Robust and Auditable Access Control with Multiple Attribute Authorities for Public Cloud Storage

V. Pandurangaiah[1], Dr. G. Anjan Babu[2]

[1] *Student, Dept. of MCA, SVU. College of C M & C's*

[2] *Professor, Dept. of MCA, SVU. College of C M & C's, Tirupati,A.P*

Abstract*- Data access control is a challenging issue in public cloud storage systems. Cipher text-Policy in 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 paper, 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. Analysis shows that our system not only guarantees the security requirements but also makes great performance improvement on key generation.

***Index Terms*- Cloud storage, Access control, Auditing, CPABE.**

I. ARCHITECTURE DIAGRAM

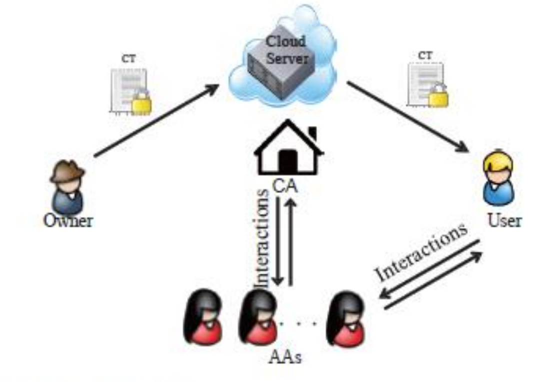

Fig. 1. System model

II. EXISTING SYSTEM

Cloud computing is internet based computing which enables sharing of services. Many users place their data in the cloud. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in cloud computing a very challenging and potentially form id able task, especially for users with constrained computing resources and capabilities. So correctness of data and security is a prime concern. This article studies the problem of ensuring the integrity and security of data storage in Cloud Computing. Security in cloud is achieved by signing the data block before sending to the cloud. Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

III. PROPOSED SYSTEM

Client/Server model are not suitable in cloud storage environment. The data access control in cloud storage environment has thus become a challenging issue. To address the issue of data access control in cloud storage, there have been quite a few schemes proposed, among which

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is regarded as one of the most promising techniques. A salient feature of CP-ABE is that it grants data owners direct control power based on access policies, to provide flexible, fine grained and secure access control for cloud storage systems. In CP-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 key is labelled with his/her own attributes. Only if the attributes associated with the user's secret key satisfy the access structure, can the user decrypt the corresponding ciphertext to obtain the plaintext. So far, the CP-ABE based access control schemes for cloud storage have been developed into two complementary categories, namely, single-authority scenario , and multi authority scenario . Although existing CP-ABE access control schemes have a lot of attractive features, they are neither robust nor efficient in key generation. Since there is only one authority in charge 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, since each of multiple authorities manages a disjoint attribute set.

Modules:

1. User Module
2. Owner Module
3. Attribute authority Module
4. Central authority Module
5. Chart Module

IV. MODULE DESCRIPTION

User Module:

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 theencrypted data.

Owner module:

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.

Admin module:

Admin is a super user. they can view all the user and owner details.admin can view the chart based on most number of word search , they can add related word ,so user can easily mapping arelated words for example Ambiguity level 2 refers to instances that most people think as ambiguous. These instances contain two or more unrelated senses, such as “apple” (fruit & company) and “jaguar” (animal & company). In this work, we only focus on disambiguation of instances.

Attribute Authority module:

*The attribute authorities (*AA*s)* 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.

Central Authority module:

*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 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.

Chart module:

chart module,chart module based on number of file download in particular user ,central authority can easily find out which file will be download more.

V. 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-*AA*s for public cloud storage. Our scheme employs multiple *AA*s 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.