ISSN:0975-9646

Chaitra C et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4608-4614

Protected Data Sharing scheme with Multi-Ownership for Non Static Groups in the Cloud

Chaitra C[1], Dr. T. R. Mahesh[2]

[1]*PG Student, Dept. of Computer Science & Engineering*
T. John Institute of Technology, Bangalore, Karnataka, India.
[2]*Professor& Head, Dept. of Computer Science & Engineering*
T. John Institute of Technology, Bangalore, Karnataka, India

***Abstract*— Cloud computing offers an economical and efficient solution for sharing group resources among cloud users i.e. the individual can now run the application from anywhere in the world, as the server provides the processing power to the application and the server is also connected to a network via Internet or other connection platforms to be accessed from anywhere and with the character of low maintenance. But, due to frequent change of the membership, sharing data provided with multi-ownership, while protecting data and identity privacy of users from an untrusted cloud is a challenging issue. To overcome this problem a protected multi-owner data sharing scheme will be proposed for non-static groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others without revealing identity to untrusted cloud and the data owner will be given with the option of traceability to reveal the identity of any user at time of disputes. In the interim, the storage overhead and encryption computation cost of the proposed scheme are independent with the number of revoked users.**

***Keywords*— Cloud Computing, Data Sharing, Access Control, Non Static groups, Group Signature, Dynamic Broadcast Encryption.**

INTRODUCTION

Cloud computing is recognized as best alternative to traditional information technology [1] due to its resource sharing and low-maintenance characteristics. In cloud computing, the cloud service providers, such as Amazon, are able to deliver various services to cloud users with the help of powerful datacenters. By migrating from local data management systems into cloud servers, users can enjoy high-quality services and save significant investments on their local infrastructures.

One of the most fundamental services offered by cloud providers is data storage. However, it also poses a significant risk to the confidentiality of the stored files. Specifically, the cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and confidential, such as business plans. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud [2]. Unfortunately, designing an efficient and secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues.

First, identity privacy is one of the most significant obstacles for the wide deployment of cloud computing. Without the guarantee of identity privacy, users may be unwilling to join in cloud computing systems because their real identities could be easily disclosed to cloud providers and attackers. On the other hand, unconditional identity privacy may incur the abuse of privacy. For example, a misbehaved staff can deceive others in the company by sharing false files without being traceable. Therefore, traceability, which enables the group manager (e.g., a company manager) to reveal the real identity of a user, is also highly desirable.

Second, it is highly recommended that any member in a group should be able to fully enjoy the data storing and sharing services provided by the cloud, which is defined as the multiple-owner manner. Compared with the single-owner manner [3], multiple-owner manner is more flexible in practical applications. More concretely, each user in the group is able to not only read data, but also modify his/ her part of data in the entire data file shared by the company.

Lastly groups are normally dynamic in practice, e.g., new staff participation and current employee revocation in a company. The changes of membership make secure data sharing extremely difficult. On one hand, the anonymous system challenges new granted users to learn the content of data files stored before their participation, because it is impossible for new granted users to contact with anonymous data owners, and obtain the corresponding decryption keys. On the other hand, an efficient membership revocation mechanism without updating the secret keys of the remaining users is also desired to minimize the complexity of key management.

Several security schemes for data sharing on untrusted servers have been proposed [4-6]. In these approaches, data owners store the encrypted data files in untrusted storage and distribute the corresponding decryption keys only to authorized users. Thus, unauthorized users as well as storage servers cannot learn the content of the data files because they have no knowledge of the decryption keys.
However, the complexities of user participation and revocation in these schemes are linearly increasing with the number of data owners and the number of revoked users, respectively. By setting a group with a single attribute, Lu et al. [7] proposed a secure provenance scheme based on the ciphertext policy attribute-based encryption technique [8], which allows any member in a group to share data with others. However, the issue of user revocation is not

addressed in their scheme. Yu et al. [3] presented a scalable and fine-grained data access control scheme in cloud computing based on the key policy attribute-based encryption (KP-ABE) technique [9]. Unfortunately, the single-owner manner hinders the adoption of their scheme into the case, where any user is granted to store and share data.

To solve the challenges presented above, we propose a Protected Data Sharing scheme with Multi-Ownership for Non-Static Groups in the Cloud.

