An Access Control Scheme For Limiting The Number Of Users Based on Cloud Computing

Shi Guozhen[1],Ye Sishui[1], Zhu Yafei[1], Zhang Meng[1]
[1]Department of Information Security
Beijing Electronic Science and Technology Institute,
Beijing, China
sgz@besti.edu.cn, dachmx@163.com, 458512191@qq.com,
zhmeng0111@163.com

Wang Shuaibing[2]
[2]School of Computer Science and Technology
Xidian University
Xi'an, China
wangshuaibing2009@163.com

***Abstract*—Attribute-based access control scheme ensures that users have legitimate attribute to access the shared data. With attributes unauthorized, users can't access the data. But if we want a finer control to the data, some users, having attributes satisfied the access control structures within a certain number, can access the data and the others can't. A simple attribute-based access control method can't meet the above requirements. In this paper, based on the access control needs, we have designed an access control scheme, which can limit the number of users, who can access the shared data. It can limit the number of users effectively and data were shared security at the same time.**

Keywords—Cloud Computing; Quantitative restrictions; Attribute-Based Encryption; Data Sharing; Access control

I. INTRODUCTION

With the development of network technology and communication technology, cloud computing technology applications are increasingly widespread. Cloud computing system[1] is a distributed computing system with distributed storage. From view on its architectural model, there are some functions about data security and data management in the basis of management. While cloud computing technology solved storage requirements for big data, the security of data induces a great threat, mainly about data security. Once the data uploaded to the cloud, its owner will not be in full possession of the management of data.

Attributes-based encryption (ABE), evolving from Fuzzy identity-based encryption algorithm (Fuzzy IBE)[2], can solve the problem of fine-grained access control to data access, which is stored in cloud, which can't be carried out by other schemes. ABE can avoid the problem of complex key management, which happens on IBE and improve the flexibility for users. ABE enables the set of attributes and access control policies and data associated with these users. Only when satisfying the access control structure, the users can decrypt the ciphertext and obtain shared data. In addition, the owner of the data, can share the data with many users with the same attributes, such that ABE can improve the efficiency of the system. At the same time, the access control structure, used in ABE, improve access flexibility and efficiency. This ensures that the data, away from its owner, is in control and stored in the cloud server safely.

ABE generally contains two directions, Key-Policy Attribute-Based Encryption[2] (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption[3] (CP-ABE). KP-ABE only can choose descriptive attributes for encrypting the data, can't decide who can decrypt the encrypted data and you must believe the entity, who publishes the key used for encrypting data. However, CP-ABE lets attributes describe the users' private keys, and the owner of the data can determine who can access the encrypted data by access policies. Therefore, CP-ABE is a better choice in some applications. Currently ABE has been widely used in the access control, such as Zhou[4], Xu[5], Guo[6].

In many applications, the owner of the data requires those users, who share the data, with legitimate attributes and want that the number of users is kept within a certain range. For example, the leader of a department needs a few people finish a task in subordinate department. He uploaded to the company cloud server, but he don't want all stuff in the subordinate department to crowd for this thing, with saving resources for his company. So, how to get it? The leader doesn't know who is not busy and who is suited for the task this time, so he can't give a direct order. It's a good measure that setting a restriction for the number of users, who get the task. In addition, the cloud servers have access peak. So if they can meet the restrictions on the number of users, who already have met the access control structure, it will certainly reduce the network bandwidth pressure.

Limiting the number of users can not only save resources in many places, but also meet practical access control requirement in many specific application scenarios. This article designed a access control mechanism for shared data to meet the requirement. The paper introduces related research first. In Section 3 we propose the access control mechanism. In Section 4, we give security analysis for our mechanism. Finally, we conclude the paper in Section 5.

II. RELATED RESEARCH

A. The Decisional Bilinear Diffie-Hellman Assumption (DBDH)

The CP-ABE generally is designed based DBDH. Choose numbers $a,b,c,z \in Z_p$ randomly and a bilinear map $e:G \times G \rightarrow G_T$, which g is the generator of G. The DBDH assumption[7] is that

there is no probabilistic polynomial-time algorithm $\mathcal{B}$ can distinguish the tuple $(g^a,g^b,g^c,e(g,g)^{abc})$ from the tuple $(g^a,g^b,g^c,e(g,g)^z)$ with non-negligible advantage.

