# MULTIPLE ATTRIBUTE AUTHORITIES BASED CLOUD DATA SECURITY USING SCP-ABE AND FILE AUDITING SCHEME

Sneha George[1] | T.B.Dharmaraj[2]

[1](UG Student, Christ the King Engineering College, snehassgeorge@gmail.com)

[2](Professor, Christ the King Engineering College, bellidharmaraj@gmail.com)

***Abstract—*** *Data access control is a challenging issue in public cloud storage systems. Cipher text-Policy Attribute-Based Encryption (CP- ABE) has been adopted as a promising technique to provide flexible, fine-grained and secure data access control for cloud storage with honest-but-curious cloud servers. However, in the existing CP-ABE schemes, the single attribute authority must execute the time-consuming user legitimacy verification and secret key distribution, and hence it results in a single-point performance bottleneck when a CP-ABE scheme is adopted in a large-scale cloud storage system. Users may be stuck in the waiting queue for a long period to obtain their secret keys, thereby resulting in low-efficiency of the system. Although multi authority access control schemes have been proposed, these schemes still cannot overcome the drawbacks of single-point bottleneck and low efficiency, due to the fact that each of the authorities still independently manages a disjoint attribute set .In this project, we propose a novel heterogeneous framework to remove the problem of single-point performance bottleneck and provide a more efficient access control scheme with an auditing mechanism. Our framework employs multiple attribute authorities to share the load of user legitimacy verification. Meanwhile, in our scheme, a CA (Central Authority) is introduced to generate secret keys for legitimacy verified users. Unlike other multi authority access control schemes, each of the authorities in our scheme manages the whole attribute set individually. To enhance security, we also propose an auditing mechanism to detect which AA (Attribute Authority) has incorrectly or maliciously performed the legitimacy verification procedure.*

***Keywords— Cipher Text-Policy, Encryption, Bottleneck, Central Authority, Attribute Authority***

## 1. INTRODUCTION

Cloud computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on- demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.,

Cloud storage is a promising and important service paradigm in cloud computing. Benefits of using cloud storage include greater accessibility, higher reliability, rapid deployment and stronger protection, to name just a few. Despite the mentioned benefits, this paradigm also brings forth new challenges on data access control, which is a critical issue to ensure data security. Since cloud storage is operated by cloud service providers, who are usually outside the trusted domain of data owners, the traditional access control methods in the Client/Server model are not suitable in cloud storage environment. The data access control in cloud storage environment has thus become a challenging issue.

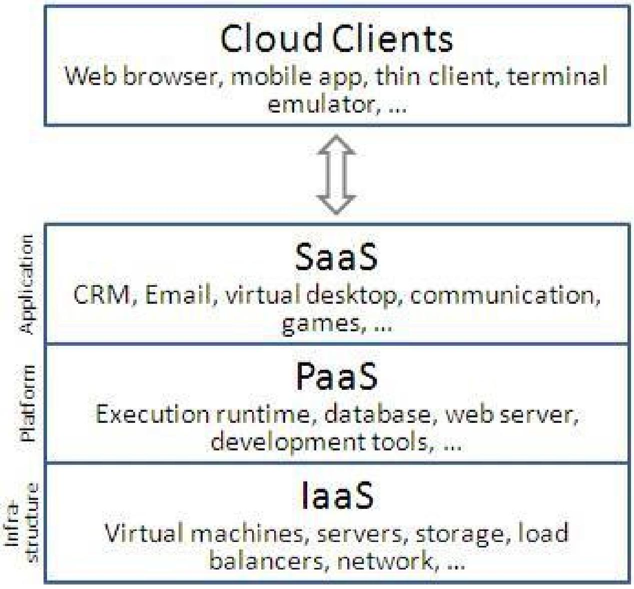
## 2.RELATED WORKS

Cipher text-Policy Attribute-Based Encryption (CP-ABE) has so far been regarded as one of the most promising techniques for data access control in cloud storage systems. This technology offers users flexible, fine-grained and secure access control of outsourced data. It was first formulated by Goyal et al. in. Then the first CP-ABE scheme was proposed by Bettencourt et al. in, but this scheme was proved secure only in the generic group model. Subsequently, some cryptographically stronger CP-ABE constructions were proposed, but these schemes imposed some restrictions that the original CP-ABE does not have. In , Waters proposed three efficient and practical CP- ABE schemes under stronger cryptographic assumptions as expressive as . To improve efficiency of this encryption technique, Emura et al. proposed a CP-ABE scheme with a constant cipher text length. Unlike the above schemes which are only limited to express monotonic access structures, Obtrovsky et al. proposed a more expressive CP-ABE scheme which can support non-monotonic access structures. Recently, Hohenberger and Waters proposed an online/ offline ABE technique for CPABE which enables the user to do as much pre-computation as possible to save online computation.

