Privacy Protection based Access Control Scheme in Cloud-based Services

Kai Fan, Qiong Tian, Nana Huang
State Key Laboratory of Integrated Service Networks
Xidian University
Xi’an, China
kfan@mail.xidian.edu.cn
tianqiongaa@163.com
1058219406@qq.com

Yue Wang
School of Information Engineering
Xi’an University
Xi’an, China
kelly8266no1@sina.com

Hui Li
State Key Laboratory of Integrated Service Networks
Xidian University
Xi’an, China
lihui@mail.xidian.edu.cn

Yintang Yang
Key Lab. of the Minist. of Educ. for Wide Band-Gap Semiconductor Materials and Devices
Xidian University
Xi’an, China
ytyang@xidian.edu.cn

Abstract—**With the rapid development of the computer technology, cloud-based services have become a hot topic. Cloud-based services not only provide users with convenience, but also bring many security issues. Therefore, the study of access control scheme to protect users' privacy in cloud environment is of great significance. In this paper, we present an access control system with privilege separation based on privacy protection (PS-ACS). In the PS-ACS scheme, we divide the users into personal domain (PSD) and public domain (PUD) logically. In the PSD, we set read and write access permissions for users respectively. The Key-Aggregate Encryption (KAE) is exploited to implement the read access permission which improves the access efficiency. A high degree of patient privacy is guaranteed simultaneously by exploiting an Improved Attribute-based Signature (IABS) which can determine the users’ write access. For the users of PUD, a hierarchical attribute-based encryption (HABE) is applied to avoid the issues of single point of failure and complicated key distribution. Function and performance testing result shows that the PS-ACS scheme can achieve privacy protection in cloud-based services.**

Keywords- access control; data sharing; privacy protection; cloud-based services

I. INTRODUCTION

With the rapid development of cloud computing, big data and public cloud services have been widely used. The user can store his data in the cloud service. Although cloud computing brings great convenience to enterprises and users, the cloud computing security has always been a major hazard. For users, it is necessary to take full advantage of cloud storage service, and also to ensure data privacy. Therefore, we need to develop an effective access control solution. Since the traditional access control strategy [1] cannot effectively solve the security problems that exist in data sharing. Data security issues brought by data sharing have seriously hindered the development of cloud computing, various solutions to achieve encryption and decryption of data sharing have been proposed. In 2007, Bethencourt et al. [2] first proposed the ciphertext policy attribute-based encryption (CP-ABE). However, this scheme does not consider the revocation of access permissions. In 2011, Hur et al. [3] put forward a fine-grained revocation scheme but it can easily cause key escrow issue. Lewko et al. [4] used multi authority ABE (MA-ABE) to solve key escrow issue. But the access policy is not flexible. Li et al [5] presented data sharing scheme based on systemic attribute encryption, which endows different users’ different access rights. But it is not efficient from the complexity and efficiency. In 2014, Chen et al. [6] proposed Key-Aggregate Encryption algorithm, effectively shortening the length of the ciphertext and the key, but only for the situation where the data owner knows the user's identity. These schemes above only focus on one aspect of the research, and do not have a strict uniform standards either. In this paper, we present a more systematic, flexible and efficient access control scheme. To this end, we make the following main contributions:

1. We propose a novel access control system called PS-ACS, which is privilege separation based on privacy protection. The system uses Key-Aggregate Encryption (KAE) scheme and Hierarchy Attribute-based Encryption (HABE) scheme to implement read access control scheme in the PSD and PUD respectively. The KAE scheme greatly improves access efficiency and the HABE scheme largely reduces the task of a single authority and protects the privacy of user data.

2. Compared with the MAH-ABE scheme which does not refer to the write access control, we exploit an Improved Attribute-based Signature (IABS) [7-9] scheme to enforce write access control in the PSD. In this way, the user can pass the cloud server’s signature verification without disclosing the identity, and successfully modify the file.

3. We provide a thorough analysis of security and complexity of our proposed PS-ACS scheme. The functionality and simulation results provide data security in acceptable performance impact, and prove the feasibility of the scheme.

II. SYSTEM MODEL

As shown in Fig.1, our system model consists of Data owner, users in PSD, and users in PUD, root authority CA, regional authority AA and cloud service provider, which are defined as follows.

1. The cloud service provider consists of two parts: data storage server and data service management. Data storage

server is responsible for storing confidential data files, and data service management is in charge of controlling external users' access to secret data and returning the corresponding ciphertext.

2. In the actual cloud environment, CA manages multiple AA, and AA each manages attributes in their own field. The attributes owned by the user are issued by different authority.

