International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

ENHANCED SECURE ACCESS CONTROL FOR CLOUD DATA SERVICES USING CP-ABE AND KEY-AGGREGATE ENCRYPTION

[1]Dr. R.V. KRISHNA REDDY, [2]Mr. M. VENKATESAM, [3]Mr. A.V. PHANI KUMAR, [4]Mr. C. VENKATESWARLU
[1]Professor, Krishna Chaitanya Degree College, Nellore, AP, India.
[2, 3, 4]Assoc. Professor, Krishna Chaitanya Degree College, Nellore, AP, India.

***Abstract* – In this study, we look at cloud-based services as an emerging field. Cloud services are highly convenient for customers, but they also come with security risks, such privacy and data sharing. In this study, we investigate a privacy-based approach to access control. In this architecture, we logically separate the cloud users into public and private zones. In private spaces, read and write access is secured using both a Key Aggregation cryptosystem and a signature-based access control mechanism. In order to overcome the challenges of challenging key distribution and single point of failure, we design and execute a new CP-ABE scheme in public space termed attribute authority based CP-ABE scheme with well-organized decryption. For it, we also create an efficient attribute revocation method.**

***Keywords:* Cloud computing, Key aggregation crypto system, Access control, Attribute based signature scheme, revocation method.**

1. INTRODUCTION

Big data and public cloud services have been widely utilized as cloud computing has grown rapidly. The cloud service allows the user to collect his data. Cloud computing security has always been a significant risk, despite the fact that it greatly expedites operations for both individuals and businesses. Users must ensure data privacy and utilize cloud storage services to their fullest potential. As a result, we must create an effective access control system. Because data sharing security issues cannot be adequately resolved by the traditional access control method [1]. The growth of cloud computing has been severely hampered by data security difficulties brought on by data sharing; numerous solutions have been put forth to accomplish encryption and decryption of data sharing.

Bethencourt et al. [2] introduced ciphertext policy attribute-based encryption (CP-ABE) in 2007. However, the revocation of access permissions is not taken into account by this scheme. Hur et al. [3] proposed a fine-grained revocation approach in 2011, however it can easily lead to important escrow issues. Multi authority ABE (MA-ABE) was utilized by Lewko et al. [4] to address a significant escrow issue. However, the policy for access is not light. A systematic attribute encryption-based data sharing system that grants varying access privileges to users was introduced by Li et al. [5]. However, it lacks efficiency due to its difficulties.

Chen et al. [6] introduced the Key-Aggregate Encryption technique in 2014, which successfully reduces the length of both the key and the ciphertext—but only in cases where the data owner is aware of the user's identity. The aforementioned systems only focus on a single aspect of the research and lack rigorous, consistent standards. We offer a more methodical, adaptable, and effective access control mechanism in this study. In light of this, we primarily contribute the following:

- We proposed PSACS, a unique access control system that separates privileges based on privacy protection. The system employs read access control schemes in the PSD and PUD using the Key-Aggregate Encryption (KAE) scheme and the Hierarchy Attribute-based Encryption (HABE) scheme, respectively. The HABE method protects user data privacy and substantially impedes the work of a single authority, whereas the KAE scheme greatly increases access efficiency.
- To implement write access control in the PSD, we create an improved Attribute-based Signature (IABS) [7-9] scheme in contrast to the MAH-ABE method, which does not dedicate to the write access control. By doing this, the user can successfully change the file and pass the cloud server's signature verification without revealing their identity.
- We offer a comprehensive examination of the security and complexity of our suggested PS-ACS method. The functionality and simulation results demonstrate the scheme's viability and guarantee data security with appropriate performance effect.

2. RESEARCH METHOD

2.1 System Framework

Our system model includes the following: root authority CA, regional authority AA, cloud service provider, data owner, users in PSD, and users in PUD, as shown in Fig. 2.1.

a) The data storage server and data service management are the two components that make up the cloud service provider. Confidential data files must be stored on a data storage server, and data service administration is

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

in charge of restricting access to secret data by outside parties and providing the associated ciphertext.

b) CA oversees several AA in the real cloud environment, while AA oversees attributes in their respective domains. The user's qualities are granted by various authorities.

c) Personal domain (PSD), where users have access to special privileges like close friends, family, personal assistants, and lovers. This domain is easy to maintain because it has few users and small-scale properties, and the data owner is aware of each user's identity.

d) The public domain (PUD), which has a vast user base with an undefined identity and several user-owned attributes.

e) The data owner encrypts uploaded files using the appropriate encryption method and sends them to the cloud server depending on user characteristics in the public and private domains to extend various access control strategies.