B. CP-ABE algorithm

CP-ABE determines the structure of the access control with attribute set first. Then each user's attribute is assigned with a non-empty set.

Access structure is attribute tree, consisted of a non-empty attribute set, which is used to describe an access control policy. Each leaf node of the tree represents an attribute item, representing the function of a relationship. The relationship can be represented by every internal node function. These functions maybe AND (n of n), OR (1of n) or n of m (m > n) threshold and so on. For example, the fig 1 is a access tree. When users need access to shared data, their attributes must meet the tree. When he is a professor of computer science and work in the Laboratory, he can access the shared data. CP-ABE can encrypt the shared data by the access tree. Generally, every node is mapped a polynomial. The constant of root node is used to encrypt the data, and other nodes in the tree can recover the polynomial of root node. The degree of each polynomial is the threshold corresponding node minus 1.

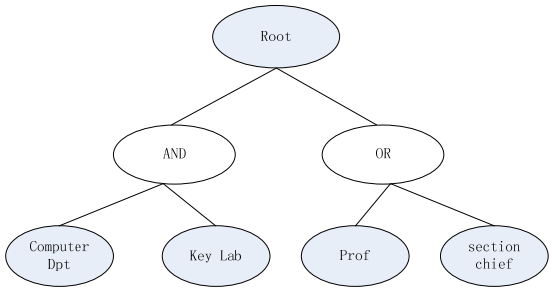

Figure 1. Access Tree

CP-ABE algorithm is consist of four steps mainly:

(1) $(MK,PK)=Setup(1^k)$: The function Setup receives the initialization parameter and output master key *MK* and public key *PK*.

(2) $CT=Encrypt(PK,M,A)$: This function encrypts plaintext *M* with the public key *PK* and access tree *A*, and output the ciphertext *CT*.

(3)$SK=KeyGen(MK,S)$: This function generates secret keys for every attributes in the set S with the master key *MK* and output the secret key *SK*.

(4) $M=Decrypt(PK,CT,SK)$: This function is the last step. Users decrypt the ciphertext *CT* with public key *PK* and attribute secret key *SK* and get the plaintext *M*.

III. Our Access Control Scheme

A. Requirement Analysis

There is a task in a company and the manager doesn't know that who is suitable for it. But he doesn't want to let all of the department complete the task, that is only want a certain number of employees finish that task .Now the better method is that he sends a notice to the department. There is a task needing to be done on the servers. Staff will go to the server for the task. When a certain number of employees have got the task, the others can't access it from the cloud server. There are a number of scenes like above. We summarized the following general requirements, based on the analysis.

(1) The shared data should exist or be transmitted as cipher text;

(2) The Scheme should meet the RBAC;

(3) The shared data can't be deleted from the cloud server after downloaded by a certain number of users for meet some requirements;

(4) The user, having accessed the shared data successfully, can access it again.

We design a mechanism mainly depending on CP-ABE, which can limit the number of users, who can access the shared data.

B. Access Control Architecture

The framework of our scheme is shown in figure 2. Content Provider, Receivers are users, Authority Server is authorization center, and all shared data is stored in the Cloud Servers. Content Provider is the shared data owner, namely cloud file is provided by him, and access tree is also set by him. Receiver as a user wants to access the shared data, if only if he has attributes, which match the access tree set for the shared data.

