Data Sharing for Multiple Groups with Privacy Preservation in the Cloud

Liting Rao Qingqing Xie Hui Zhao
Jiangsu University
School of Computer Science and Communication Engineering

优胜奖

Abstract

With almost unlimited storage capacity and low maintenance cost, cloud storage becomes a convenient and efficient way for data sharing among cloud users. However, this introduces the challenges of access control and privacy protection when data sharing for multiple groups, as each group usually has its own encryption and access control mechanism to protect data confidentiality. In this paper, we propose a multiple-group data sharing scheme with privacy preservation in the cloud. This scheme constructs a flexible access control framework by using group signature, ciphertext-policy attribute-based encryption and broadcast encryption, which supports both intra-group and cross-group data sharing with anonymous access. Furthermore, our scheme supports efficient user revocation. The security and efficiency of the scheme are proved thorough analysis and experiments.

Scheme Design

1. System Model

The architecture of our multi-group data sharing system is shown in Fig. 1. The system consists of four entities: group manager, group user, cloud and key generation center (KGC). The key idea is to divide the system into multiple trust domains according to the group, and each group is a trust domain. To protect user privacy, users anonymously access data in the cloud, only the group manager knows the identity of the group members in his group. File access in the same trust domain is called intra-group file access, and file access between different trust domains is called cross-group data access.

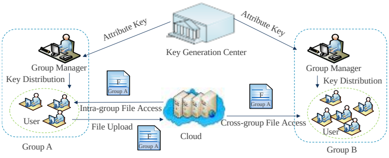

Fig.1.System Model

2. System Construction

This section describes the details of our scheme for the system construction, including system initialization, key distribution, file generation, file access and user revocation.

1) System Initialization

System initialization is performed by the key generation center and the manager of each group. The key generation center generates the system public parameter *PK* and the master key *MSK* by algorithm *AttSetup*. The manager is responsible for the initialization within his own group. The group public parameter *GPK* and the group master key GMK are generated by algorithm *GroupSetup*.

2) Key Distribution

First, the group manager generates a set of attributes *S* for the group, and uses algorithm *Sign*(m, K_{sig}) to generate the group signature σ_S for message $m = (ID_{group}, S)$, where K_{sig} denotes the key of the group signature. Then he sends the message (ID_{group}, S, σ_S) to the KGC, where ID_{group} stands for group identification. KGC runs algorithm *Verify*(m, σ_S, *GPK*) to verify whether the signature is valid, then runs algorithm *AttKeyGen*(*PK*, *S*, *MSK*) to generate the corresponding attribute key *AttKey* and send it to the group manager. After that the group manager runs the algorithm *UserKeyGen*(*GPK*, *GMK*,ID_i) to generate the user key *SK* for each group member, and saves (ID_i, *SK*) into the group user list. Finally, the group manager computes the proxy key *PXK* and uploads it into the cloud.

$$PXK = (\forall ID_i \in \mathrm{RL}: P(ID_i), A_i, x)$$

3) File Generation

First, the user randomly chooses a symmetric encryption key *K* to encrypt the file *M* and get the ciphertext C_K.The algorithm *BroEnc*(*K*, *SK*) and *AttEnc*(*K*, *SK*, *Γ*) are respectively used to encrypt the key *K* to obtain the ciphertext *BCT* and *ACT*. Computing the group signature σ for the message (ID_{group} , ID_{data} , C_K , *BCT*, *ACT*, t_{data}), where ID_{data} denotes the identity of file, and t_{data} denotes the current time. The generated file F as shown in Table 1. Uploading F to the cloud. The cloud runs *Revocation Verification*(*PXK*, *GPK*, σ) to verify if the user has been revoked. If not revoked, file F will be successfully uploaded to the cloud.

