Encryption (CP-ABE) [2] is regarded as one of the most suitable technologies for data access control in cloud storage systems, because it gives the data owner more direct control on access policies and does not require the data owner to distribute keys. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. It provides a way of defining access policies based on different attributes of the requester, environment, or the data object. Especially, ciphertext-policy attribute based encryption (CP-ABE) enables an encryptor to define the attribute set over a universe of attributes that a decryptor needs to possess in order to decrypt the ciphertext Thus, each user with a different set of attributes is allowed to decrypt different pieces of data per the security policy. This effectively eliminates the need to rely on the data storage server for preventing unauthorized data access.

II.PROPOSED SCHEME AND ITS DESIGN GOALS

The main contributions of proposed system work can be summarized as follows.

1) We propose Data Access Control is distributed in to Multiauthority with a central certified authority in Cloud Storage systems, an effective and secure data access control scheme for multiauthority cloud storage systems, which is secure in the random oracle model and has better performance than existing schemes.

2) We construct a new multiauthority CP-ABE scheme with efficient decryption. Specifically, we outsource the main computation of the decryption by using a token-based decryption method.

3) We also design an efficient immediate attribute revocation method for multiauthority CP-ABE scheme that achieves both forward security and backward security.

A. Proposed Scheme Design Goals

Due to the inefficiency of computation in multiauthority systems, they cannot be directly applied to construct the data access control scheme. Basically, there are two operations in access control that require efficient computation, namely decryption and revocation.

i. *Efficiency improvement in Revocation* :
Data access in cloud storage systems is not static, as users are changing or employees are hired/fired from

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

the company, it will be necessary to change the attributes of users. To guarantee the security of attribute revocation, there are two requirements: i) Backward Security: The revoked user (whose attributes are revoked) cannot decrypt new ciphertexts that require the revoked attributes for decryption; ii) Forward Security: The newly joined users who have sufficient attributes are also able to decrypt the previously published ciphertexts. This motivates us to develop a new method that can efficiently deal with the attribute revocation of users.

ii. *Efficiency improvement in Decryption:*
In CP-ABE systems, the data consumers need to decrypt the ciphrtext file by using their secret keys. However, nowadays, data consumers may use their PCs or mobile devices (e.g., smart phones, tablets etc.) to access the cloud data, and the computation abilities of mobile devices are not as powerful as the one of PCs. So this motivates us to outsource the main computation of decryption into the cloud server, while still keep the data privacy against the cloud server.

III.SYSTEM MODEL

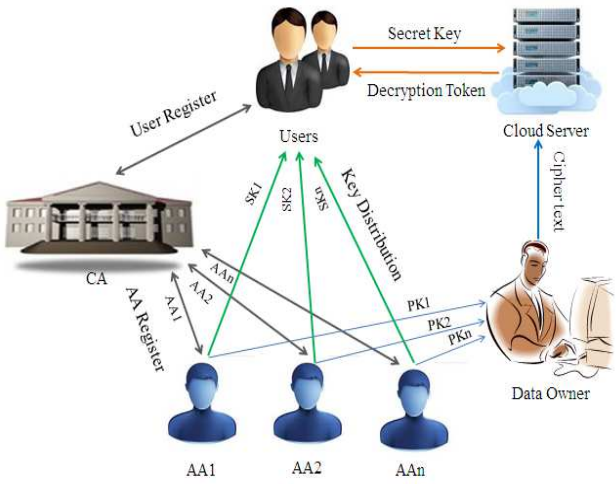

Fig.1 System Model

We consider a cloud storage system with multiple authorities, as shown in Fig. 1. The system model consists of five types of entities: a global certificate authority (CA), attribute authorities (AAs), cloud server (server), data owners (owners) and data consumers (users)

