

Multi - Authority Based Approach for Manipulation of Encrypted Data in Cloud

R Mariammal¹, Venkatasubramanian C², Vishal Surya P A³, Goutham Kumar M⁴

Assistant Professor, Department of Computer Science and Engineering¹

UG Scholar, Department of Computer Science and Engineering^{2,3,4}

Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India

Abstract: *In this paper, we address the protection concerns of cloud storage under the scenario where users encrypt-then-outsource data, share their outsourced data with other users, and so the service provider may be queried for searching and retrieval of encrypted data. Because the main distinctive, we propose a security approach for storage, sharing, and retrieval of encrypted data within the fully as constructed supported attribute-based encryption (ABE) thus enabling access control mechanisms over both the encrypted data and also for the data retrieval task through search access control. Efforts have studied problems around this application scenario on different fronts: efficiency, flexibility, reliability, and security. Our suggested secure Multi-authority CP-ABKS (MABKS) system addresses such limitations and minimizes the computation and storage burden on resource-limited devices in cloud systems. Additionally, the MABKS system is extended to support malicious attribute authority tracing and attribute update. proposed a practical CP-ABE scheme, which offers user revocation and attributes updates. We proposed an efficient and feasible MABKS system to support multiple authorities, to avoid having performance bottlenecks at one point in cloud systems. Furthermore the presented MABKS system allows us to trace malicious.*

Keywords: Cloud Storage.

I. INTRODUCTION

Cloud storage service offers user an efficient thanks to share data and work as a team. Once someone of the team uploads a file to the server, other members are able to access and modify the file by Internet we propose a Multi-authority Attribute-Based Keyword Search (MABKS) scheme for cloud systems to mitigate challenges because of single-point performance bottleneck and high storage and computation requirements (which are unrealistic for resource-limited devices). Key differences between multi-authority architecture within the MABKS system and single-authority architecture in existing schemes are presented in Fig. 1. Specifically, each AA within the MABKS system maintains the complete attribute set and is answerable for verifying the validity of information users' certificates and generating intermediate secret keys for data users, and also the CA outputs the ultimate secret keys for DUs. For instance, the sole fully-trusted department (that acts as CA) in an exceedingly large company can generate the entire secret keys for staffs who are authorized to access important company documents, but are visiting be burdened with much computation overhead when there are massive staffs, and even suffer from single-point performance bottleneck if this department is compromised or breaks down. The corporate can rent multiple public servers (that act as AA's) provided by other enterprises (i.e., Tencent, Amazon, Alibaba, etc.) to eliminate the fully-trusted department's computation burden. Multi-authority architecture. Different from the previous single-authority CP-ABKS schemes (or traditional multi-authority CP-ABE schemes) that also cannot avoid the limitation of single-point performance bottleneck, the data structure within the MABKS system enables multiple AA's to separately execute time-consuming user certificate verification and intermediate secret key generation on behalf of CA, which significantly reduces CA's computation requirements.

II. SYSTEM ANALYSIS

2.1 Existing System

To address the problem of knowledge access control in cloud storage, there are quite a few schemes proposed, among which Attribute-Based Encryption (ABE) is considered one among the foremost promising techniques. A salient feature

of ABE is that it grants data owners direct control power supported access policies, to supply flexible, fine-grained and secure access control for cloud storage systems. In ABE schemes, the access control is achieved by using cryptography, where an owner's data is encrypted with an access structure over attributes, and a user's secret is labeled with his/her own attributes. Providing the attributes related to the user's secret key satisfy the access structure, can the user decrypt the corresponding cipher-text to get the plain-text. So far, the ABE based access control schemes for cloud storage are developed into two complementary categories, namely, single-authority scenario, and multi-authority scenario. Although existing ABE access control schemes have lots of attractive features, they're neither robust nor efficient in key generation. Since there's only 1 authority to blame of all attributes in single-authority schemes, offline/crash of this authority makes all secret key requests unavailable during that period. The similar problem exists in multi-authority schemes. Despite the amount of research efforts on this subject, existing ABE schemes haven't entirely solved the matter of keyword-based data retrieval.

2.1.1 Disadvantages

- Single-point performance bottleneck issue affects the efficiency of secret key generation service and immensely degrades the utility of the present schemes to conduct access control in large cloud storage systems.
- The inefficiency of the authority's service ends up in single-point performance bottleneck.
- In single-authority schemes, the sole authority must verify the legitimacy of users' attributes before generating secret keys for them.

