International Journal of Science Engineering and Advance Technology

A Well-Organized Revocable Data Access Control for Multi-Authority Cloud Storage

Harika Mallisetti[1], Jaladi Praveena[2]
[1] M.Tech ,(CSE), NRI Institute of Technology, A.P., India.
[2]Assistant Professor, Dept. of Computer Science & Engineering, NRI Institute of Technology, A.P., India.

***Abstract* — Ensuring data security while accessing data in the cloud is a paramount importance. Due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Ciphertext-Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners more direct control on access policies. However, it is difficult to directly apply existing CP-ABE schemes to data access control for cloud storage systems because of the attribute revocation problem. In this paper, we design an expressive, efficient and revocable data access control scheme for multi-authority cloud storage systems, where there are multiple authorities co-exist and each authority is able to issue attributes independently. Specifically, we propose a revocable multi-authority CP-ABE scheme, and apply it as the underlying techniques to design the data access control scheme. Our attribute revocation method can efficiently achieve both forward security and backward security. The analysis and simulation results show that our proposed data access control scheme is secure in the random oracle model and is more efficient than previous works.**

***Keywords* — Access control, multi-authority, CP-ABE, attribute revocation, cloud storage**

I. INTRODUCTION

Among the services in the cloud, Cloud storage is an important service of cloud computing [1], which offers services for data owners to host their data in the cloud. This new paradigm of data hosting and data access services introduces a great challenge to data access control. Because the cloud server cannot be fully trusted by data owners, they can no longer rely on servers to do access control. Ciphertext-Policy Attribute-based Encryption (CP-ABE) [2], [3] is regarded as one of the most suitable technologies for data access control in cloud storage systems, because it gives the data owner more direct control on access policies. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. The data owner defines the access policies and encrypts data according to the policies. Each user will be issued a secret key reflecting its attributes. A user can decrypt the data only when its attributes satisfy the access policies.

There are two types of CP-ABE systems: single-authority CP-ABE [2], [3], [4], [5] where all attributes are managed by a single authority, and multi-authority CP-ABE [6] where attributes are from different domains and managed by different authorities. Multi-authority CP-ABE is more appropriate for data access control of cloud storage systems, as users may hold attributes issued by multiple authorities and data owners may also share the data using access policy defined over attributes from different authorities. For example, in an E-health system, data owners may share the data using the access policy ‘‘Doctor AND Researcher’’, where the attribute ‘‘Doctor’’ is issued by a medical organization and the attribute ‘‘Researcher’’ is issued by the administrators of a clinical trial. However, it is difficult to directly apply these multi-authority CP-ABE schemes to multi-authority cloud storage systems because of the attribute revocation problem.

In multi-authority cloud storage systems, users’ attributes can be changed dynamically. A user may be entitled some new attributes or revoked some current attributes. And his permission of data access should be changed accordingly. However, existing attribute revocation methods[7] either rely on a trusted server or lack of efficiency, they are not suitable for dealing with the attribute revocation problem in data access control in multi-authority cloud storage systems.

In this paper, we first propose a revocable multiauthority CP-ABE scheme, where an efficient and secure revocation method is proposed to solve the attribute revocation problem in the system. As described in Table 1, our attribute revocation method is efficient in the sense that it incurs less communication cost and computation cost, and is secure in the sense that it can achieve both backward security (The revoked user cannot decrypt any new ciphertext that requires the revoked attribute to decrypt) and forward security (The newly joined user can also decrypt the previously published ciphertexts1, if it has sufficient attributes). Our scheme does not require the server to be fully trusted, because the key update is

enforced by each attribute authority not the server. Even if the server is not semi-trusted in some scenarios, our scheme can still guarantee the backward security. Then, we apply our proposed revocable multi-authority CP-ABE scheme as the underlying techniques to construct the expressive and secure data access control scheme for multi-authority cloud storage systems.

Compared to the conference version [8] of this work, we have the following improvements:

1. We modify the framework of the scheme and make it more practical to cloud storage systems, in which data owners are not involved in the key generation. Specifically, a user's secret key is not related to the owner's key, such that each user only needs to hold one secret key from each authority instead of multiple secret keys associated to multiple owners.

