DATE OF ACCEPTANCE: MAY 10, 2014
DATE OF PUBLICATION: MAY 17, 2014
ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

EFFICIENT USER REVOCATION AND MULTIOWNER DATASHARING SCHEME FOR CLOUD BASED DYNAMIC GROUPS

G. L. SARANYA
M.E. SCHOLAR
DEPT OF COMPUTER SCIENCE, ARASU ENGINEERING COLLEGE, KUMBAKONAM
Email: sweetsarancse@gmail.com

ABSTRACT

Cloud computing provides an economical and efficient solution for sharing group resource among cloud users. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue. In this paper, we propose a secure multi- owner data sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation cost of our scheme are independent with the number of revoked users.Groupsignature will provide security to cloud users.We are providing blocked list,, revoked list to the cloud. In addition, we analyze the security of our scheme with rigorous proofs, and demonstrate the efficiency of our scheme in experiments.

KEYWORDS- *Cloud computing, data sharing, privacy-preserving, access control, Group Signature, dynamic groups.*

1. INTRODUCTION

CLOUD computing is recognized as an alternative to traditional information technology [1] due to its intrinsic resource-sharing and low-maintenance characteristics.In cloud computing, the cloud service providers (CSPs), such as Amazon, are able to deliver various services to cloud users with the help of powerful datacenters. By migrating the 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

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

.First, identity privacy is one of the most significant obstacles for the wide deployment of cloud computing.Without the guarantee of identity privacy. Compared with the single-owner manner [3], where only the group manager can store and modify data in the cloud, the 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. Last but not least, groups are normally dynamic in practice, e.g., new staff participation and current employee revocation in a company. The changes membership makes 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. 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 etal. [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.

1.1 OUR CONTRIBUTIONS

To solve the challenges presented above, we propose Mona, a secure multi-owner datasharing scheme for dynamic groups in the cloud. The main contributions of this paper include:

1. We propose a secure multi-owner data sharing scheme. It implies that any user in the group can securely share data with others by the untrustedcloud.

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.

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

3. We provide secure and privacy-preserving access control to users, which guarantees any member in a group 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.

2. RELATED WORKS

In [4], Kallahalla et al. proposed a cryptographic storage system that enables secure file sharing on untrusted servers, named Plutus. By dividing files into file groups and encrypting each file group with a unique file-block key, the data owner can share the file groups with others through delivering 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. 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.Ateniese et al. [6] leveraged proxy 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 untrusted server and

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

any revoked malicious user can be launched, which enables them to learn the decryption keys of all the encrypted 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 delegate's 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 how to securely share 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 Mona protocol for secure data sharing in cloud computing. Compared with the existing works, Mona 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 cipher texts 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.

3. PRELIMINARIES

3.1 BILINEAR MAPS

Let G1 and G2 be an additive cyclic group and a multiplicative cyclic group of the same prime order q, respectively [11]. Let e: G1 *G1 -≥G2 denote a bilinear map constructed with the following properties:

1. Bilinear: For all $a, b \in Z_q^*$ and $P, Q \in G_1$, $e(aP, bQ) = e(P, Q)^{ab}$.
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 $P, Q \in G_1$.

3.2 Complexity Assumptions

Definition 1 (q-strong Diffie-Hellman (q-SDH) Assumption [12]). *Given $(P_1, P_2, \gamma P_2, \gamma^2 P_2, \ldots, \gamma^q P_2)$, it is infeasible to compute $\frac{1}{\gamma+x} P_1$, where $x \in Z_q^*$.*

Definition 2 (Decision linear (DL) Assumption [12]). *Given $P_1, P_2, P_3, aP_1, bP_2, cP_3$, it is infeasible to decide whether $a + b = c \bmod q$.*

Definition 3 (Weak Bilinear Diffie-Hellman Exponent (WBDHE) Assumption [13]). *For unknown $a \in Z_q^*$, given $Y, aY, a^2Y, .., a^lY, P \in G_1$, it is infeasible to compute $e(Y, P)^{\frac{1}{a}}$.*

