Data Sharing for Multiple Groups with Privacy Preservation in the Cloud

Liting Rao
Jiangsu University
School of Computer Science and Communication Engineering
Zhenjiang, China
rltujs@163.com

Qingqing Xie
Jiangsu University
School of Computer Science and Communication Engineering
Zhenjiang, China
xieqq@ujs.edu.cn

Hui Zhao
Jiangsu University
School of Computer Science and Communication Engineering
Zhenjiang, China
zhaohui@ujs.edu.cn

2020 International Conference on Internet of Things and Intelligent Applications (ITIA) | 978-1-7281-9301-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITIA50152.2020.9312318

***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.**

Keywords—cloud computing, multiple groups data sharing, privacy-preserving, access control

I. INTRODUCTION

Nowadays data have become important resources, sharing is now considered to be an inevitable trend to improve the value of data resources. With the help of cloud service, users can enjoy high quality sharing services while saving a lot of local infrastructure investment. Data sharing in the cloud, however, has a series of privacy and security risks as the cloud is out of the trust domain of the data owner.

There are many practical scenarios for data sharing. Consider the members of a research institution want to store and share their research data with each other. In order to reduce management and storage overhead, they store and share data in the cloud. But some research projects include sensitive commercial or national secrets. To protect data confidentiality, files are often encrypted before uploading to the cloud. Furthermore, users prefer to share data anonymously for preserving identity privacy [1]. In addition, some projects may need to be completed together by multiple research institutions, and data may need to be shared among different groups, but each institution usually has its own encryption and access control mechanism, thus, data sharing for multiple groups presents some challenges.

First, the identity privacy of users is an urgent issue to be considered. On the one hand, they must be authenticated by the cloud to access the data. Without privacy preservation, the cloud may collect their identity information. On the other hand, if the identity privacy of users is unconditionally protected, accountability is difficult when they upload maliciously faulty shared data.

Second, the multi-group access control is a thorny problem. To preserve data privacy, data owners usually encrypt their data, and then upload the ciphertexts into the cloud [2]. It is necessary to consider not only the access rights management within a group, but also the access control of sharing among multiple groups. Since each group has its own encryption mechanism, when users want to access the data of other groups, they need to request the decryption key from the data owner or key manager of the group, which undoubtedly increases the communication cost and calculation cost, so it is not feasible in the real scene. Furthermore, group members are often dynamically changed. Therefore, it is necessary to ensure that the revoked users can no longer access any data in the cloud.

To solve the above challenges, we propose a multi-group data sharing scheme with privacy protection in the cloud. The main contributions of this paper are as follows:

*1) Privacy protecting multi-group data sharing:*Users can access the cloud anonymously, and only the group manager can reveal their real identity. Furthermore, users do not need to submit their identity for authentication when sharing data across groups .

2) Flexible and efficient access control: a flexible access control framework is built by ciphertext-policy attribute-based encryption (CP-ABE) and broadcast encryption, supporting both intra-group and cross-group data sharing, reducing key management overhead for data owners and providing efficient user revocation.

3) Security guarantee and performance analysis: We prove the security and evaluate the performance of the proposed scheme in different phases, and compare our scheme with others in terms of the computation overhead of user revocation. The experiment result shows that our scheme is efficient.

II. RELATED WORK

Some schemes were proposed to ensure secure data sharing [1],[3]-[7]. In these encryption systems, the data owner encrypts the data and specifies authorized visitors who can access the encrypted data. In [4], the cloud converted ciphertext from one encryption system to another encryption system via proxy re-encryption to realize the data sharing among different encryption systems. However, an important issue is the privacy of user identity. The data visitor needs to submit the identity to obtain the decryption key. In addition, the data owner needs to stay online to authorize visitors of other groups.

Liu et al. [1] proposed a multi-owner group data sharing scheme, in which identity-based dynamic broadcast encryption (IBBE) achieves flexible access control, the group administrators are responsible for key management of all users. with respect to the group signature technology, it protects user identity privacy and realizes anonymous and tractable data sharing. However, the scheme requires the

identity of all data visitors and is usually only applicable to data sharing within a group. Shen et al. [8] proposed an anonymous traceable group data sharing scheme, which generates shared session keys through multi-person key negotiation to reduce the burden of key management brought by centralized distribution to the central controller, but it can only realize data sharing within the same group.