3. Personal domain (PSD), in which users have special privileges, such as family, personal assistant, close friends and partners. This domain has a small number of users and small scale attributes, and the data owner knows the user's identity, which is easy to manage.

4. Public domain (PUD), which owns a huge number of users with unknown identity and a lot of attributes owned by the user.

5. Data Owner, based on the characteristics of users in public and personal domain to develop different access control strategy, encrypt uploaded files using the corresponding encryption method and then send to the cloud server.

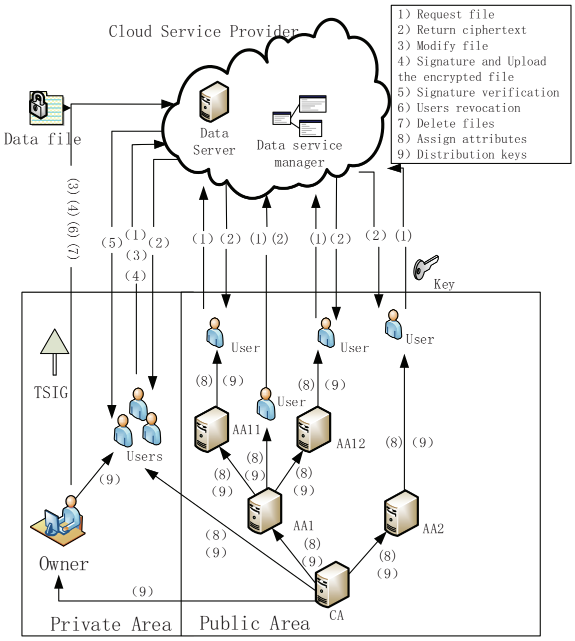

Figure 1. System framework

III. ACCESS CONTROL SCHEME IN PSD

A. *Read Access Control*

The PSD has a small number of users, and their identities are known to the owner. In general, the data owner only wants the users to access or modify parts of data files, and different users can access and modify different parts of the data. For example, the blogger can allow his friend to browse part of his private photos; enterprises can also authorize employees to access or modify part of sensitive data. This requires the data owner to grant users read or write access permission to some data. In Chen's MAH-ABE scheme, the CP-ABE is used to achieve the read access permission, but there are some defects to be considered. Firstly, since in the PSD, the users are all have a close relationship with the owner and the number is small, there is no need to use the CP-ABE which is applicable to the scenario which has a lot of users, and their identities are unknown to the owner, while the KAE scheme is set for the small users with certain identities. Besides, the distribution and management of keys and attributes, encryption and decryption process of CP-ABE are much more complex compared with the KAE scheme. Therefore, the KAE is exploited to implement the read access permission which improves the access efficiency.

Based on the above analysis, the paper uses the Aggregate Key Encryption scheme to encrypt the data files to realize different read access control. The specific application process of the KAE algorithm is as follows.

1. System setup and file encryption. The system first runs *Setup* of KAE to establish the public system parameter and master key. Each owner classified the file by its data attribute, such as "photo files", "blog files" and "game files". Fig.2 shows the way to classify the files. Choose and label the files, denoted by $i(i \in \{1,2,..., n\})$, note that a file class i cannot be the subset of another file class $j(j \in \{1,2,..., n\})$. Then the owner's client application runs Encrypt of KAE using the public key and the number of classification file to encrypt the PHR files and sends them to the cloud.

2. Access and key distribution. When the user send access request to the cloud server, and his file index number is i, then the cloud server returns the corresponding encrypted classification file to the user. The owner authorized users access permission with the file index number denoted by j and sent the collection S of all the index number j to CA, CA generate an aggregate decryption key for a set of ciphertext classes via *Extract* of KAE and sent it to the corresponding user, Finally, any user with an aggregate key can decrypt any ciphertext whose class is contained in the aggregate key via *Decrypt* of KAE.