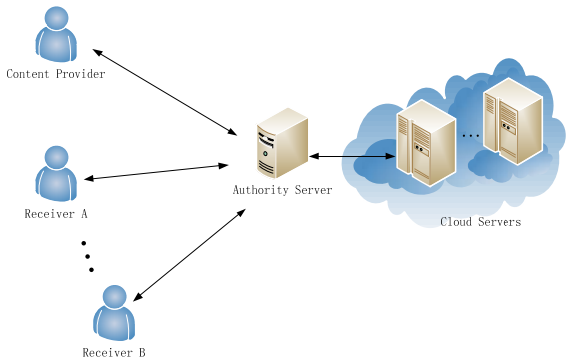

Figure 2. The implementation framework of our scheme

If beginning to work, the system must go through system initialization, when the keys such as public key *PK*, used by CP-ABE will be generated, the database used to recording will be initialized and other function will be completed. When the data owner wants to share data, he should determine a access control structure for the data, which is the access control policy. Then he could encrypt the data by CP-ABE and get the ciphertext. In addition, if wanting to upload data to Cloud Servers, he must go through the Authority Server. Authority server parses the ciphertext uploaded from Content Provider and creates record for the data, then delivering the ciphertext to Cloud Servers. If wanting to get the shared data, Receivers must be verified by the Authority Server first. If passing, Receivers can get the shared data by decrypting ciphertext download from Cloud Servers. The Authority Server verifies

the attributes of each Receiver and restricts the number of Receivers, who access to shared data.

C. *Specific access control scheme*

- System initialization

Authority Server consists of two parts, Authority Module (AM) and Private Key Generator Module (PKGM), shown as figure 3. AM is mainly used for verifying the attributes of Receivers and limiting the number of Receivers. PKGM is used for generating all encryption keys such as credential for every user, parameters for CP-ABE. If a user wants service from the system, he must apply a credential issued by Authority Server, which is used to represent its identity and create a safe channel for transferring data between them. Any users, wanting keys for decrypting or encrypting, must go through AM to protect PKGM.

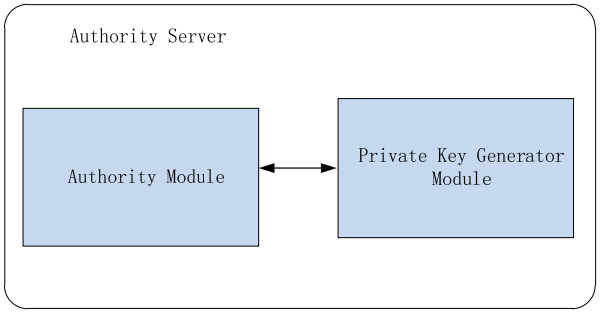

Figure 3. Authority Server Structure

- Sharing Data

When Content Provider wants to share some data to other users through the cloud platform. The process is shown in figure 4. He will go some steps as follows:

(1) Content Provider chooses data , being shared to others, and determines the number *k*, which limits the total of users, who can accessed the data, a flag *flag*, which is used to tell Authority Server whether the shared data can be destroyed after *k* users have accessed the data, and access tree, which is the access control policy.

(2) Content Provider requests parameters, including the master key *MK*, attribute set S from Authority Server.

(3) Authority server verifies the identity of Content Provider. If Content Provider is invalid, Authority Server creates a secure channel, and transfers information, requested by Content Provider.

(4) Content Provider encrypts data with parameters, getting from Authority Server, by CP-ABE, and get ciphertext *CT*. *CT* will be packaged with *k* and other data, and uploaded to Authority Server.

(5) Authority Server received data package from Content Provider and parse it. The ciphertext in the package will be delivered to Cloud Servers. The number *k* will be used to limit the total of users, who access the data. At the same time, *Record={k', UserR}* will be create to record the total of users, and the *UserR* is used to record public key hash value of each user, who has accessed the data..

(6) Cloud servers get ciphertext delivered by Authority Server. Ciphertext will be stored in the distributed storage system, and a success message will be sent to Authority server.

(7) Authority Server gives Content Provider a message based on the message from Cloud Servers.

If Content Provider receives a success message from Authority Server, the sharing process is completed, otherwise he must check and repeat sharing.