III. PRELIMINARIES

A. Group Signature

The concept of group signatures was first proposed in [9] in 1991 by David Chaum and Eugene van Heyst. Any member of the group can anonymously sign a message or file on behalf of the group. Furthermore, the group manager can open any anonymous group signature to obtain the identity of the member. The short group signature scheme in [10] will be used to achieve anonymous data sharing and effective user accountability.

B. Ciphertext-Policy Attribute-Based Encryption

The first CP-ABE scheme was proposed in [11]. The CP-ABE scheme enables a private key is generated according to a set of attributes, and the ciphertext is encrypted based on an access policy. The data owner does not need to know the complete list of users, encrypts the file based on an access policy, and users can decrypt the file when his attributes meet the ciphertext access policy. The CP-ABE scheme in [12] will be used to implement efficient cross-group data sharing.

C. Broadcast Encryption

Broadcast Encryption is a public key cryptography primitive for one-to-many communications [13]. Broadcaster encrypts data and transmits these to all users so that only authorized users can decrypt. The manager can dynamically grant new members access to previously encrypted data, there is no need to update all user decryption keys and file encryption keys when modifying member permissions. We have improved the broadcast encryption scheme proposed in [14] to implement efficient intra-group data sharing.

IV. SYSTEM MODEL AND DESIGEN GOALS

A. 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. The description of the four entities is as follows:

Cloud: verifies the identity of each visitor and provides cloud storage and data sharing services for legitimate users. We assume that the cloud is trustworthy but honest, meaning that the cloud does not intentionally modify or delete uploaded files, but is curious about what is stored in the cloud and the identity of its users.

Key Generation Center: is a key authority that generates public and secret parameters for CP-ABE. KGC is a trusted party that responsible for issuing attribute keys for groups ,which means that all members of a group have the same attributes.

Group Manager: is responsible for user registration, user revocation by key distribution and revealing the real identity of a user. In our scheme, the group manager is fully trusted by group members, KGC and the cloud.

Group User: is the member of the group, every authorized user could store their data into the cloud and want to share data with others.

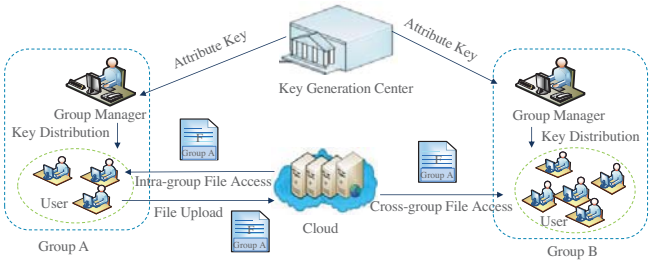

Fig. 1. System Model

B. Security Goals

1) Data confidentiality: Data confidentiality requires that unauthorized users cannot know the content of data stored in the cloud. In addition, the revoked user can no longer decrypt the ciphertext.

2) Conditional privacy protection: The identity of the user cannot be acquired by the cloud and other users who are not in the trust domain. In addition, if users upload incorrect data, the group manager can trace the identity of the group member.

3) Access control: Users within the same group should be guaranteed to directly share data with each other; users from the different groups can access across-group data after obtaining authorization. Revoked group members and unauthorized cross-group users cannot access the data.

V. THE PROPOSED SCHEME

To realize data sharing of security and privacy protection for multiple groups in the cloud, we combine group signature, attribute-based encryption of ciphertext policy and broadcast encryption technology to realize conditional privacy protection and efficient access control. The group manager is responsible for key management within the group, and the key involved in cross-group data sharing is generated by the key generation center. The data owner does not have to manage all the visitors and be online all the time to authorize them.

A. 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 $(ID_{group}, S, \sigma_s)$ to the KGC, where ID_{group} stands for group identification. KGC runs algorithm $Verify(m, \sigma_s, GPK)$ 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

a) First, the user randomly chooses a symmetric encryption key K to encrypt the file M and get the ciphertext C_K.

b) The algorithm $BroEnc(K, SK)$ and $AttEnc(K, SK, \Gamma)$ are respectively used to encrypt the key K to obtain the ciphertext *BCT* and *ACT*.

c) 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 I.

TABLE I. FILE FORMAT FOR UPLOADING DATA

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

d) Uploading F to the cloud. The cloud runs $Revocation\ Verification(PXK, GPK, \sigma)$ to verify if the user has been revoked. If not revoked, file F will be successfully uploaded to the cloud.