1) **CA**. The CA is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and AAs in the system. For each legal user, the CA assigns a global unique user identity to it and also generates a global secret/public key pair for this user.
2) **AA**. Every AA is needs to get registration from CA and it assigning a authority identity each AA also each is an independent attribute authority that is responsible for issuing, revoking and updating user's attributes according to its role or identity. Each AA is responsible for generating a secret key for each user reflecting their attributes.
3) **Server.** The cloud server stores owners' data and provides data access service to users. It also helps users decrypt ciphertexts by generating decryption tokens and it helps when an attribute revocation happens then data owners needs to update ciphertexts.
4) **Owners.** It is a client who owns data, and wishes to upload it into the cloud data storage center. for ease of sharing or for cost saving. Before outsourcing the data, each owner first encrypts the data with content keys by using symmetric encryption techniques. Then, the owner defines the access policies over attributes from multiple AAs and encrypts content keys under the policies. That is only when the user's attributes satisfy the access policy defined in the ciphertext the user is able to decrypt the ciphertext.
5) **Users.** Each user needs to get registration from CA and it assigning a global user identity to each user. To decrypt a ciphertext, each user may submit their secret keys issued by some AAs together with its global public key to the server and ask for a decryption token. The user then uses the received decryption token to decrypt the ciphertext along with its global secret key. Only when the user's attributes satisfy the access policy defined in the ciphertext, the server can generate the correct decryption token.

IV.SYSTEM DESCRIPTION

This section describes the overview and system description of proposed scheme, it including User Registration, AA Registration, Secret Key Generation by AAs, Data Encryption by Owners, Data Decryption by Users (With the Help of Cloud), and Efficient Attribute Revocation.

A .Overview

Although the existing multiauthority CP-ABE scheme [5] proposed by Lewko and Waters has high policy expressiveness and has been extended to support attribute revocation in [4], it still cannot be applied to access control for multiauthority cloud storage systems due to the inefficiency of decryption and revocation. Thus, the main challenge is to construct a new underlying multiauthority CP-ABE scheme that supports efficient decryption and revocation.

To design a multiauthority CP-ABE scheme, the most challenging issue is how to tie different secret keys together but still prevent the collusion attack. Similar to [3], in DAC-MACS, we separate the authority into a global certificate authority (CA) and multiple attribute authorities (AAs). The CA sets up the system and assigns a global user identity *uid* to each user and a global authority identity *aid* to each attribute

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

authority. The global unique *uid* can tie secret keys issued by different AAs together for decryption, and the global unique can distinguish attributes issued by different s. Thus, by using *uid* and *aid*, the collusion attack can be resisted. However, different from [3], the CA in the proposed scheme is *not* involved in any attribute management and the creation of secret keys reflecting the user's attributes. The proposed scheme also requires all the AAs to generate their own public keys which can be used to encrypt data together with the global public parameters, instead of only using the system unique public key for data encryption.

To achieve efficient decryption on the user, we propose a token-based decryption outsourcing method. We apply the decryption outsourcing idea from [4] and extend it to multiple authority systems by letting the CA generate a pair of global secret key and global public key for each legal user in the system. During the decryption, the user submits its secret keys issued by AAs to the server and asks the server to compute a decryption token for the ciphertext. The user can decrypt the ciphertext by using the decryption token together with its global secret key.

To solve the attribute revocation problem, we assign a version number for each attribute, such that for each attribute revocation, only those components associated with the revoked attribute in secret keys and ciphertexts need to be updated. When an attribute is revoked from a user, the corresponding AA will generate a new version key for this revoked attribute, and computes an update key containing a Ciphertext Update Key (CUK) and several user's Key Update Keys (KUKs). With the KUKs, each nonrevoked user can update its secret key to the current version, while the revoked user cannot update its secret key even using other users' update keys, since each is associated with the (Backward Security). The ciphertexts can also be updated to the current version with the, such that the newly joined users who have sufficient attribute are also able to decrypt the previous published data (Forward Security). Moreover, all the users only need to hold the latest secret key, rather than all the previous secret keys. To improve the efficiency, we delegate the workload of ciphertext update to the server by using the proxy re-encryption method.

B. System Initiation

The CA Accept both user registration and AA registration

a. *User Registration*

Every user should get registration from CA during the system initialization. The CA runs the user registration algorithm which takes the user information as inputs. If the user is legal in the system, it assigns a global user identity to this user, and generates the global public key and the global secret key to that particular user.

b. *AA Registration*

Each AA should also get registration from CA during the system initialization. The CA runs the registration algorithm by taking the information of as input. If the AA is a legal authority in the system, the CA assigns a global authority identity and its verification key to this particular AA.

c. Secret Key Generation by AAs

For every user, each Attribute authority first authenticates whether this user is a legal user by verifying its certificate by using the verification key. If the user is not legal, it aborts. Otherwise, the assigns a set of attributes to this user according to its role or identity in its administration domain. Then, the AAn runs the secret key generation algorithm to generate the user's secret key.