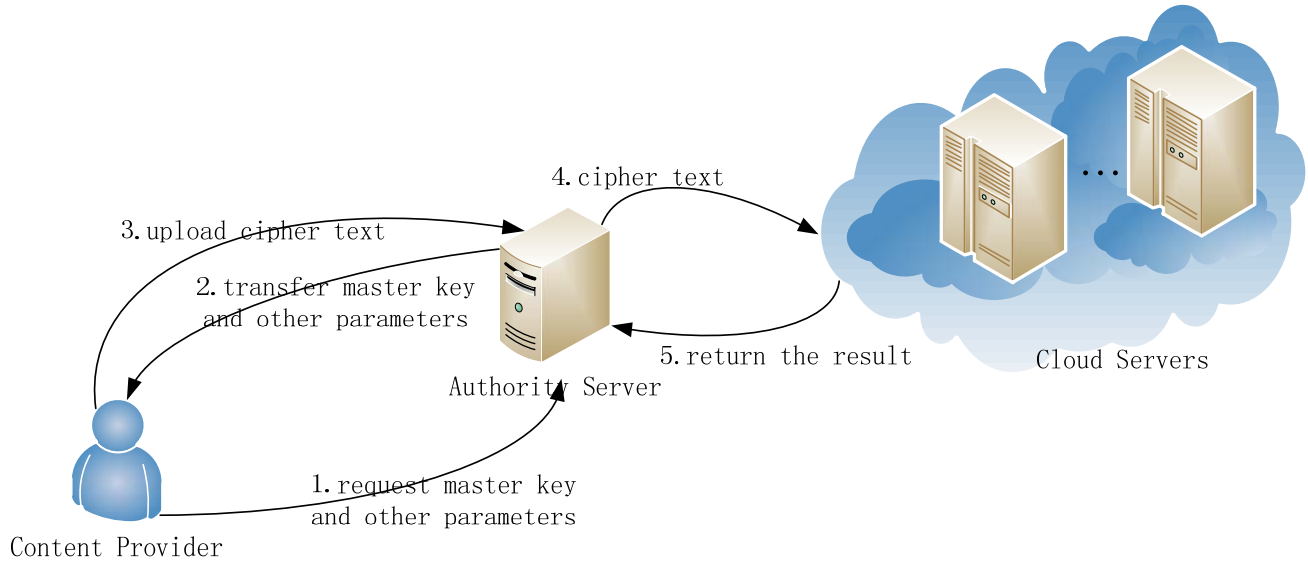

Figure 4. Sharing Data

- Data Access

When a Receiver wants to access shared data in cloud system, he will have to get through the following steps. The process is shown in figure 5.

(1) Receiver requests access to shared data from Authority Server;

(2) Authority Server verify the attributes of the Receiver. if the Receiver passed, the following situations will be checked:

1) If *flag=True*

a. If the user already exists in the Record, he can't access the data.

b. If *(k'+1)<k*, the Receiver is allowed to get the data. Let *k'=k'+1* and record the Receiver in *UserR* of *Record*;

c. If *(k'+1)=k*, do as step b, and remove the data from cloud after downloaded by the Receiver.;

d. If *(k'+1)>k*, the Receiver will be refused.

2) If *flag=False*

a. If the Receiver already exists in the record, he can access the data.

b. If *(k'+1)≤k*, the Receiver is allowed to get the data. Let *k^'=k'+1* and record the Receiver in *UserR* of *Record*;

c. If *(k'+1)>k*, the Receiver will be refused.

If being refused, the Receiver can't access the data. Otherwise, Authority Server transfer secret keys *SK* of attributes and other parameters to him.

(3) Authority Server request ciphertext from Cloud Servers.

(4) Cloud Servers transfer ciphertext to Authority Server.

(5) Receiver receives ciphertext from Authority Server and decrypts ciphertext with parameters by CP-ABE. Finally, the Receiver gets the plaintext, and this process is completed.