4) File Access

The user sends the message $(ID_{group}, ID_{data}, \lambda_i(i \in (\mathrm{RL}), \sigma_{data})$ 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 $Open(PK, GMK, \sigma)$ to get A_i, and then reveals the identity of the signer by looking up the group user list.

B. Algorithm

The algorithms *Sign*, *Verify* and *Open* used in our scheme are the same as those in the group signature scheme [10] . The algorithms *AttEnc* and *AttDecrypt* are the same as *Encrypt* and *Decrypt* in the revocable CP-ABE [12] . We only focus on the specific construction of other algorithms used in our scheme.

1) AttSetup: Key generation center selects an asymmetric pairinge:$G_0 \times G_1 \to G_2$, two elements $g_0 \in G_0, g_1 \in G_1, \alpha, \beta \in Z_p$ and the random polynomial P of degree t (the maximum number of revoked users), then computs g_0^{β}, $e(g_0, g_1)^{\alpha}$. The public system parameter *PK* and the master key *MSK* of the KGC are as follows:

$$PK = (G_0, G_1, g_0, g_1, g_0^{\beta}, e(g_0, g_1)^{\alpha}),\ MSK = \beta, g_1^{\alpha}$$

2) GroupSetup: The group manager selects $h \in G_0$ and $a, b, \gamma \in Z_p$, computs $g_1^b, g_1^{b^2}, e(g_0, g_1)^a, g_0^{\gamma}, u = a^{-1} \cdot h$ and $v = b^{-1} \cdot h$ such that $u^a = v^b = h$. The public group parameter *GPK* and the master key *GMK* of the manager are as follows:

$$GPK = (G_0, G_1, g_0, g_1, g_1^b, g_1^{b^2}, g_0^{\gamma}, u, v, e(g_0, g_1)^a),$$

$$GMK = (a, b, \gamma, P)$$

3) AttKeyGen(PK, MSK): KGC selects the random $r \in Z_p$, and selects $r_j \in Z_p$ for each $j \in S$. The attributes key *AttKey* is as follows:

$$AttKey = \left(D = g_1^{\frac{\alpha + r}{\beta}}, D' = g_1^{r},\ \forall j \in S: D_j = H(j)^{r_j}, D_j' = g_0^{r_j}\right)$$

4) UserKeyGen(GPK, GMK, ID_i): Assuming that the identity of the user i is ID_i, the number of revoked users is t. The group manager selects a random polynomial P of degree t, and selects $l, \mu \in Z_p$, calculates the following values:

$$K_{sig} = \left(A_i = g_1^{\frac{1}{\gamma + \mu}}, \mu\right)$$

$$K_{mem} = \left(K_0 = g_1^a g_1^{b^2 l P(0)}, K_1 = g_0^{l \cdot P(ID_i)}, K_2 = g_0^l,\right.$$
$$\left.\forall j \in S: KD_j = D' \cdot D_j'^{P(0)}, KD_j' = \left(D_j'\right)^{P(ID_i)}\right)$$

Finally, the user key is *SK*:

$$SK = (D, K_{sig}, K_{mem})$$

5) BroEnc(M,SK): Picking a random number $s \in Z_p$, calculate ciphertext *BCT* as:

$$\widehat{C} = Me\left(g_0, g_1\right)^{as},\ \widehat{C}' = g_0^{s},\ C' = g_1^{b^2 \cdot s}$$

6) RevocationVerification(PXK, GPK, σ): Computing $ver = e(h^{\delta_1 + \delta_2}, g_1)$, and judging if the following equations are true.

$$\forall i \in \{1, \cdots, t\}: e\left(T_3 / A_i, g_1^x\right) = ver$$

If all the equations are false, the output is true.

7) ProxykeyGen(Ciphertext,PXK): If the user accesses data within the group and the user is not revoked, the cloud uses

the proxy key PXK to calculate the latest partial decrypted key *BK* for the user:

$$BK = (\forall i \in \{1,\cdots,t\}: C'' = (C')^{\sum_{i=1}^{t} \lambda_i P(ID_i)})$$

If the user accesses the across-group data, the cloud calculates the latest partial decryption key *AK* for the user using the proxy key *PXK*:

$$AK = (\forall y \in Y: C_y'' = (C_y'')^{\sum_{i=1}^{t} \lambda_i P(ID_i)})$$

8) BroadcastDecrypt(*SK*, *BCT*)*:* Computing:

$$\frac{e(\widehat{C}', K_0)}{e(K_1, C')^{\lambda_i} \cdot e(K_2, C'')} = e(g_0, g_1)^{as}$$

Finally, the plaintext *M* can be recovered from *C*".