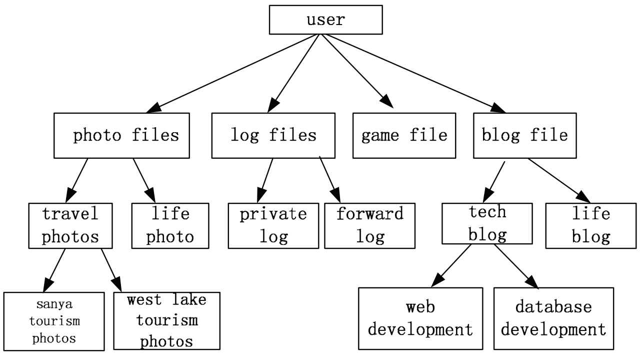

Figure 2. Data file classification

B. *Write Access Control*

As Chen's MAH-ABE scheme does not refer to the write access control, and in the PSD some cases exist, for example, the owner needs his friends to modify his file after he read it. So we proposed the write access permission in the PSD. For the user, the public key and file class label are all known, he can implement the algorithm to encrypt the files after he modified, and then upload them to the cloud. But whether the cloud server saves the modified file is decided by the write access control policy. On the one hand, in the complex cloud environment, if a user's modification operations are very frequent, maybe he is very important to the user, so that the user may be stricken from outside attacks. Therefore, the user worries the leak of identity after the signature. On the other hand, in the data sharing scheme, the separate access of read and write to the file is extremely important. In PSD, not all users who have read permissions also have write permissions to the files. Whether the user has write permissions to the file is decided by the data owner. Therefore, this paper selects the improved attribute-based signature (IABS) to determine the user's write permission.

The main structure of the scheme includes five parts: an authentication center (CA), the data owner, users, mediator and cloud servers. The CA is responsible for generating master key which is sent to the owner and system parameters which are shared for all users. The mediator holds part components of the signature keys and is responsible for the validity check of attributes and users. The data owner produces the signature tree and sends it directly to the cloud server. The user encrypts the modified files and signs them using the attribute-based signature, then uploads them to the cloud server. The cloud server verifies the attribute-based signature, if the authentication is successful, the user has permission to modify files and the cloud server stores the file. Own to the limited space we will omit the specific description of the IABS scheme in PSD.

IV. ACCESS CONTROL SCHEME IN PUD

Before introducing our proposed secure authentication protocol, we first make a statement for the notations used in the later, all of them are listed in Table I.

Notation	Description
PUD	Public Domain
PRD	Private Domain
CP-ABE	Ciphertext-policy Attribute-Based Encryption
MA-ABE	Multi-authority Attribute-based Encryption
HABE	Hierarchical Attribute Encryption
CK	Encryption Key
K	Key Space
PK	Public Key
SK	Secret Key
KAE	Key-Aggregate Encryption
CA	Authorization Center

Table I

A. Scheme Design

The PUD is characterized by a huge number of users, a lot of attributes owned by the user, complexity management, and indefinite users' identity. In view of the above characteristics, the user can only have the read access permission. Although the attribute-based encryption scheme (CP-ABE) can achieve access control, it cannot meet the needs of complex cloud environment. In traditional CP-ABE scheme, there is only one authorized agency responsible for the management of attributes and distribution of keys. The authority may be a university registrar's office, the company's HR department or government educational organizations and so on. The data owner defines access policies and encrypts the data files in accordance with this policy. Each user is distributed a key related to his attribute. As long as the user's attributes meet the access policy he can decrypt the file. However, if there is only one authority in the system and all public and private keys are issued by the authority. Two problems will appear in the practical application:

1. In the practical cloud environment, there are a lot of authorities and each authority in their own field manages part of users' attributes. The attributes owned by the user are issued from different authorities. For example, a data owner may want to share his medical data with a user who owns the doctor attribute issued by medical institutions and the medical researcher attribute by the clinic practice management. Therefore, exploiting multi authority is more realistic in the practical scenarios.

2. If there is only one authority, all the distribution of the keys are handed over by one trusted authority. The frequent interaction between the user and trust authority will not only bring bottlenecks for the system load capacity, but also increase the potential security risks. Therefore, multi authority ABE (MA-ABE) is used in this paper.

Users in PUD do not need to interact directly with the data owner, and the attributes of the user are called role attributes. Firstly the data owner uploads the attribute-based encrypted data files to the cloud server. Then after authorized, the data owner receives the corresponding decryption key and sends a data file access request directly from the cloud server. Finally, after the cloud server returns the ciphertext, users can use their own decryption key to decrypt the ciphertext. The framework of this area is shown in Fig.3.

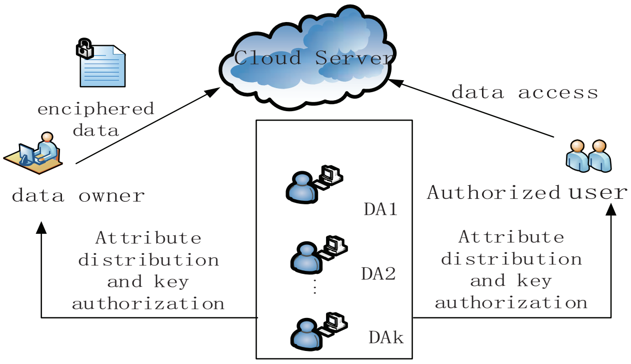
Cloud Server
enciphered data
data access
data owner
DA1
DA2
DAk
Authorized user
Attribute distribution and key authorization
Attribute distribution and key authorization

Figure 3. Access control framework of PUD

B. Access Control Process

Based on the above analysis, we use a hierarchical attribute encryption scheme (HABE) to implement access control in PUD.