The main contributions of this paper include:

1. We propose a Secure Multi-ownership system for Data Sharing in Non-Static Groups in the Cloud. It implies that any user in the group can securely share data with others by the cloud.
2. Our proposed scheme is able to support dynamic groups efficiently. Specifically, new granted users can directly decrypt data files uploaded before their participation without contacting with data owners. User revocation can be easily achieved through a novel revocation list without updating the secret keys of the remaining users. The size and computation overhead of encryption are constant and independent with the number of revoked users.
3. We provide secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. Moreover, the real identities of data owners can be revealed by the group manager when disputes occur.
4. We provide rigorous security analysis, and perform extensive simulations to demonstrate the efficiency of our scheme in terms of storage and computation overhead.

I. RELATED WORK

In [5], files stored on the untrusted server include two parts: file metadata and file data. The file metadata implies the access control information including a series of encrypted key blocks, each of which is encrypted under the public key of authorized users. Thus, the size of the file metadata is proportional to the number of authorized users. The user revocation in the scheme is an intractable issue especially for large-scale sharing, since the file metadata needs to be updated. In their extension version, the NNL construction [10] is used for efficient key revocation. However, when a new user joins the group, the private key of each user in an NNL system needs to be recomputed, which may limit the application for dynamic groups. Another concern is that the computation overhead of encryption linearly increases with the sharing scale.

In [4], Kallahalla et al. proposed a cryptographic storage system that enables secure file sharing on untrusted servers, named Plutus. By dividing files into filegroups and encrypting each filegroup with a unique file-block key, the data owner can share the filegroups with others through deliver-ing the corresponding lockbox key, where the lockbox key is used to encrypt the file-block keys. However, it brings about a heavy key distribution overhead for large-scale file sharing. Additionally, the file-block key needs to be updated and distributed again for a user revocation.

Ateniese et al. [6] leveraged proxy re encryptions to secure distributed storage. Specifically, the data owner encrypts blocks of content with unique and symmetric content keys, which are further encrypted under a master public key. For access control, the server uses proxy cryptography to directly re encrypt the appropriate content key(s) from the master public key to a granted user's public key. Unfortunately, a collusion attack between the un-trusted server and any revoked malicious user can be launched, which enables them to learn the decryption keys of all the encrypted blocks.

Lu et al. [7] proposed a secure provenance scheme, which is built upon group signatures and ciphertext-policy attribute-based encryption techniques. Particularly, the system in their scheme is set with a single attribute. Each user obtains two keys after the registration: a group signature key and an attribute key. Thus, any user is able to encrypt a data file using attribute-based encryption and others in the group can decrypt the encrypted data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. However, user revocation is not supported in their scheme.

In [3], Yu et al. presented a scalable and fine-grained data access control scheme in cloud computing based on the KP-ABE technique. The data owner uses a random key to encrypt a file, where the random key is further encrypted with a set of attributes using KP-ABE. Then, the group manager assigns an access structure and the corresponding secret key to authorized users, such that a user can only decrypt a ciphertext if and only if the data file attributes satisfy the access structure. To achieve user revocation, the manager delegates tasks of data file re encryption and user secret key update to cloud servers. However, the single-owner manner may hinder the implementation of applications with the scenario, where any member in a group should be allowed to store and share data files with others.

From the above analysis, we can observe that secure sharing of data files in a multiple-owner manner for dynamic groups while preserving identity privacy from an untrusted cloud remains to be a challenging issue. In this paper, we propose a novel protocol A Protected Data Sharing scheme with Multi-Ownership for Non-Static Groups in the Cloud for secure data sharing in cloud computing. Compared with the existing works, the system offers unique features as follows:

1. Any user in the group can store and share data files with others by the cloud.
2. The encryption complexity and size of ciphertexts are independent with the number of revoked users in the system.
3. User revocation can be achieved without updating the private keys of the remaining users.
4. A new user can directly decrypt the files stored in the cloud before his participation

II. PRELIMINARIES

A. BILINEAR MAPS

Let G_1 and G_2 be an additive cyclic group and a multiplicative cyclic group of the same prime order q, respectively [11]. Let $e: G_1 X G_1 \rightarrow G_2$ denote a bilinear map constructed with the following properties:

1. *Bilinear:* For all $a,b \in Z^*q$ and $P,Q \in G1$, $e(aP,bQ) = e(P,Q)\ ab$.
2. *Nondegenerate:* There exists a point P such that $e(P,P) \neq 1$.
3. *Computable:* There is an efficient algorithm to compute $e(P,Q)$ for any $P,Q \in G1$

B. GROUP SIGNATURE

The concept of group signatures was first introduced in [15] by Chaum and van Heyst. In general, a group signature scheme allows any member of the group to sign messages while keeping the identity secret from verifiers. Besides, the designated group manager can reveal the identity of the signature's originator when a dispute occurs, which is denoted as traceability. In this paper, a variant of the short group signature scheme [12] will be used to achieve anonymous access control, as it supports efficient membership revocation.

C. DYNAMIC BROADCAST ENCRYPTION

Broadcast encryption is the main cryptographic problem of delivering encrypted content or encrypted files over a broadcast channel in such a way that only qualified or authentic users can decrypt the content. Broadcast encryption enables a broadcaster to transmit encrypted data to a set of users so that only a privileged subset of users can decrypt the data. As efficient revocation is the primary objective of broadcast encryption solutions are also referred to as revocation schemes. Besides the above characteristics, dynamic broadcast encryption also provides group manager the privilege to dynamically include new members if necessary while preserving previously computed information, i.e., user decryption keys need not be recomputed, the morphology and size of ciphertexts remain unchanged and no modification is required for the group encryption key. The first formal definition and construction of dynamic broadcast encryption are introduced based on the bilinear pairing technique in [14], which will be used as the basis for file sharing in dynamic groups

III. SYSTEM DESIGN AND GOALS

A. SYSTEM MODEL

We consider a cloud computing architecture by combining with an example that a company uses a cloud to enable its staffs in the same group or department to share files. The system model consists of three different entities: the cloud, a group manager (i.e., the company manager), and a large number of group members (i.e., staffs) as illustrated in Fig. 1.

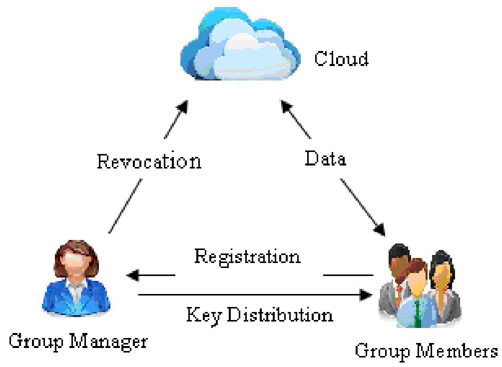

Fig.1. System model

Cloud is operated by CSPs and provides priced abundant storage services. However, the cloud is not fully trusted by users since the CSPs are very likely to be outside of the cloud users' trusted domain. Similar to [1],[3], [6], we assume that the cloud server is honest but curious. That is, the cloud server will not maliciously delete or modify user data due to the protection of data auditing schemes, but will try to learn the content of the stored data and the identities of cloud users.

Group manager takes charge of system parameters generation, user registration, user revocation, and revealing the real identity of a dispute data owner. In the given example, the group manager is acted by the administrator of the company. Therefore, we assume that the group manager is fully trusted by the other parties.

Group members are a set of registered users that will store their private data into the cloud server and share them with others in the group. In our example, the staffs play the role of group members. Note that, the group membership is dynamically changed, due to the staff resignation and new employee participation in the company.

B. DESIGN GOALS

In this section, we describe the main design goals of the proposed scheme with architectural view as in fig.2 including access control, data confidentiality, anonymity and traceability, and efficiency as follows:

Access Control: The requirement of access control is two fold. First, group members are able to use the cloud resources provided for data operations. Second, unauthorized users can never access the cloud resource any time, and revoked users will be incapable of using the cloud again once they are revoked from group.

Traceability: To tackle the inside attack problem, the group manager should have the ability to reveal the real identities of data owners in case of any disputes.