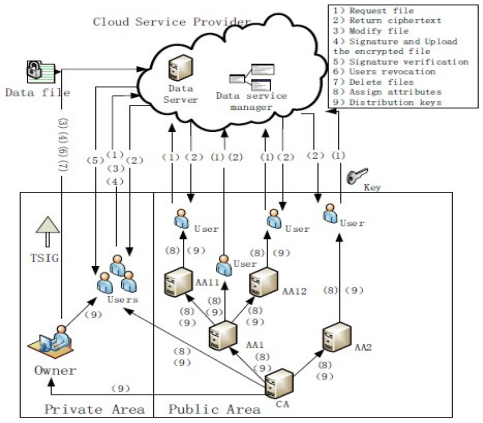

Fig 2.1: System Framework

2.2 ACCESS CONTROL SCHEME IN PSD

2.2.1 Read Access Control

There are very few users on the PSD, and the owner is aware of who they are. Generally speaking, different users can access and alter different parts of the data, and the data owner only needs users to access or change specific parts of the data files. For instance, the blogger may assign a friend to view a portion of his private images; businesses may likewise let staff members to view or edit a portion of sensitive information. This necessitates that the data owner provide users the ability to read or write certain data. Although there are certain difficulties to take into account, Chen's MAH-ABE technique uses the CP-ABE to obtain the read access permission. First off, the PSD has a small number of users who are all in close contact with the owner, so there's no need to use the CP-ABE, which is appropriate for situations where there are many users and the owner doesn't know who they are. In contrast, the KAE scheme is designed for small users who have clear identities.

In addition, CP-ABE's encryption and decryption procedures, as well as the distribution and administration of keys and characteristics, are far more complex than those of the KAE scheme. In order to achieve read access permission, which increases access effectiveness, the KAE is thus abused.

In order to understand various read access controls, the study employs the Aggregate Key Encryption technique to encrypt the data files. The KAE algorithm's particular application procedure is as follows.

1. File encryption and system configuration. To generate the master key and public system parameter, the system first runs Setup of KAE. Each owner groups the file according to its data attribute, such as "game files," "blog files," and "photo files." The files can be categorized as shown in Fig. 3.2. Select the files and give them labels. After that, the owner's client request encrypts the PHR files and sends them to the cloud by running Encrypt of KAE using the public key and the number of categorization files.

2. Distribution of keys and access. Upon receiving an access request from the user with a file index number of I, the cloud server provides the user with the encrypted categorization file. With the file index number j, the owner granted users access rights. The owner then transmitted the collection S of all the index numbers j to the CA, who used the Extract of KAE to generate an aggregate decryption key for a set of ciphertext classes and sent it to the appropriate user. Lastly, any user having an aggregate key can utilize Decrypt of KAE to decrypt any ciphertext whose class is contained in the aggregate key.

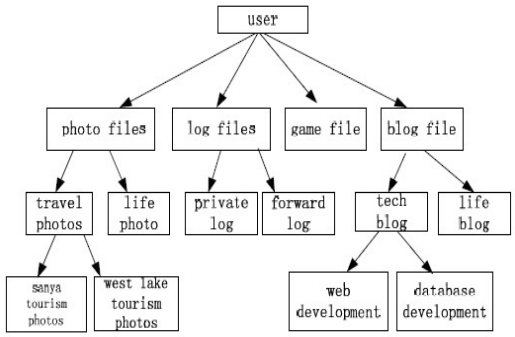

Fig 2.2: Data Files Classification

2.2.2 Write Access Control

Since Chen's MAH-ABE scheme does not grant write access control, there are instances in the PSD where the owner requires his friends to make changes to his file after he has read it. In the PSD, we therefore projected the write access permission. After modifying the files, the user can realize the

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

encryption algorithm and upload them to the cloud because he knows the public key and the file class label. However, the write access control policy determines whether the cloud server keeps the altered file. On the one hand, if a user makes changes often in the composite cloud environment, Perhaps he plays a vital role in protecting the user from external threats.

The user is therefore skeptical of the identity leak following the signature. On the other hand, independent read and write access to the file is crucial in the data sharing strategy. Not every user with read permissions in PSD can also write to the files.

The data owner decides whether the user has write permissions to the file. Therefore, the enhanced attribute-based signature (IABS) is used in this work to ascertain the user's write permission.