It's a promising technique for resource-limited devices. In general, there are two categories of CP-ABE schemes classified by the number of participating authorities in key distribution process. One category is the single-authority scheme, the other is multi-authority scheme. In single authority schemes , only one authority is involved to manage the universal attribute set, generate and distribute secret keys for all users. In , the authors respectively proposed CP-ABE schemes with efficient attribute revocation capability for data outsourcing systems. Wu et al. proposed a Multi-message Cipher text-Policy Attribute Based Encryption(MCP-ABE) which encrypts multiple messages within one cipher text so as to enforce flexible attribute based access control on scalable media .The literatures took the efficiency issue into consideration, but they mainly considered the computation complexity inside the cryptography algorithms rather than interaction protocols between different entities in the real world, such as the procedure of user legitimacy verification. To sum up, in single-authority schemes, the single-point performance bottleneck has not been widely addressed so far.

To meet some scenarios where users' attributes come from multiple authorities, some multi-authority schemes have been proposed.

Based on the basic ABE scheme, Chase et al proposed the first multi- authority scheme which allows multiple independent authorities to monitor attributes and distribute corresponding secret keys ,but involves acentral authority (CA). Subsequently, some multi-authority ABE schemes without CA have been proposed, such as . Since the first construction of CP-ABE a great many multi authority schemes have been conducted over CP-ABE. Muller et al. proposed the first multi-authority CP-ABE scheme in which a user's secret key was issued by an arbitrary number of attribute authorities and a master authority. Then Lewko et al. proposed a decentralized CP-ABE scheme where the secret keys can be generated fully by multiple authorities without a central authority. Ruj et alapplied Lewko's work for access control in cloud storage systems, and also proposed a revocation method. Lin et al. proposed a decentralized access control scheme based on threshold mechanism. In the authors proposed two efficient multi-authority CP-ABE schemes for data access control in cloud storage systems, where a central authority is only needed in system initialization phase. Based on the basic multi authority architecture, some other literatures tried to address the user identity privacy issue, policy update , and the accountability to prevent key abusing .However, in above multi-authority schemes, multiple authorities separately manage disjoint attribute sets. That is to say, for each attribute, only one authority could issue secret keys associated with it. Therefore,in large-scale systems, the single-point performance bottleneck still exists in multi-authority schemes due to the property that each of the multiple authorities maintains only a disjoint subset of attributes. Recently, we considered the single-point performance bottleneck of CP-ABE based schemes and devised a threshold multi-authority CP-ABE access control scheme in our another work . Different from other multi-authority schemes, in , multiple authorities jointly manage a uniform attribute set. Taking advantage of (t,n) threshold secret sharing, the master secret key can be shared among multiple authorities, and a legal user can generate his/her secret key by interacting with any t authorities. This scheme actually addressed the single-point bottleneck on both security and performance in CP-ABE based access control in public cloud storage. However, it is not efficient, because a user has to interact with at least t authorities, and thus introduces higher interaction overhead. In this paper, we present an efficient heterogeneous framework with single CA/multiple AAs to address the problem of single-point performance bottleneck. The novel idea of our proposed scheme is that the complicated and time-consuming user legitimacy verification is executed only once by one selected AA. Furthermore, an auditing mechanism is proposed to ensure the traceability of malicious AAs. Thus our scheme can not only remove the single- point performance bottleneck but also be able to provide a robust, high-efficient, and secure access control for public cloud storage.


Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

## 3 RAAC FRAMEWORK GENERATION

The system model of our design, which involves five entities: a central authority (CA), multiple attribute authorities (AAs), many data owners (Owners), many data consumers (Users), and a cloud service provider with multiple cloud servers(here, we mention it as cloud server.).