Data confidentiality: Data confidentiality requires that unauthorized users including the cloud are incapable of learning the content of the data stored in cloud. An important and challenging issue for data confidentiality is to maintain its availability for dynamic groups. Specifically, new users should decrypt the data stored in the cloud before their participation, and revoked users are unable to decrypt the data moved into the cloud after the revocation.

Anonymity: Anonymity guarantees that group members can access the cloud without revealing their real identity to anyone in group. Although anonymity represents an effective protection for user identity, it also poses a potential inside attack risk to the system. For example, an inside attacker may store and share a malicious information to get benefited.

Efficiency: The efficiency can be defined as follows: Any group member can store and share data files with other group members by the cloud. User revocation can be achieved without involving the remaining users. That means, the remaining users need not update their private keys or re encryption operations. New granted users can learn and use all the content data files stored before his participation without contacting with the data owner but only with the secret key provided by group manager can download or view files in the cloud storage.

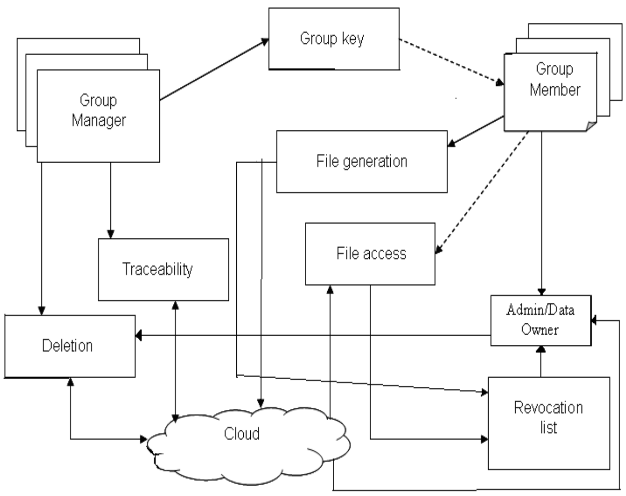

Fig.2.System architecture

IV. PROPOSED SCHEME

A. OVERVIEW

To achieve Protected Data Sharing scheme with Multi-Ownership for Non-Static Groups in the Cloud, we expect to combine the group signature and dynamic broadcast encryption techniques. Specially, the group signature scheme enables users to anonymously use the cloud resources, and the dynamic broadcast encryption technique allows data owners to securely share their data files with others including new joining users.

Unfortunately, each user has to compute revocation parameters to protect the confidentiality from the revoked users in the dynamic broadcast encryption scheme, which results in that both the computation overhead of the encryption and the size of the ciphertext increase with the number of revoked users. Thus, the heavy overhead and large ciphertext size may hinder the adoption of the broadcast encryption scheme to capacity-limited users.

To tackle this challenging issue, we let the group manager compute the revocation parameters and make the result public available by migrating them into the cloud. Such a design can significantly reduce the computation overhead of users to encrypt files and the ciphertext size. Specially, the computation overhead of users for encryption operations and the ciphertext size is constant and independent of the revocation users.

B. SCHEME DESCRIPTION

This section describes the details of the system proposed including system initialization, user registration, user revocation, file generation, file deletion, file access and traceability.

System initialization

The group manager takes charge of system initialization as follows:

- Generating a bilinear map group system.
- Selecting two random elements $_q$ along with two random numbers, and computing U and V In addition, the group manager computes H_1 and H_2.
- Randomly choosing two elements $P,G \in G_1$ and number $\gamma \in Z^*_q$.
- Publishing the system parameters including $(S,P,H,H_0,H_1,H_2,U,V,W,Y,Z,f,f_1,Enc())$, where f is one-way hash function; $f1$ is hash function and $Enc()$ is a secure symmetric encryption algorithm with secret key k.

User Registration

For the registration of user i with identity ID_i, the group manager randomly selects a number $x_i \in Z^*_q$ and computes A_i, B_i as the following equation:

$A_i=1/(\gamma+x_i)\cdot P \in G_1$ (1)

$B_i=x_i/(\gamma+x_i)\cdot G \in G_1$

Then, the group manager adds (A_i, x_i, ID_i) into the group user list, which will be used in the traceability phase. After the registration, user i obtains a private key (x_i, A_i, B_i), which will be used for group signature generation and file decryption.