An authentication center (CA), the data owner, users, the mediator, and cloud servers make up the scheme's five essential components. The CA is in charge of producing the system parameters that are shared by all users and the master key that is supplied to the owner. Part of the signature keys are held by the mediator, who is also responsible for the validity test of users and attributes. The signature tree is generated by the data owner and sent straight to the cloud server. The user uploads the altered files to the cloud server after encrypting them and signing them with an attribute-based signature. After the attribute-based signature is verified by the cloud server, the user is granted authorization to edit files, and the file is stored on the cloud server. We will not include a detailed discussion of the IABS system in PSD because to space constraints.

2.3 ACCESS CONTROL SCHEME IN PUD

2.3.1 Scheme Design

The PUD is distinguished by its vast user base, numerous user-owned properties, complexity management, and ambiguous user identities. The user can only be granted read access due to the aforementioned features.

Despite its ability to achieve access control, the attribute-based encryption technique (CP-ABE) is unable to meet the demands of a complex cloud environment. The distribution of keys and administration of characteristics are under the purview of a single authorized agency in a traditional CP-ABE scheme. The authority could be the HR division of a business, the registrar's office at a university, government educational institutions, etc. Data files are encrypted in compliance with access policies that are described by the data owner. A key associated with his attribute is given to each user.

The user can decrypt the file if his attributes match the access policy. However, if the system has a single authority and that authority issues all public and private keys. In the real-world application, two issues will arise:

- There are a number of authorities in the real-world cloud environment, and each authority oversees a portion of the user attributes in their respective fields. The characteristics that the user possesses originate from several authorities. For instance, a data owner might wish to share his medical records with a user who possesses the medical researcher attribute from the clinic practice management and the doctor attribute from medical institutions. As a result, in more realistic situations, multi-authority development makes more sense.
- In the event that there is only one authority, one reliable authority distributes all of the keys.

Regular communication between the user and the trust authority will increase the potential security threats in addition to obstructing the system's load capacity. Consequently, this study uses multi authority ABE (MA-ABE).

In PUD, users' traits are referred to as role attributes, and they are not required to interact directly with the data owner. The attribute-based encrypted data files are first uploaded to the cloud server by the data owner. Following authorization, the data owner gets the appropriate decryption key and sends a request for data file access straight from the cloud server. Users can then decrypt the ciphertext using their own decryption key once the cloud server has returned it. Fig. 3.3 depicts the area's framework.

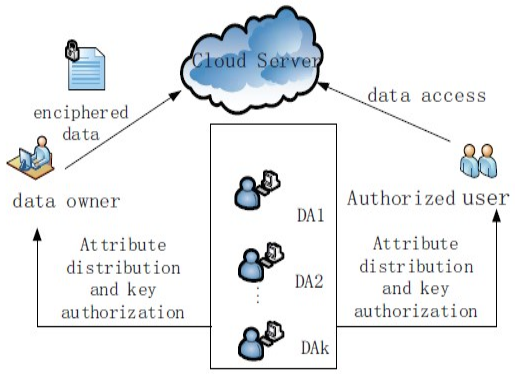

Fig 2.3: Access control framework of PUD

2.3.2 Access Control Process

We implement access control in PUD using a hierarchical attribute encryption method (HABE) in accordance with the aforementioned analysis.

i. **Creation of files:** The data owner stops files from being created. Generally speaking, the data owner encrypts the data file before storing it on the cloud to protect its privacy. The data owner combines the symmetric encryption technique with the public key

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

encryption strategy in order to reduce the size and complexity of the ciphertext. Specifically, each file is first encrypted using a symmetric encryption key called CK, which is subsequently encrypted using the HABE program. The following steps are involved in producing a data file before it is posted to the cloud:

- Give the data file a distinctive ID.
- Select a symmetric encryption key at random. K stands for key space, and CK is used to encrypt the data file.
- Define access tree T, then encrypt CK and return the CT using the technique H A B E.E n c r y p t (P K e, C K, T).
- To ensure the integrity of the data and to let the cloud and user verify the data owner's identity, the data owner processes the CT using hash operations and signs h(CT) to obtain the signature SG.

ii. **Data access:** The user should obtain the encrypted data file from the cloud server and decode it if he needs to access it. This is connected to the decryption procedure. There are two steps: first, the symmetric encryption key CK is decrypted using the algorithm HABE Encrypt (PKe, CK, T); next, the data file is decrypted using the key CK.

iii. **File deletion:** The data owner can provide the cloud server the file ID and his signature SG if he wishes to remove a file. The cloud server will then remove the files after confirming the data owner's signature.

iv. **Attribute revocation:** Each user is given a set of characteristics by the authority, which also gives an expiration time T to the set. A time attribute T' is enclosed in the attributes of the access control tree; if T > T' and the attributes match, this file can be accessed. Therefore, by changing the time attributes, the data owner can restrict users' access capabilities.

v. Features of users Revocation: T(Amin) yields nil when the DA determines the minimal set of characteristics Amin that let users to access revocation, and Anew = A-Amin new. Give each attribute set a new expiration date, generate new private key components, and send them back to the client.