2. We greatly improve the efficiency of the attribute revocation method. Specifically, in our new attribute revocation method, only the ciphertexts that associated with the revoked attribute needs to be updated, while in [8], all the ciphertexts that associated with any attribute from the authority (corresponding to the revoked attribute) should be updated. Moreover, in our new attribute revocation method, both the key and the ciphertext can be updated by using the same update key, instead of requiring the owner to generate an update information for each ciphertext, such that owners are not required to store each random number generated during the encryption.

3. We also highly improve the expressiveness of our access control scheme, where we remove the limitation that each attribute can only appear at most once in a ciphertext.

We believe that sharing data among multiple users is perhaps one of the most engaging features that motivate cloud storage. Therefore, it is also necessary to ensure the integrity of shared data in the cloud is correct. Existing public auditing mechanisms can actually be extended to verify shared data integrity. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers.

II. PROBLEM STATEMENT

System Model

We consider a data access control system in multi-authority cloud storage, as described in Fig. 1. There are five types of entities in the system: a certificate authority (CA), attribute authorities (AAs), data owners (owners), the cloud server (server) and data consumers (users).

The CA is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and AAs in the system. For each legal user in the system, the CA assigns a global unique user identity to it and also generates a global public key for this user. However, the CA is not involved in any attribute management and the creation of secret keys that are associated with attributes. For example, the CA can be the Social Security Administration, an independent agency of the United States government. Each user will be issued a Social Security Number (SSN) as its global identity.

Every AA is an independent attribute authority that is responsible for entitling and revoking user's attributes according to their role or identity in its domain. In our scheme, every attribute is associated with a single AA, but each AA can manage an arbitrary number of attributes. Every AA has full control over the structure and semantics of its attributes. Each AA is responsible for generating a public attribute key for each attribute it manages and a secret key for each user reflecting his/her attributes.

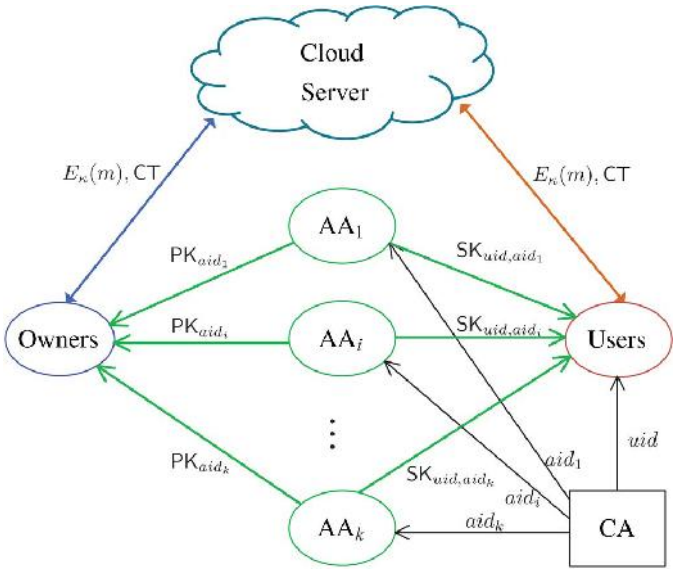

Figure 1: System model of data access control in multi-authority cloud storage

Each user has a global identity in the system. A user may be entitled a set of attributes which may come from multiple attribute authorities. The user will receive a secret key associated with its attributes entitled by the corresponding attribute authorities.

Each owner first divides the data into several components according to the logic granularities and encrypts each data component with different content keys by using symmetric encryption techniques. Then, the owner defines the access policies over attributes from multiple attribute authorities and encrypts the content keys under the policies. Then, the owner sends the encrypted data to the cloud server together with the ciphertexts. They do not rely on the server to do data access control. But, the access control happens inside the cryptography.

That is only when the user's attributes satisfy the access policy defined in the ciphertext; the user is able to decrypt the ciphertext. Thus, users with different attributes can decrypt different number of content keys and thus obtain different granularities of information from the same data.

Security Model

In multi-authority cloud storage systems, we make the following assumptions:

- The CA is fully trusted in the system. It will not collude with any user, but it should be prevented from decrypting any ciphertexts by itself.
- Each AA is trusted but can be corrupted by the adversary.
- The server is curious but honest. It is curious about the content of the encrypted data or the received message, but will execute correctly the task assigned by each attribute authority.
- Each user is dishonest and may collude to obtain unauthorized access to data.