1. Files creation: The creating of files is completed by the data owner. In general, in order to protect the privacy of the data file, the data owner firstly encrypts data file, and then stores it in the cloud. To reduce the ciphertext size and complexity, the data owner combines the symmetric encryption scheme with public key encryption scheme, namely that each file is firstly encrypted with symmetric encryption key called CK, then CK is encrypted with the HABE program. Before the data file uploaded to the cloud, the process of creating a data file is as follows:

1) Select a unique *ID* for the data file.

2) Choose a random symmetric encryption key $CK \xleftarrow{R} K$. K means key space, and encrypt the data file with *CK*.

3) Define access tree T, use the algorithm $HABE.Encrypt(PK_e, CK, T)$ to encrypt *CK* and return the *CT*.

4) The data owner computes the *CT* by hash operations and signs $h(CT)$ to get the signature SG , on the one hand to ensure the integrity of the data, on the other hand to facilitate the cloud and user to authenticate the identity of the data owner.

2. Data access: If the user wants to access a data file, he should get the file from the cloud server and decrypt the encrypted data file, which corresponds to the decryption process. There are two stages: firstly use the algorithm $HABE.Encrypt(PK_e, CK, T)$ to decrypt the symmetric encryption key *CK*, then use the key *CK* to decrypt the data file.

3. Files deletion: If the data owner wants to delete a file, he can send the file *ID* and his signature SG to the cloud server, then the cloud servers delete the files after verifying the signature of the data owner.

4. Attribute revocation: The authority assigns attributes to each user and attaches the set of attributes with an expiration time T . The attributes of access control tree contain a time attribute T', if $T > T'$ and the attributes match, then this file can be access to. So the data owner can restrict users' access permissions by changing the time attributes.

5. Users' attributes Revocation: The *DA* calculates the minimum set of attributes $A_{\min}$ that allows users' access revocation, and $A_{new} = A - A_{\min}$, making $T(A_{\min})$ returns null. Set a new expiration time to each attribute set, generate new private key components and return it to the client.

V. SYSTEM SIMULATION AND PERFORMANCE ANALYSIS

A. Security Analysis

In PSD, the user can only decrypt the files corresponding to the received aggregate keys and do not have access to other files, so that the data owner controls the users' access permissions. When the data file is modified, although CA is trusted, also the system parameters and revocation instructions are generated by the CA. The signature policy is formulated by the data owner and sent directly to the cloud server. The CA does not know the signature policy. Assuming that CA cannot give itself authorization, as long as the attributes of CA cannot meet the access policy, it is not valid to modify the file. Thus, the write access permissions still belong to the data owner. In the process of the users' signature, the signature key is only related to the users' attributes, so the user's identity is safe. On the whole, the IABS scheme can protect users' identity privacy.

In PUD, this paper employs the HABE scheme for the large number of users with uncertain identity in this region. For the trusted CA, it can only issue the private key and the corresponding attribute structure to the authority in the first level not to the users, so that the CA does not directly control the user's private key, thus reducing the trust in CA. In addition, the user's private keys are managed by multiple authorized agencies, which can avoid users' privacy leakage.

B. Simulation Analysis

In our KAE scheme in the PSD, the system parameters are generated by the trusted authority, which is not within our consideration. Furthermore, the $\hat{e}(g_1, g_n)$ can be calculated in the system setup. In addition, the aggregate key only needs one pairing operation, and to calculate a pairing operation is very fast, the specific comparison can be seen in Fig.4.