2.2.2 Proposed System

We propose a Multi-authority Attribute-Based Keyword Search (MABKS) scheme for cloud systems to mitigate challenges thanks to single-point performance bottleneck and high storage and computation requirements (which are unrealistic for resource-limited devices). Key differences between multi-authority architecture within the MABKS system and single-authority architecture in existing schemes are presented in Fig. 1. Specifically, each AA within the MABKS system maintains the whole attribute set and is chargeable for verifying the validity of knowledge users' certificates and generating intermediate secret keys for data users, and so the CA outputs the ultimate secret keys for DUs. For instance, the sole fully-trusted department (that acts as CA) in an exceedingly large company can generate the full secret keys for staffs who are authorized to access important company documents, but are burdened with much computation overhead when there are massive staffs, and even suffer from single-point performance bottleneck if this department is compromised or breaks down.

2.2.1 Advantages

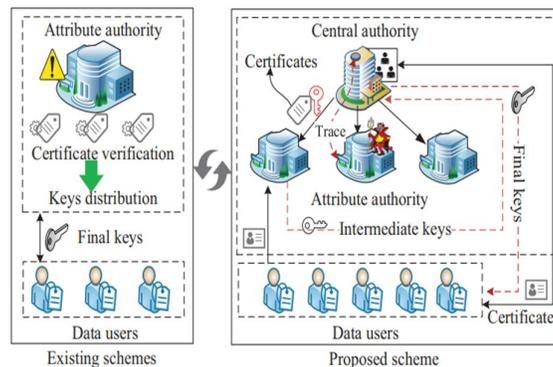
The heterogeneous access control framework to deal with the low

1. Efficiency and single-point performance bottleneck for cloud storage.
2. Our scheme includes an auditing mechanism that helps the system trace an AA's misbehavior on user's legitimacy verification.
3. Data confidentiality. Data content must be kept confidential to unauthorized users yet because the curious cloud server.
4. The procedure of key generation is split into two sub-procedures: 1) the procedure of user legitimacy verification; 2) the procedure of secret key generation and distribution.

III. SYSTEM ARCHITECTURE

3.1 User Module

The data consumer (User) is assigned a world user identity UID by CA. The user possesses a group of attributes and is provided with a secret key related to 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 provided that his/her attribute set satisfies the access policy embedded within the encrypted data.



3.2 Owner Module

The data Owner (Owner) defines the access policy about who can get access to every file, and encrypts the file under the defined policy. First, 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 in step with public keys obtained from CA. After that, the owner sends the full encrypted data and therefore the encrypted symmetric key (denoted as ciphertext CT) to the cloud server to be stored within the cloud.

3.3 Admin Module

Admin could be a superuser. They'll view all the user and owner details. Admin can view the chart supported most number of word search, they will add related word, so user can easily map related words for instance Ambiguity level 2 refers to instances that almost all people think as ambiguous. These instances contain two or more unrelated senses, like "apple" (fruit & company) and "jaguar" (animal & company). During this work, we only specialize in disambiguation of instances.

3.4 Attribute Authority Module

The attribute authorities (AA's) are accountable for performing user legitimacy verification and generating intermediate keys for legitimacy verified users. Unlike most of the present 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 every AA can perform this process for any user independently. When an AA is chosen, it'll verify the users' legitimate attributes by toil or authentication protocols, and generate an intermediate key related to the attributes that it's legitimacy-verified. Intermediate secret's a brand-new concept to help CA to get keys.

3.5 Central Authority Module

The central authority (CA) is the administrator of the whole system. It's chargeable for the system construction by putting in place the system parameters and generating a public key for every attribute of the universal attribute set. Within the system initialization phase, it assigns each user a singular UID and every attribute authority a novel Aid. For a key request from a user, CA is liable for generating secret keys for the user on the premise of the received intermediate key related to the user's legitimate attributes verified by an AA. As an administrator of the complete system, CA has the capacity to trace which AA has incorrectly or maliciously verified a user and has granted illegitimate attribute sets. 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 within the cloud server will be downloaded freely by any user.

3.6 Malicious AA's Tracing

The traditional traceable CPABE schemes mainly focus on the malicious data users who may leak their secret keys to unauthorized entities, while the extended MABKS system focuses on tracing the malicious AA's that incorrectly generate intermediate secret keys for data users in two phases (i.e., secret key ownership confirming, malicious AA's tracing).