User Revocation

User revocation is performed by the group manager via a public available revocation list (RL), based on which group members can encrypt their data files and ensure the confidentiality against the revoked users. As illustrated in Table 1, the revocation list is characterized by a series of time stamps $(t1 < t2 <,\ldots,tr)$. Let ID_{group} denote the group identity. $P_1,P_2,\ldots,P_r$ *and* Z_r are calculated by the group manager with the private secret γ as follows:

$P_1=1/(\gamma+x_1)\cdot P \in G_1$ (2)

$P_2=1/(\gamma+x_1)(\gamma+x_2)\cdot P \in G_1$

$P_r=1/(\gamma+x_1)(\gamma+x_2)\cdots(\gamma+x_r)\cdot P \in G_1$

$Zr=1/(\gamma+x_1)(\gamma+x_2)\cdots(\gamma+x_r)\cdot P \in G_2$.

TABLE 1. REVOCATION LIST

ID_{group}	A_1	x_1	t_1	P_1			
	A_2	x_2	t_2	P_2			
	.	.	.	.			
	A_r	x_r	t_r	P_r	Z_r	t_{RL}	sig(RL)

Motivated by the verifiable reply mechanism in, to guarantee that users obtain the latest version of the revocation list, we let the group manger update the revocation list each day even no user has being revoked in the day. In other words, the others can verify the freshness of the revocation list from the contained current date t_{RL}. In addition, the revocation list is bounded by a signature *sig(RL)* to declare its validity. The signature is generated by the group manager with the BLS signature algorithm. Finally, the group manager migrates the revocation list into the cloud for public usage.

File Generation

To store and share a data file in the cloud, a group member performs the following operations:

1. Getting the revocation list from the cloud. In this step, the member sends the group identity ID_{group} as a request to the cloud. Then, the cloud responds the revocation list *RL* to the member.
2. Verifying the validity of the received revocation list. First, checking whether the marked date is fresh. Second, verifying the contained signature *sig(RL)*. If the revocation list is invalid, the data owner stops this scheme.
3. Encrypting the data file *M*. This encryption process can be divided into two cases according to the revocation list.
   Case 1. There is no revoked user in the revocation list.
   Case 2. There are r revoked users in the revocation list.
4. Selecting a random number *T* and computing *f(T)*. The hash value will be used for data file deletion operation.
5. Constructing the uploaded data file as shown in table 2.

Uploading the data shown in Table 2 into the cloud server and adding the ID_{data} into the local shared data list maintained by the manager. On receiving the data, the cloud first invokes Algorithm 2 to check its validity. If the algorithm returns true, the group signature is valid; otherwise, the cloud abandons the data. In addition, if several users have been revoked by the group manager, the cloud also performs revocation verification by using Algorithm 3.

TABLE 2. MESSAGE FORMAT FOR UPLOADING DATA

Group ID	Data ID	Ciphertext	Hash	Time	Signature
ID_{group}	ID_{data}	C_1,C_2,C	$f(\tau)$	t_{data}	σ

Finally, the data file will be stored in the cloud after successful group signature and revocation verifications.

File Deletion

File stored in the cloud can be deleted by either the group manager or the data owner (i.e., the member who uploaded the file into the server). To delete a file ID_{data}, the group manager computes a signature $\gamma f_1(ID_{data})$ and sends the signature along with ID_{data} to the cloud. The cloud will delete the file if the equation $e(\gamma f_1(ID_{data}),P)=e(W,\gamma f_1(ID_{data}))$ holds.

Algorithm (1). Signature generation

Input: private key *(A, x),* system parameter *(P, U, V, H, W)* and data *M*.

Output: generate a valid group signature on *M.*

Begin

Select random numbers $\alpha, \beta, \gamma_\alpha, \gamma_\beta, \gamma_x, \gamma_{\delta1}, \gamma_{\delta2} \in Z^*_q$

Set $\delta_1 = x\alpha$ and $\delta_2 = x\beta$

Computes the following values

$T_1 = \alpha \cdot U$

$T_2 = \beta \cdot V$

$T_3 = A_i+(\alpha+\beta) \cdot H$

$R_1 = \gamma_\alpha \cdot U$

$R_2 = \gamma_\beta \cdot V$

$R_3 = e(T_3,P)^{rx} e(H,W)^{-\gamma\alpha-\gamma\beta} e(H,P)^{-\gamma\delta1-\gamma\delta2}$