3. RESULTS AND ANALYSIS

3.1 Security Analysis

To allow the data owner to control user access permissions, the PSD only allows the user to restore the files that correspond to the obtained aggregate keys. Even though the CA is trusted, when the data file is modified, the CA also creates the system parameters and voidance of instructions. The data owner creates the signature policy and sends it straight to the cloud server. The signature policy is unknown to the CA. Assuming that the CA is unable to provide itself authorization, it is not appropriate to change the file as long as the CA's characteristics cannot guarantee the access policy. As a result, the data owner still has write access rights. The user's identity is protected during the signing process since the signature key is only connected to the user's attributes. Overall, the IABS method can safeguard users' privacy and identification.

This work implements the HABE system in PUD for the vast number of users in this area who have a provisional identity. The trustworthy CA can only provide the private key and associated attribute structure to the first-level authority, not to users. This prevents the CA from having direct control over the user's private key, which erodes CA credibility.

Additionally, several approved agencies produce the user's private keys, preventing privacy leaks.

3.2 Simulation Analysis

The confidential authority creates the system specifications for our KAE plan in the PSD, which is outside of our purview. Furthermore, the system design allows for the calculation of the eˆ(g_1,g_n). Furthermore, the aggregate key only requires a single pairing operation, and the computation of a pairing operation is relatively quick. Fig. 3.1 shows the specific comparison.

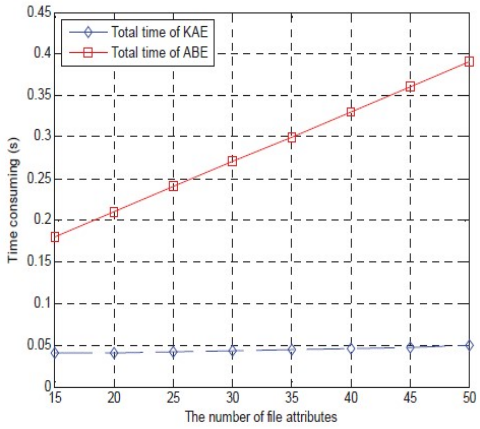

Fig 3.1: Total time of KAE and ABE

The MAH-ABE scheme's attribute-based encryption method took a lot longer to execute in Fig. 3.1 than our scheme's KAE algorithm. The ABE algorithm will take longer if the attribute voidance happens. Furthermore, the KAE technique is far less privileged than the increase rate of time spent with the amount of file attributes. The outcomes of the simulation demonstrate our scheme's increased efficiency.

The user needs very little time to sign the customized files in Fig. 3.2. The process of signing and authenticating takes very little time because the validation time only accounts for a small portion of the total. Thus, the program is successful in the eyes of the client.

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

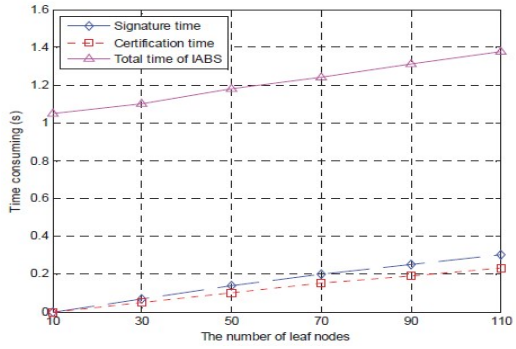

Fig 3.2: The signature and authentication time of IABS

4. CONCLUSION

The access control system (PS-ACS) proposed in this research is based on privacy protection and privilege separation. We logically divide users into personal domains (PSD) and public domains (PUD) based on the analysis of the cloud environment and user characteristics. The KAE algorithm is used in the PSD to detect people with read access permissions and significantly increase efficiency. To protect the secrecy of the user's identity, the IABS technique is used to obtain write permissions and distinguish between read and write permissions. We employ the HABE method in the PUD to achieve data sharing and avoid single point of failure problems. Additionally, the study examines the plan from the perspectives of efficiency and security, and the outcomes of the simulation are shown. The suggested plan demonstrates the viability and superiority of protecting data privacy in cloud-based services as compared to the MAH-ABE scheme.

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.