IV. RESULTS

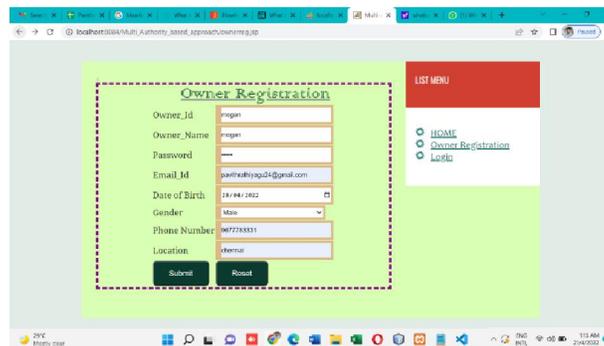
Home Page

This is the Homepage of the Website Where all the authorities, owner and users navigate from to their own portals.



Owner Registration

Data Owners use this page to register themselves and Upload the encrypted files to cloud.



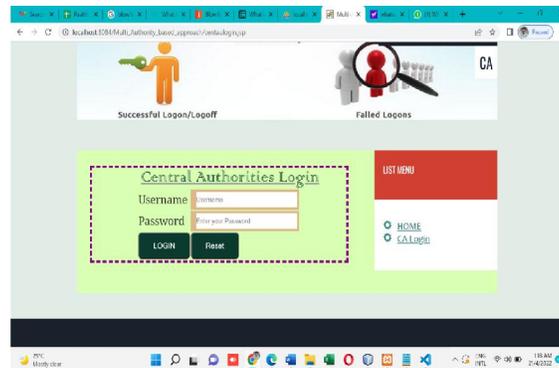
Attribute Authorities Registration

Attribute authorities use this portal to login and navigate through the page.



Central Authority Login:

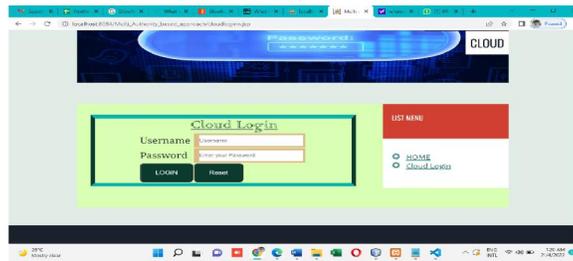
Central Authorities Enter their Login information here to Gain access to their portals.



Cloud Server Page

We have to login on the cloud server to access the files uploaded on cloud





User Registration Page

User registration is done here on this tab to upload the required Files on cloud.



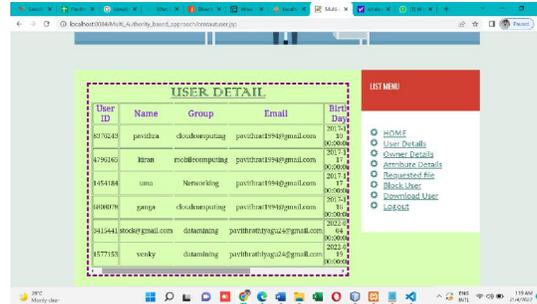
Central Authorities Home Page

Central Authorities use this homepage to navigate and do their actions



User Details

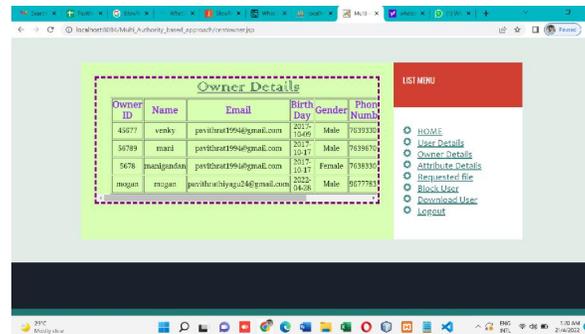
The list of Users Are given here on this section for the Attribute Authority to verify.



User ID	Name	Group	Email	Birth Day	Gender
877025	pavithra	cloudcomputing	pavithra1994@gmail.com	2017-11-19	Female
879162	kiran	cloudcomputing	pavithra1994@gmail.com	2017-11-19	Male
854184	uma	Networking	pavithra1994@gmail.com	2017-11-19	Female
808078	ganja	cloudcomputing	pavithra1994@gmail.com	2017-11-19	Female
841544	roshan@gmail.com	data mining	pavithratraya24@gmail.com	2022-04-15	Male
877125	venky	data mining	pavithratraya24@gmail.com	2022-04-15	Male

Owner details

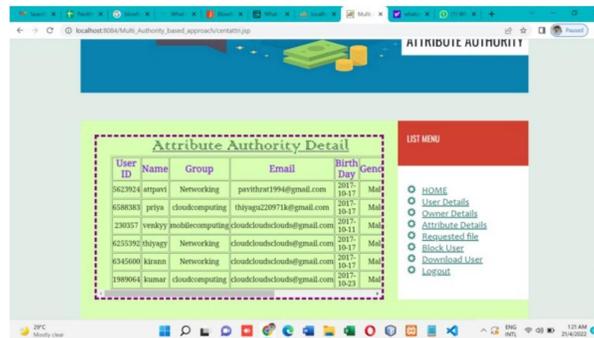
The List of the Data Owners is listed on this section.