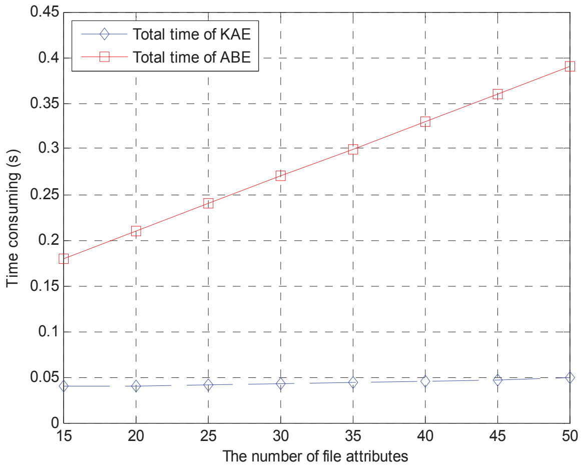

Figure 4. Total time of KAE and ABE

In Fig.4, the attribute-based encryption algorithm of the MAH-ABE scheme spent much more time than the KAE algorithm used in our scheme. If the attribute revocation occurs, the ABE algorithm will be more time-consuming. More importantly, the growth rate of time spent with the number of file attributes is much higher than KAE algorithm. The simulation results show the high efficiency of our scheme.

In Fig.5, the user only needs a very short time to sign the modified files. While, the authentication time only makes up a small part, so the process of signature and authentication consume a very small time. Therefore, from the client's perspective, the program is efficient.

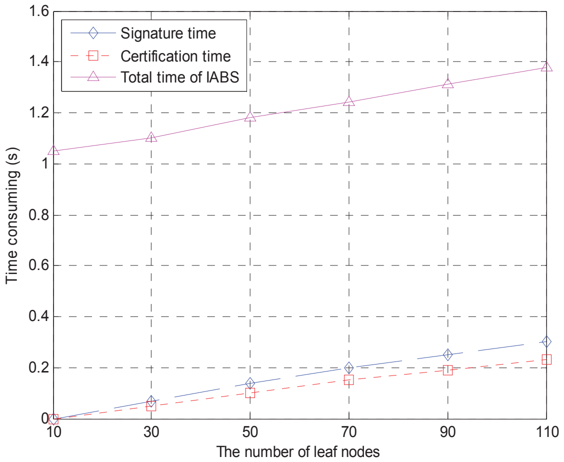

Figure 5. The signature and authenticationtime of IABS

VI. CONCLUSIONS

In this paper, we propose access control system (PS-ACS), which is privilege separation based on privacy protection. Through the analysis of cloud environment and the characteristics of the user, we divide the users into personal domain (PSD) and public domain(PUD) logically. In the PSD, the KAE algorithm is applied to implement users read access permissions and greatly improved efficiency. The IABS scheme is employed to achieve the write permissions and the separation of read and write permissions to protect the privacy of the user's identity. In the PUD, we use the HABE scheme to avoid the issues of single point of failure and to achieve data sharing. Furthermore, the paper analyzes the scheme from security and efficiency, and the simulation results are given. By comparing with the MAH-ABE scheme, the proposed scheme shows the feasibility and superiority to protect the privacy of data in cloud-based services.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China (No. 61303216, No. 61272457, No. U1401251, and No. 61373172), the National High Technology Research and Development Program of China (863 Program) (No. 2012AA013102), the China Postdoctoral Science Foundation funded project (No.2013M542328), and National 111 Program of China B16037 and B08038.

REFERENCES

[1] S. Yu, C. Wang, K. Ren, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” Proc. IEEE INFOCOM, pp. 1-9, 2010.

[2] J. Bethencourt, A. Sahai, B. Waters, “Ciphertext-policy attribute-based encryption,” Proc. Security and Privacy, pp. 321-334, 2007.

[3] J. Hur, D.K. Noh, “Attribute-based access control with efficient revocation in data outsourcing systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7 pp. 1214-1221, 2011.

[4] A. Lewko, B. Waters, “Decentralizing attribute-Based encryption,” Proc. Advances in Cryptology-EUROCRYPT, pp. 568-588, 2011.

[5] M. Li, S. Yu, Y. Zheng, “Scalable and secure sharing of personal health records in cloud computing using attribute-Based Encryption,” IEEE Transactions on Parallel and Distributed System, vol. 24, no. 1, pp. 131-143, 2013.

[6] C.K. Chu, S.S.M. Chow, W.G. Tzeng, “Key-aggregate cryptosystem for scalable data sharing in cloud storage,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp.468-477, 2014.

[7] J. Li, K. Kim, “Hidden attribute-based signatures without anonymity revocation,” Information Sciences, vol. 180, no. 9, pp. 1681-1689, 2010.

[8] H.K. Maji, M. Prabhakaran, M. Rosulek, “Attribute-Based Signatures,” Proc. Topics in Cryptology - CT-RSA, pp. 376-392, 2011.

[9] S. Kumar, S. Agrawal, S. Balaraman, “Attribute based signatures for bounded multi-level threshold circuits,” Proc. Public Key Infrastructures, Services and Applications, pp. 141-154, 2011.