VI. Analysis of Our Proposed Scheme

A. Security Analysis

In this section, we prove the security of our scheme in terms of data confidentiality, conditional privacy protection, and access control that are defined in section IV.

Theorem 1: The proposed scheme protects data confidentiality.

Proof: Theorem 1 can be deduced from the following two lemmas:

Lemma 1.1: The cloud server is unable to learn the content of the stored files.

Suppose that the cloud can decrypt the ciphertext *BCT*, i.e., given $\widehat{C}' = g_0^{\ s}$, $C' = g_1^{\ b^2 \cdot s}$, public parameters g_0, g_1, $g_1^b, g_1^{b^2}$, $e(g_0, g_1)^a$ for unknown *a*, computing $e(g_0, g_1)^{as}$. This contradicts with the CDH assumption. Similarly, if the cloud wanted to decrypt the *ACT*, it would not be computationally feasible, which inherits the data confidentiality of CP-ABE in [11].

Lemma 1.2: Even under the collusion with revoked users, the cloud is also unable to learn the content of the stored files.

Suppose the user has been revoked, but the cloud still sends part of the decryption key to the user, that is, the revoked user can get *BK* and *AK*. According to data sharing [15], only has $t+1$ different shares $P(x_0), \cdots, P(x_t)$, the user can recover $P(0)$. However, one of the *t* polynomials in *BK* or *AK* must be the same as $P(ID_i)$ contained in KD_j'.

Theorem 2: The proposed scheme supports privacy preserving and traceability.

Proof: The demonstration of Theorem 2 can be derived from the following two lemmas:

Lemma 2.1: Signatures do not reveal their signer's identity, and the group manager can obtain the user's real identity.

Proof: Given the group signature σ, attackers cannot reveal the real identity of the signer without private tuple (*a*, *b*). That is, the attacker cannot calculate A_i. For more detailed proofs about anonymity and traceability of group signature, refer to [10].

Lemma 2.2: When user access cross-group file, the owner or the manager of owner's group cannot know the real identity of the user.

Proof: During the key distribution phase, the group manager converts the attribute key *AttKey* to ($\forall j \in S$: $KD_j = D' \cdot D_j^{P(0)}$, $KD_j' = (D_j')^{P(ID_j)}$) and distribute to user, then send *PXK* to cloud. Therefore, when users access files across groups, if their attributes of the keys meet the access policy and they are not revoked, the ciphertext can be decrypted. There is no need to provide identity.

Theorem 3: Based on CP-ABE and broadcast encryption technique, our scheme can achieve efficient access control.

Proof: When users access the intra-group file, they can directly decrypt the file if they are non-revoked, based on the correctness of the broadcast decryption:

$$\frac{e(\widehat{C}', K_0)}{e(K_1, C')^{\lambda_i} \cdot e(K_2, C'')}$$

$$= \frac{e\left(g_0^{\ s}, g_1^a g_1^{b^2 lP(0)}\right)}{e\left(g_0^{\ lP(ID_j)}, g_1^{\ b^2 \cdot s}\right)^{\lambda_i} \cdot e\left(g_0^l, \left(g_1^{\ b^2 \cdot s}\right)^{\sum_{j=1}^{t} \lambda_j P(ID_j)}\right)}$$

$$= \frac{e(g_0, g_1)^{as} e(g_0, g_1)^{sb^2 lP(0)}}{e(g_0, g_1)^{sb^2 l \cdot P(ID_j)\, \lambda_i} \cdot e(g_0, g_1)^{sb^2 l \cdot \sum_{j=1}^{t} \lambda_j P(ID_j)}}$$

$$= e(g_0, g_1)^{as}$$

For cross-groups data sharing, data owners specify an access policy for ciphertext, only the visitor's group properties satisfy the access strategy and has not been revoked, who can decrypt files. Detailed proof process reference [11] .

B. Performance Analysis

1) Theoretical analysis

We summarize the computation overhead of each operation at each entity side in Table II. 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.