Owner ID	Name	Email	Birth Day	Gender	Phone Number
45277	venky	pavithra1994@gmail.com	2017-11-19	Male	9833130
56789	marci	pavithra1994@gmail.com	2017-11-19	Male	9833130
5678	haanigandaa	pavithra1994@gmail.com	2017-11-19	Female	9833130
	mogan	pavithratraya24@gmail.com	2022-04-15	Male	8077781

AA's Details

The Details of all the Attribute Authorities are viewed in this Section



User ID	Name	Group	Email	Birth Day	Gender
862924	arpasi	Networking	pavithra1994@gmail.com	2017-11-19	Male
6588383	pritya	cloudcomputing	thiyaga229971@gmail.com	2017-11-19	Male
23037	venky	cloudcomputing	cloudcloudcloud@gmail.com	2017-11-19	Male
8255392	thiyagi	Networking	cloudcloudcloud@gmail.com	2017-11-19	Male
6345000	kiran	Networking	cloudcloudcloud@gmail.com	2017-11-19	Male
198064	kumar	cloudcomputing	cloudcloudcloud@gmail.com	2017-11-19	Male

V. CONCLUSION

In this paper, we proposed an efficient and feasible MABKS system to support multiple authorities, in order to avoid having performance bottleneck at a single point in cloud systems. Furthermore, the presented MABKS system allows us to trace malicious AAs (e.g., to prevent collusion attacks) and support attribute update (e.g., to avoid unauthorized access using outdated secret keys). We also evaluated the system’s performance and demonstrated that significant computation and storage cost reductions. We review the features, advantages and disadvantages of different multi-authority attribute based encryption schemes. The ultimate goal of designing a MAABE scheme is to develop a secure, robust, expressive and efficient multi-authority attribute based encryption system. Supporting client renouncement is an essential issue in the original application, and this is an impressive test the application. Blocking system is very useful for our concept to Avoid from attacker we Include blocking system.

REFERENCES

- [1]. A. Bagherzandi, B. Hore, and S. Mehrotra, Search over Encrypted Data. Boston, MA, USA: Springer, 2011, pp. 1088–1093.
- [2]. H. Pham, J. Woodworth, and M. A. Salehi, “Survey on secure search over encrypted data on the cloud,” *Concurrency Comput. Pract. Exper.*, vol. 31, p. 1–15, Apr. 2019.
- [3]. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: Improved definitions and efficient constructions,” in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2011, pp. 79–88, 2006.
- [4]. M. Zeng, H.-F. Qian, J. Chen, and K. Zhang, “Forward secure public key encryption with keyword search for outsourced cloud storage,” *IEEE Trans. Cloud Comput.*, early access, Sep. 27, 2019, doi: 10.1109/TCC.2019.2944367.
- [5]. S. Kamara, C. Papamanthou, and T. Roeder, “Cs2: A searchable cryptographic cloud storage system,” Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2011-58, May 2011.
- [6]. W. Song, B. Wang, Q. Wang, Z. Peng, W. Lou, and Y. Cui, “A privacy preserved full-text retrieval algorithm over encrypted data for cloud storage applications,” *J. Parallel Distrib. Comput.*, vol. 99, pp. 14–27, Jan. 2017.
- [7]. A. G. Kumbhare, Y. Simmhan, and V. Prasanna, “Designing a secure storage repository for sharing scientific datasets using public clouds,” in *Proc. 2nd Int. workshop Data Intensive Comput. Clouds*, 2011, pp. 31–40.
- [8]. Z. Yang, J. Tang, and H. Liu, “Cloud information retrieval: Model description and scheme design,” *IEEE Access*, vol. 6, pp. 15420–15430, 2018.
- [9]. H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, and K. Li, “CP-ABSE: A ciphertext-policy attribute-based searchable encryption scheme,” *IEEE Access*, vol. 7, pp. 5682–5694, 2019.
- [10]. A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Advances in Cryptology (Lecture Notes in Computer Science)*, vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.