d. Data Encryption by Owners

Before outsourcing data into the cloud storage, the owner encrypts the data by running the data encryption algorithm. It takes a set of public keys from the involved authority set, a set of public attribute keys, the data and an access structure over all the selected attributes from the involved AAs. The algorithm first divides the data into several data components as according to the logic granularities. For example, the personal data may be divided into {name, address, security number, employer ID}. It then encrypts data components with different symmetric content keys by using symmetric encryption methods, where is used to encrypt.

e. Data Decryption by Users (With the Help of Cloud)

All the legal users in the system are able to decrypt ciphertexts from the cloud server. But only when the user's attributes satisfy the access structure embedded in the ciphertext, he/she is able to decrypt the content keys and further use them to decrypt the data. This phase consists of two steps: Token Generation by Cloud Server and Data Decryption by Users:

1) Token Generation by Cloud Server: The user sends its secret keys to the server and asks for a decryption token for the ciphertext. Only when the user's attributes satisfy the access structure defined in the ciphertext, then only the server can successfully compute the correct decryption token.

2) Data Decryption by Users: Upon receiving decryption token from cloud server the user can use it to decrypt the ciphertext together with its global secret key from CA.

f. Efficient improvement in Attribute Revocation

Suppose an attribute of the user is revoked from the AAn. The attribute revocation includes three phases: Update Key Generation by AAs, Secret Key Update by Nonrevoked Users and Ciphertext Update by Cloud Server. The secret key update can prevent the revoked user from decrypting the new ciphertexts which are encrypted by the new public attribute keys (Backward Security). The ciphertext update can also guarantee that the newly joined user who has sufficient attributes can still access the previous published data (Forward Security).

i. *Update Key Generation by AAs:* The corresponding authority AAn runs the update key generation algorithm to compute the update keys. The algorithm takes as inputs the secret authority key, the current attribute version key and the user's global public keys. It generates a new attribute version key. It first calculates

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

the Attribute Update Key, and then applies it to compute the user's Key Update Key and the Ciphertext Update Key.

ii. *Secret Key Update by Nonrevoked Users:* For each nonrevoked user who holds the revoked attribute, the AAn sends the corresponding user's key update key to it. Upon receiving Key Update Key, the user runs the key update algorithm to update its secret key.

iii. *Ciphertext Update by Cloud Server:* The AAn sends ciphertext update key to the server. Upon receiving the ciphertext update key, the server runs the ciphertext update algorithm to update all the ciphertexts which are associated with the revoked attribute.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Secure and efficient data access control scheme for multiauthority cloud storage systems. We also constructed a new multiauthority CP-ABE scheme, in which the main computation of decryption is outsourced to the server. We further designed an efficient attribute revocation method that can achieve both forward security and backward security.

Upon finishing our system design, we realized cryptographic techniques represent only one pillar in an overall approach to effective and efficient distributed data access control. Future work of this project might include:

1) Although this work is for multiauthority cloud storage systems, the techniques designed in this paper can be applied into other applications, such as any remote storage systems, online social networks etc.

2) Explore more dynamic and efficient way to illustrate access structures. Currently, we describe access structures in a static way, which is not suitable for a dynamic system. A better description method can further reduce static access policy complexity.

3) Combine other distributed data access control techniques with cryptographic techniques. Cryptographic techniques are an essential, but not the only one, method to protect data against partially trustworthy cloud server.

ACKNOWLEDGMENT

We are really thankful to the Almighty. We are also thankful HOD Prof. Shivakumar G.S, Dept. of computer science and Engineering SIT Mangalore. We also convey our thanks to my guide Asst.Prof .Mohan K and all the staff members of SIT for helping in the project.

REFERENCES

[1]. Kan Yang, Xiaohua Jia, Fellow, and KuiRen, *"DAC MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems,"* IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 11, NOVEMBER 2013

[2]. J. Bethencourt, A. Sahai, and B. Waters, *"Ciphertext-policy attribute based encryption,"* in f the 2007 IEEE Symposium on Security and Privacy (S&P'07). IEEE Computer Society, 2007, pp. 321–334.

[3]. M. Chase, "Multi-authority attribute based encryption," in Proc. TCC'07, 2007, pp. 515–534, Springer.

[4]. M. Green, S. Hohenberger, and B.Waters, *"Outsourcing the decryption of abe ciphertexts,"* in Proc. 20th USENIX Security Symp., 2011, pp. 1–16, USENIX Association.

[5]. A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT'11*, 2011, pp. 568–588, Springer.