$R_4 = \gamma_x \cdot T_1 - \gamma_{\delta1} \cdot U$

$R_5 = \gamma_x \cdot T_2 - \gamma_{\delta2} \cdot V$

Set $c=f(M,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5)$

Construct the following numbers

$S_\alpha = \gamma_\alpha + c\alpha$

$S_\beta = \gamma_\beta + c\beta$

$S_x = \gamma_x + cx$

$S_{\delta1} = \gamma_{\delta1} + c\delta1$

$S_{\delta2} = \gamma_{\delta2} + c\delta1$

Return $\sigma = (T_1,T_2,T_{3,\ c,\ S\alpha,\ S\beta,\ Sx,\ S\delta1,\ S\delta2})$

End

Algorithm (2). Signature Verification

Input: system parameter *(P, U, V, H, W), M and a signature* $\sigma = (T_1, T_2, T_{3,\ c,\ S\alpha,\ S\beta,\ Sx,\ S\delta1,\ S\delta2})$

Output: True or False.

Begin

Compute the following values

$R_1=s_\propto \cdot U - c \cdot T_1$

$R_2=s_\beta \cdot U - c \cdot T_2$

$R_3=(e(T_3,W)/e(P,P))^c e(T_3,P)^{sx} e(H,W)^{-s\propto -s\beta1} e(H,P)^{-s\delta1 -s\delta2}$

$R_4=s_x \cdot T_1 - s_{\delta1} \cdot U$

$R_5= s_x \cdot T_2 - s_{\delta2} \cdot V$

If $c=f(M,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5)$

Return True

Else

Return False

End

Algorithm (3). Revocation Verification

Input: system parameter *(H_0, H_1, H_2), a group signatureσ, and a set of revocation keys $A_1,\ldots,A_\gamma$*

Output: Valid or Invalid.

Begin

Set $temp=e(T_1,H_1)e(T_2,H_2)$

For *i=1 to n*

If $e(T_3-A_i, H_0)=temp$

Return Valid

End if

End for

Return Invalid

end

In addition, proposed system also allows data owners to delete their files stored in the cloud.

File Access

To learn the content of a shared file, a member does the following actions:

1. Getting the data file and the details of revocation list from the cloud server.
2. Checking the validity of the revocation list. This operation is similar to the step 2 of file generation phase.
3. Validity verification of the file and decrypting it. The format of the downloaded file coincides with that given in Table 2. The operation can be divided further into three cases according to time stamp t_{data} and revocation list details. If there are r revoked users in the list.

Case 1($t_{data}<t_1$): There is no revoked user before the file is uploaded.

- Invoke algorithm 2 to check the group signature σ. If algorithm returns false, user will stop this protocol.
- Using his Partial private key (A,B) to compute $\hat{K}=e(C_1,A)e(C_2,B)$.
- Decrypting ciphertext C with the computed key $\hat{K}$.

Case 2($t_i<t_{data}<t_{i+1}$): Indicates that i revoked users have been revoked before the data file is been uploaded to the storage.

- Verifying group signature σ by algorithm 2.
- Input $A_1, A_2, \ldots A_i$ to call algorithm 3. If it returns invalid, user terminates this operation.
- Computes value $A_{i,r}$ by using algorithm 4 with input $(A,x),(P_1,x_1),\ldots,(P_i,x_i)$.
- Calculating the decryption key $\hat{K}=e(C_1,A_{i,r})e(C_2,B)$.
- Decrypt the ciphertext C with key $\hat{K}$.

Case 3($t_r<t_{data}$): Indicates that r revoked users have been revoked before data file is been uploaded.

- Verifying group signature σ by algorithm 2.
- Input $A_1, A_2, \ldots A_i$ to call algorithm 3. If it returns invalid, user terminates this operation.
- Computes value $A_{r,r}$ by using algorithm 4 with input $(A,x),(P_1,x_1),\ldots,(P_r,x_r)$.
- Calculating the decryption key $\hat{K}=e(C_1,A_{r,r})e(C_2,B)$.
- Decrypt the ciphertext C with key $\hat{K}$.