The central authority (CA) is the administrator of the entire system. It is responsible for the system construction by setting up the system parameters and generating public key for each attribute of the universal attribute set. In the system initialization phase, it assigns each user a unique Uid and each attribute authority a unique Aid. For a key request from a user, CA is responsible for generating secret keys for the user on the basis of the received intermediate key associated with the user's legitimate attributes verified by an AA. As an administrator of the entire system, CA has the capacity to trace which AA has incorrectly or maliciously verified a user and has granted illegitimate attribute sets .The attribute authorities (AAs) are responsible for performing user legitimacy verification and generating intermediate keys for legitimacy verified users. Unlike most of the existing multi-authority schemes where each AA manages a disjoint attribute set respectively, our proposed scheme involves multiple authorities to share the responsibility of user legitimacy verification and each AA can perform this process for any user independently. When an AA is selected, it will verify the users' legitimate attributes by manual labor or authentication protocols, and generate an intermediate key associated with the attributes that it has legitimacy-verified. Intermediate key is a new concept to assist CA to generate keys.The data owner (Owner) defines the access policy about who can get access to each file, and encrypts the file under the defined policy. First of all, each owner encrypts his/her data with a symmetric encryption algorithm. Then, the owner formulates access policy over an attribute set and encrypts the symmetric key under the policy according to public keys obtained from CA. After that, the owner sends the whole encrypted data and the encrypted symmetric key (denoted as ciphertext CT) to the cloud server to be stored in the cloud.The data consumer (User) is assigned a global user identity Uid by CA. The user possesses a set of attributes and is equipped with a secret key associated with his/her attribute set. The user can freely get any interested encrypted data from the cloud server. However, the user can decrypt the encrypted data if and only if his/her attribute set satisfies the access policy embedded in the encrypted data.The cloud server provides a public platform for owners to store and share their encrypted data. The cloud server doesn't conduct data access control for owners. The encrypted data stored in the cloud server can be downloaded freely by any user.

## 4. ENCRYPTION AND DECRYPTION

Encryption: The procedure of Encryption is performed by the data owner himself/herself. To improve the system's performance, the owner first chooses a random number $\kappa \in GT$ as the symmetric key

and encrypts the plaintext message M using κ with the symmetric encryption algorithm. The encrypted data can be denoted as $E\kappa(M)$. Then the owner encrypts the symmetric key κ using CP-ABE under the access policy A defined by himself/herself. Decryption: The procedure of Decryption is performed by the user. A user can freely query and download any interested encrypted data from the public cloud storage. However, he/she cannot decrypt data unless his/her attribute set satisfies the access structure embedded in the ciphertext.

## 5. KEY GENERATION AND DISTRIBUTION:

Key generation and distribution is totally different from those existing CP-ABE schemes. It involves the given user, a selected AA and CA. It divide the procedure into the following 4 steps.

STEP 1: Uj→ AAi. When a user Uj with the identity Uidj makes a secret key request, the user selects an AA(AAi with the identity Aidi) by a certain scheduling algorithm and sends the CertUid to show the validity of his/her identity, along with some proofs to show that he/she has the attribute set that he/she claims to have.

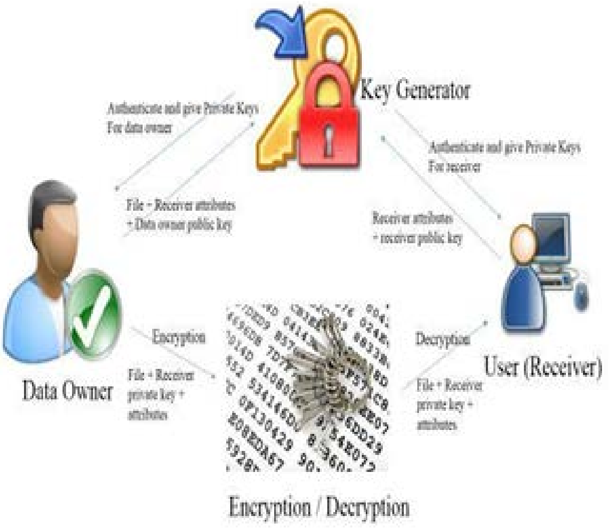
STEP 2: AAi → CA. The user legitimacy verification process may involve manual labor or verification protocols performed by AAi. After successful verification, AAi obtains the current timestamp value T S