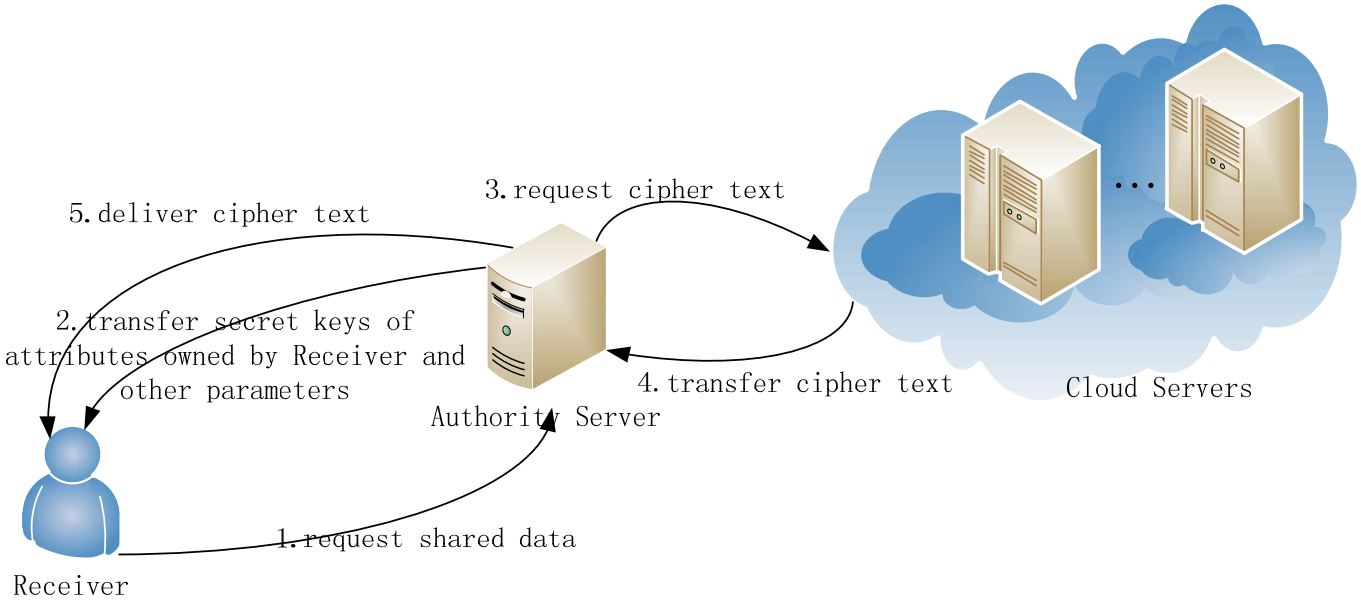

Figure 5. Data Access

IV. SECURITY ANALYSIS

The CP-ABE is generally based on the assumption DBDH[7], which has been proved secure. When attributes of user don't meet the requirements of the access tree, the user can't recover constant value of the root. In other word, he can't get the plaintext according to DBDH assume. In addition, RSA algorithm is used to authentication, and set up secure channels for communication.

V. CONCLUSIONS

Our scheme can be used in many application scenarios, which need restrictions on the number of users, and provide a reference for the implementation and designing of those applications. The introduction of CP-ABE can make the scheme more secure and flexible than the traditional RBAC scheme. Though our scheme meets some requirements happening on the cloud computing, there are still many problems not being considered and many exceptions not being processed. So it can't be used directly in practical applications, and its flexibility and availability are limited. we will continue to carry out more deep study.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61170251), the Digital Rights Management Technology Research and Development Projects (No.1681300000119), the Beijing Natural Science Foundation of China (No. 4152048).

REFERENCES

[1] Zheng W, Xu P, Huang X, et al. Design a cloud storage platform for pervasive computing environments[J]. Cluster Computing, 2010, 13(2): 141-151.

[2] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM conference on Computer and communications security. Acm, 2006: 89-98.

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007: 321-334.

[4] Zhou H, Wen Q. A new solution of data security accessing for Hadoop based on CP-ABE[C]//Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on. IEEE, 2014: 525-528.

[5] Xu R, Wang Y, Lang B. A Tree-Based CP-ABE Scheme with Hidden Policy Supporting Secure Data Sharing in Cloud Computing[C]//Advanced Cloud and Big Data (CBD), 2013 International Conference on. IEEE, 2013: 51-57.

[6] Guo F, Mu Y, Susilo W, et al. CP-ABE with constant-size keys for lightweight devices[J]. 2014.

[7] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]//Advances in Cryptology—CRYPTO 2001. Springer Berlin Heidelberg, 2001: 213-229.