III. OUR DATA ACCESS CONTROL SCHEME

Overview

To design the data access control scheme for multiauthority cloud storage systems, the main challenging issue is to construct the underlying Revocable Multiauthority CP-ABE protocol. In [6], Chase proposed a multi-authority CP-ABE protocol, however, it cannot be directly applied as the underlying techniques because of two main reasons: 1) Security Issue: Chase's multi-authority CP-ABE protocol allows the central authority to decrypt all the ciphertexts, since it holds the master key of the system; 2) Revocation Issue: Chase's protocol does not support attribute revocation.

We propose a new revocable multi-authority CP-ABE protocol based on the single-authority CP-ABE proposed by Lewko and Waters in [9]. That is we extend it to multiauthority scenario and make it revocable. We apply the techniques in Chase's multi-authority CP-ABE protocol [6] to tie together the secret keys generated by different authorities for the same user and prevent the collusion attack. Specifically, we separate the functionality of the authority into a global certificate authority (CA) and multiple attribute authorities (AAs). The CA sets up the system and accepts the registration of users and AAs in the system.7 It assigns a global user identity uid to each user and a global authority identity aid to each attribute authority in the system. Because the uid is globally unique in the system, secret keys issued by different AAs for the same uid can be tied together for decryption. Also, because each AA is associated with an aid, every attribute is distinguishable even though some AAs may issue the same attribute.

To deal with the security issue in [6], instead of using the system unique public key (generated by the unique master key) to encrypt data, our scheme requires all attribute authorities to generate their own public keys and uses them to encrypt data together with the global public parameters. This prevents the certificate authority in our scheme from decrypting the ciphertexts.

Secret Key Generation

Each user *uid* is required to authenticate itself to the AA_{aid} before it can be entitled some attributes from the AAaid. The user submits its certificate CertificateðuidÞ to the AAaid. The AA_{aid} then authenticates the user by using the verification key issued by the CA. If it is a legal user, the AA_{aid} entitles a set of attributes $S_{uid,aid}$ to the user uid according to its role or identity in its administration domain. Otherwise, it aborts. Then, the AAaid generates the user's secret key $SK_{uid,aid}$ by running the secret key generation algorithm SKeyGen. It chooses a random number $t_{uid,aid}$ € Z_p and computes the user's secret key as

$$\mathsf{SK}_{uid,aid} = \Big(K_{uid,aid} = g^{\alpha_{aid}} g^{v u_{uid}} g^{b l_{uid,aid}}, K'_{uid,aid} = g^{l_{uid,aid}},$$
$$\forall x_{aid} \in S_{uid,aid} : K_{x_{aid},uid} = g^{u'_{uid} l_{uid,aid} \beta_{aid}} H(x_{aid})^{v r_{aid} \beta_{aid} (u_{uid} + \gamma_{aid})} \Big).$$

Data Decryption

All the legal users in the system can freely query any interested encrypted data. Upon receiving the data from the server, the user runs the decryption algorithm Decrypt to decrypt the ciphertext by using its secret keys from different AAs. Only the attributes the user possesses satisfy the access structure defined in the ciphertext CT, the user can get the content key.

Attribute Revocation

As we described before, there are two requirements of the attribute revocation: 1) The revoked user (whose attribute is revoked) cannot decrypt new ciphertexts encrypted with new public attribute keys (Backward Security); 2) the newly joined user who has sufficient attributes should also be able to decrypt the previously published ciphertexts, which are encrypted with previous public attribute keys (Forward Security). For example, in a university, some archive

documents are encrypted under the policy ‘‘CS Dept. AND (Professor OR PhD Student)’’, which means that only the professors or PhD students in CS department are able to decrypt these documents. When a new professor/PhD student joins the CS department of the university, he/she should also be able to decrypt these documents. Our attribute revocation methods can achieve both forward security and backward security.

Communication Cost

The communication cost of the normal access control is almost the same. Here, we only compare the communication cost of attribute revocation. The communication cost of attribute revocation in [13] is linear to the number of ciphertexts which contain the revoked attribute

Computation Efficiency

We implement our scheme and DACC scheme on a Linux system with an IntelCore 2 DuoCPU at 3.16GHz and 4.00 GB RAM. The code uses the Pairing-Based Cryptography (PBC) library version 0.5.12 to implement the access control schemes. We use a symmetric elliptic curve α curve, where the base field size is 512-bit and the embedding degree is 2. The α -curve has a 160-bit group order, which means p is a 160-bit length prime. All the simulation results are the mean of 20 trials.