STEP 3&STEP 4: CA → AAi → Uj. After receiving the message from the AA, CA first uses Aidi to obtain the corresponding stored public key PKAidi. Then CA checks whether the transmission delay

is within the allowed time interval $\Delta$ T . We assume that the current

time is T ′. If $T' - TS > \Delta T$ , CA stops here and sends REJ to the AA. Otherwise, CA continues to compute t1 = H(Uidj||TS||0), t2 =

H(Uidj||TS||1), and makes sure t1 and t2 haven't yet been re-used from the same user.



## 6. AUDITING & TRACING

Each AA may generate an intermediate key for any attribute set associated with a specific user, and then CA can generate the secret key for this user without any more verification. However, AAs can be compromised and cannot be fully trusted. Meanwhile, the user legitimacy verification is conducted by manual labor, and therefore AAs may maliciously or incorrectly generate an intermediate key for an unverified attribute set. A malicious user will try any possible means to gain the secret key associated with the specific attribute set to obtain the data access permission. Under this assumption, the user would often show abnormal behaviors. Usually, we need to hold the accountability of AAs to prevent the compromised or misbehaved ones from freely generating secret keys for malicious users. Like user accountability addressed, we assume that we have appropriate techniques to detect users' abnormal behaviors. The procedure of Auditing & Tracing is periodically performed or event-triggered by

CA to mandatorily ask a suspected user to securely submit Kx′ of a given attribute, L and TS in his/her gained secret key. In order to continue to obtain data, users have to cooperate to perform the process correctly. However, in order to deceive CA, a suspected user still has the motivation to submit a secret key component that doesn't belong to him/her. Thus, to implement an effective tracing, CA must confirm the received secret key components really belong to the given user. Based on the reasons mentioned above, the tracing method should be executed as the following two sub procedures

.Secret key ownership confirming. This procedure is executed to confirm that the received secret key component really belongs to the user who has submitted it .AA Tracing. This procedure is executed to trace and confirm which AA has generated the suspected user's secret key.

## 7. CONCLUSION

In this paper, we proposed a new framework, named RAAC, to eliminate the single-point performance bottleneck of the existing CP- ABE schemes. By effectively reformulating CPABE cryptographic technique into our novel framework, our proposed scheme provides a fine -grained, robust and efficient access control with one-CA/multi- AAs for public cloud storage. Our scheme employs multiple AAs to share the load of the time-consuming legitimacy verification and standby for serving new arrivals of users' requests. We also proposed an auditing method to trace an attribute authority's potential misbehavior. We conducted detailed security and performance analysis to verify that our scheme is secure and efficient . The security analysis shows that our scheme could effectively resist to individual and colluded malicious users, as well as the honest-but- curious cloud servers. Besides, with the proposed auditing & tracing scheme, no AA could deny its misbehaved key distribution. Further performance analysis based on queuing theory showed the superiority of our scheme over the traditional CP-ABE based access control schemes for public cloud storage.

## REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology Gaithersburg, 2011.

[2] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Advances in Cryptology–EUROCRYPT 2011. Springer, 2011, pp. 568–588

[3] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Transactions on Parallel & Distributed Systems, vol. 27, no. 9, pp. 2546–2559, 2016.

[4] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in in Proceedings of 2016 IEEE Conference on Computer Communications (INFOCOM 2016). IEEE, 2016, pp. 1–9.

[5] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," IEEE Transactions on Cloud Computing, vol. 2, no. 4, pp. 459–470, 2014.

[6] Y. Wu, Z. Wei, and H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing," IEEE Transactions on Multimedia, vol. 15, no. 4, pp. 778–788, 2013.

[7] J. Hur, "Improving security and efficiency in attribute based data sharing," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2271–2282, 2013.

[8] J. Hur and 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.

[9] J. Hong, K. Xue, W. Li, and Y. Xue, "TAFC: Time and attribute factors combined access control on time sensitive data in public cloud," in Proceedings of 2015 IEEE Global Communications Conference (GLOBECOM 2015). IEEE, 2015, pp. 1–6.