Table 1. File Format For Updating Data

Group ID	Data ID	Ciphertext	Time	Signature
ID_{group}	ID_{data}	C_K, *BCT*, *ACT*	t_{data}	σ

4) File Access

The user sends the message (ID_{group}, ID_{data}, $\lambda_i(i \in (\mathrm{RL})$, σ_{data}) to the cloud to access the file with ID_{data}, where σ_{data} denotes the signature for (ID_{group}, ID_{data} $\lambda_i(i \in (\mathrm{RL}))$, For the calculation method of λ_i, see the scheme [12]. The cloud first verifies revocation and the group signature, denying access if the user has been revoked or the group signature is invalid. Then the cloud runs *ProxykeyGen*(*Ciphertext*, *PXK*) to compute the latest partial decryption key according to access type. If the user wants to access data within the group, the cloud sends (*Ciphertext*, *BK*) to user, otherwise, sends (*Ciphertext*, *AK*) for cross-group file access. Finally the user runs *BroadcastDecrypt*(*SK*, *BCT*, *BK*) or *AttDecrypt*(*SK*, *ACT*, *AK*) to get the symmetric encryption key *K* and decrypts the ciphertext.

5) User Revocation and Accountability

User revocation is performed via a revocation list (RL), which store the tuple (ID_i, A_i, x) for each user that has been revoked. When a user is revoked, the manager updates the revocation list and computes the new *PXK* for the cloud.

User accountability is also done by the group manager. Given a group signature σ the manager runs the algorithm *Open*(*PK*, *GMK*, σ) to get A_i, and then reveals the identity of the signer by looking up the group user list.

Performance Analysis

1. Theoretical analysis

We summarize the computation overhead of each operation at each entity side in Table 2. We mainly consider the most expensive cryptographic operations, i.e., exponentiations and bilinear maps. We let t_{exp0}, t_{exp1}, t_{exp2} and t_{par} denote the evaluation time of an exponentiation operations in G_0 , G_1 , G_2 and bilinear pairing respectively.

2. Experimental analysis

To evaluate the performance, we conducted some experiments for user revocation. The bilinear cryptographic operations are implemented by using Java programming language with JPBC Library. We ran the experiments on Windows machine with Intel(R) Core (TM) i7-9700 @ 3.00 GHZ,16GB memory. And the implementation uses MNT curves with a 159-bit base field. We assumed that the maximum number of revoked users is 30 and used symmetric key of 256-bits AES to encrypt data (about 1KB).

Table 2. Computation Complexity of Each Operation

Operation	Computations	Entity
System Setup	$3t_{exp0} + 2t_{exp1} + 1t_{par}$	Manager
Key distribution	$3t_{exp0} + 4t_{exp1}$	Manager
File Generation	$3t_{exp0} + 2t_{exp1} + 2t_{par}$	User
File Access	ACT: $1t_{exp2} + 3t_{par}$	User
	BCT: $nt_{exp2} + (3n + 2)t_{par}$	

In Fig. 2, we compare the overhead of user revocation on the cloud side in our scheme with that in Shen et al.'s scheme [8]. In Shen et al.'s scheme, in order to revoke group users, the cloud computes re-encrypting RSA ciphertext *CipherC*=*(*CipherC*)*e*e* for all files stored in cloud. In contrast, the cloud in our scheme does not need any operations.

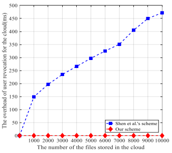

Fig. 2.Computation overhead on the cloud side

In Fig. 3, we compare the overhead of user revocation on the group manager side in our scheme with that in Shen et al.'s scheme. our proposed scheme achieves high revocation efficiency on both the cloud side and the group manager side.

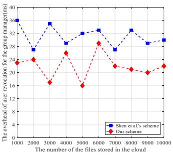

Fig. 3. Computation overhead on the group manager side.

结论

In this paper, We proposed a privacy preserving data sharing scheme for multiple groups in the cloud. Users can access the cloud anonymously, and do not need to present their identity to obtain cross-group access rights. In addition, based on CP-ABE and broadcast encryption, Moreover, our scheme supports the flexible access control with efficient user revocation. The analysis shows that the proposed scheme meets the expected safety requirements and ensures the efficiency.

主要参考文献

[1] X. Liu, Y. Zhang, B. Wang and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," in IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191, June 2013.

[2] S. Kamara and K. Lauter, "Cryptographic Cloud Storage, " Proc.Int'l Conf. Financial Cryptography and Data Security (FC), pp. 136149, Jan. 2010.

[3] S. Maiti and S. Misra, "P2B: Privacy Preserving Identity-Based Broadcast Proxy Re-Encryption," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5610-5617, May 2020.

联系方式

Contact Person：Liting Rao
Email：1044220275@qq.com