In file access stage, if the user access within the group file only need to decrypt the ciphertext *BCT*, while users access across-group files need according to the access structure in ciphertext *ACT*, there are n leaf nodes in the access tree Γ of the ciphertext, the computation overhead associated with n linearly. In the user revocation phase, the group manager only needs to calculate the proxy key for the cloud, the computational overhead requires only some point multiplication operations.

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 II. 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: $nt_{exp2} + (3n + 2)t_{par}$	

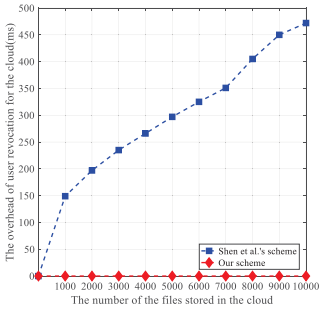

Fig. 2. Computation overhead of user revocation on the cloud side.

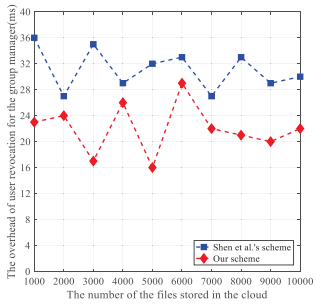

Fig. 3. Computation overhead of user revocation on the group manager side.

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]. We assume that the size of all ciphertexts is 1KB, and set the length of the RSA public key in Shen et al.'s scheme to 1024-bit. In Shen et al.'s scheme, the overhead of the cloud is linear with the number of files stored in the cloud. In order to revoke group users, the cloud computes re-encrypting RSA ciphertext $Cipher_C^* = (Cipher_C)^{e^*/e}$ for all files stored in cloud. In contrast, the cloud in our scheme does not need any operations.

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. In Shen et al.'s scheme, the overhead of user revocation on the group manager comes from the re-encrypting key generation, which includes some exponentiations and a pairing. The total overhead is about 32ms. In our scheme, the overhead of user revocation on the group manager side comes from proxy key generation, i.e. computing $P(ID_i)$, which only needs some point multiplication operations costs about 21ms. Therefore, our proposed scheme achieves high revocation efficiency on both the cloud side and the group manager side.

VII. CONCLUSION

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.

VIII. ACKNOWLEDGMENTS

The research work was supported by the Youth Program of National Natural Science Foundation of China under grant No.62002139, the Project funded by China Postdoctoral Science Foundation under grant No.2019M651738 and the National Natural Science Foundation of China under grant U1736216.

REFERENCES

[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.

[4] H. Deng et al., "Identity-Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3168-3180, 2020.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable,and Fine-Grained Data Access Control in Cloud Computing,"Proc. IEEE INFOCOM, pp. 534-542, 2010.

[6] D. Zheng, B. Qin, Y. Li and A. Tian, "Cloud-Assisted Attribute-Based Data Sharing with Efficient User Revocation in the Internet of Things," in IEEE Wireless Communications, vol. 27, no. 3, pp. 18-23, June 2020.

[7] Y. Zhang, A. Wu, D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," J. Ambient Intell. Humaniz. Comput. vol. 9, no. 4, pp.1039-1048, August 2018.

[8] J. Shen, T. Zhou, X. Chen, J. Li and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 4, pp. 912-925, April 2018.

[9] D. Chaum and E. van Heyst, "Group Signatures," Proc. Int'l Conf.Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 257-265, 1991.

[10] D. Boneh, B. Lynn, and H. Shacham, "Short Signature from the Weil Pairing, " Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, pp. 514-532, 2001.

[11] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," 2007 IEEE Symposium on Security and Privacy (SP '07), Berkeley, CA, pp. 321-334, 2007.

[12] S. Jahid, P. Mittal, and N Borisov, "EASiER: Encryption-based Access Control in Social Networks with Efficient Revocation," Proceedings of the 6th International Symposium on Information, Computer and Communications Security(ASIACCS), pp: 411-415, 2011.

[13] A. Fiat and M. Naor, "Broadcast Encryption," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 480-491, 1993.

[14] A. Lewko,A Sahai, and B Waters, "Revocation Systems with Very Small Private Keys," 2010 IEEE Symposium on Security and Privacy, pp. 273-285, 2010.

[15] A. Shamir, "How to Share a Secret," Communications of the ACM, pp. 612-613, 1979.