Traceability

When a data dispute occurs, the tracing operation is performed by the group manager to identify the real identity of the data owner. Given a signature, the group manager employs his private key to compute *Ai.* Given the parameter *Ai,* the group manager can look up the user list to find the corresponding identity.

Algorithm (4). Parameters Computing

Input: The revoked user parameters $(P_1,x_1),\ldots,(P_\gamma, x_\gamma)$, *and the private key (A,x).*
Output: $A_{\gamma,\gamma}$ or NULL

```
Begin
        Set temp=A
        For λ=1 to γ
            If x=xλ
                return NULL
            else
                set temp=1/(x-xλ)*(Pλ-temp)
        return temp
end
```

By the analysis above, we conclude that the proposed scheme Protected Data Sharing scheme with Multi-Ownership for Non Static Groups in the Cloud achieves the security goals including access control, data confidentiality as well as anonymity and traceability.

V. Proposed Scheme

In this section, we first analyse the storage cost of proposed scheme, without loss of generality, we set q=160 and the elements in G1 and G2 to be 161 and 1,024 bit, respectively. In addition, we assume the size of the data identity is 16 bits, which yield a group capacity of 2^{16} data files. Similarly, the size of user and group identity are also set as 16 bits.

Group manager: In Proposed scheme, the master private key of the group manager is $(G, \gamma, \xi_1, \xi_2) \in G_1 \times Z_q^3$ Additionally, the user list and the shared data list should be stored at the group manager. Considering an actual system with 200 users and assuming that each user share 50 files in average, the total storage of the group manager is (80.125+42.125*200+ 2*10000)*10-3≈28.5 Kbytes, which is very acceptable.

Group members: Essentially, each user in our scheme only needs to store its private key $(A_i, B_i, x_i) \in G_1^2 \times + Z_q$, which is about 60 bytes. It is worth noting that there is a tradeoff between the storage and the computation overhead.

The extra storage overhead in the cloud: In Proposed scheme, the format of files stored in the cloud is shown in Table 2. Since C_3 is the ciphertext of the file under the symmetrical encryption, the extra storage overhead to store the file is about 248 bytes.

VI. Simulation Methodology

To study the performance, we will simulate proposed scheme by using JAVA programming language with windows 7 operating system and MYSQL database. The simulation consists of three components: client side, manager side as well as cloud side. Both client-side and manager-side processes will be conducted on a system and the cloud-side process will be implemented on another system that is equipped with core Pentium IV 2.4 GHz.

VII. CONCLUSION

In this paper, we design A Protected Data Sharing scheme with Multi-Ownership for Non Static Groups in the Cloud. In this system, a user is able to share data with others in the group without revealing identity privacy to the cloud. Additionally, it supports efficient user revocation and new user joining. More specially, efficient user revocation can be achieved through a public revocation list without updating the private keys of the remaining users, and new users can directly decrypt files stored in the cloud before their participation. Moreover, the storage overhead and the encryption computation cost are constant satisfying the desired security requirements and guarantees efficiency as well.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing,” Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[2] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. Int’l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.

[3] 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.

[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable Secure File Sharing on Untrusted Storage,” Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing Remote Untrusted Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[7] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing,” Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[8] B. Waters, “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,” Proc. Int’l Conf. Practice and Theory in Public Key Cryptography Conf. Public Key Cryptography, http://eprint.iacr.org/2008/290.pdf, 2008.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 89-98, 2006.

[10] D. Naor, M. Naor, and J.B. Lotspiech, “Revocation and Tracing Schemes for Stateless Receivers,” Proc. Ann. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[11] D. Boneh and M. Franklin, “Identity-Based Encryption from the Weil Pairing,” Proc. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 213-229, 2001.

[12] D. Boneh, X. Boyen, and H. Shacham, “Short Group Signature,” Proc. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-55, 2004.

[13] D. Boneh, X. Boyen, and E. Goh, “Hierarchical Identity Based Encryption with Constant Size Ciphertext,” Proc. Ann. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 440-456, 2005.

[14] C. Delerablee, P. Paillier, and D. Pointcheval, “Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys,” Proc. First Int’l Conf. Pairing-Based Cryptography, pp. 39-59, 2007.

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