We compare the computation efficiency of both encryption and decryption in two criteria: the number of authorities and the number of attributes per authority. Fig. 3a describes the comparison of encryption time versus the number of authorities, where the involved number of attributes per authority is set to be 10. Fig. 3c gives the encryption time comparison versus the number of attributes per authority, where the involved number of authority is set to be 10. It is easy to find that our scheme incurs less encryption time than DACC scheme.

IV. Related Work

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [2]-[3] is a promising technique that is designed for access control of encrypted data. There are two types of CP-ABE systems: single authority CP-ABE [2], [3], [4], [5] where all attributes are managed by a single authority, and multi-authority CP-ABE [6], [7], [8] where attributes are from different domains and managed by different authorities. Multi-authority CP-ABE is more appropriate for the access control of cloud storage systems, as users may hold attributes issued by multiple authorities and the data owners may share the data using access policy defined over attributes from different authorities. However, due to the attribute revocation problem, these multi-authority CP-ABE schemes cannot be directly applied to data access control for such multi-authority cloud storage systems.

To achieve revocation on attribute level, some re encryption-based attribute revocation schemes [9], [10] are proposed by relying on a trusted server. We know that the cloud server cannot be fully trusted by data owners, thus traditional attribute revocation methods are no longer suitable for cloud storage systems. Ruj, Nayak and Ivan proposed a DACC scheme [11], where an attribute revocation method is presented for the Lewko and Waters’ decentralized ABE scheme [8]. Their attribute revocation method does not require a fully trusted server. But, it incurs a heavy communication cost since it requires the data owner to transmit a new ciphertext component to every non-revoked user.

V. Conclusion

In this paper, we proposed a revocable multi-authority CPABE scheme that can support efficient attribute revocation. Then, we constructed an effective data access control scheme for multi-authority cloud storage systems. We also proved that our scheme was provable secure in the random oracle model. The revocable multi-authority CPABE is a promising technique, which can be applied in any remote storage systems and online social networks etc.

References

[1] P. Mell and T. Grance, ‘‘The NIST Definition of Cloud Computing,’’ National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., 2009.

[2] J. Bethencourt, A. Sahai, and B. Waters, ‘‘Ciphertext-Policy Attribute-Based Encryption,’’ in Proc. IEEE Symp. Security and privacy (S&P’07), 2007, pp. 321-334.

[3] B. Waters, ‘‘Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,’’ in Proc. 4th Int’l Conf. Practice and Theory in Public Key Cryptography (PKC’11), 2011, pp. 53-70.

[4] V. Goyal, A. Jain,O. Pandey, andA. Sahai, ‘‘Bounded Ciphertext Policy Attribute Based Encryption,’’ in Proc. 35th Int’l Colloquium on Automata, Languages, and Programming (ICALP’08), 2008, pp. 579-591.

[5] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B.Waters, ‘‘Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption,’’ in Proc. Advances in Cryptology-EUROCRYPT’10, 2010, pp. 62-91.

[6] M. Chase, ‘‘Multi-Authority Attribute Based Encryption,’’ in Proc. 4th Theory of Cryptography Conf. Theory of Cryptography (TCC’07), 2007, pp. 515-534.

[7] M. Chase and S.S.M. Chow, ‘‘Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,’’ in Proc. 16th ACM Conf. Computer and Comm. Security (CCS’09), 2009,pp. 121-130.

[8] A.B. Lewko and B. Waters, ‘‘Decentralizing Attribute-Based Encryption,’’ in Proc. Advances in Cryptology-EUROCRYPT’11, 2011, pp. 568-588.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, ‘‘Attribute Based Data Sharing with Attribute Revocation,’’ in Proc. 5th ACM Symp. Information, Computer and Comm. Security (ASIACCS’10), 2010, pp. 261-270.

[10] M. Li, S. Yu, Y. Zheng, K. Ren, andW. Lou, ‘‘Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption,’’ IEEE Trans. Parallel Distributed Systems, vol. 24, no. 1, pp. 131-143, Jan. 2013.

[11] J. Hur and D.K. Noh, ‘‘Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,’’ IEEE Trans. Parallel Distributed Systems, vol. 22, no. 7, pp. 1214-1221, July 2011.