Definition 4 ((t,n)-general Diffie-Hellman Exponent (GDHE) Assumption [14]). *Let $f(X) = \Pi_{i=1}^{r}(X + x_i)$ and $g(X) = \Pi_{i=1}^{n-r}(X + x_i')$ be the two random univariate polynomials. For unknown $k, \gamma \in Z_q^*$, given*

$G_0, \gamma G_0, ..., \gamma^{t-1} G_0, \gamma f(\gamma) G_0, P_0, ..., \gamma^{t-1} P_0, kg(\gamma) H_0 \in G_1$ and $e(G_0, H_0)^{f^2(\gamma) g(\gamma)} \in G_2$,

it is infeasible to compute $e(G_0, H_0)^{kf(\gamma)g(\gamma)} \in G_2$.

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

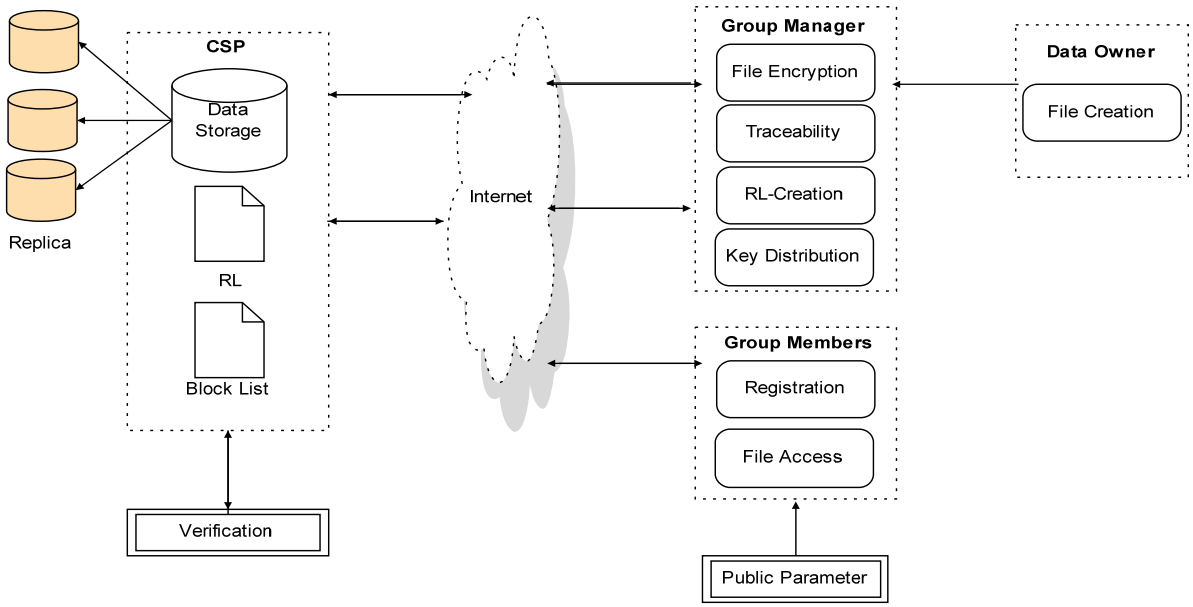
3.2 GROUP SIGNATURE

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 denoted. 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.

3.3 DYNAMIC BROADCAST ENCRYPTION

Broadcast encryption [16] enables a broadcaster to transmit encrypted data to a set of users so that only a privileged subset of users can decrypt the data. Besides the above characteristics, dynamic broadcast encryption also allows the group manager to dynamically include new members while preserving previously computed information, i.e., user decryption keys need not be recomputed, the morphology and size of cipher texts are unchanged and the group encryption key requires no modification.

4. SYSTEM ARCHITECTURE



ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

4.1 SYSTEM MODEL AND DESIGN GOALS

4.1.1 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., the staffs) as illustrated in Fig. 1.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 [3], [7], 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 [17], [18], but will try to learn the content of the stored data and the identities of cloud users .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.

4.1.2 DESIGN GOALS

In this section, we describe the main design goals of the proposed scheme including access control, data confidentiality, anonymity and traceability, and efficiency as follows:

- **Access control:** The requirement of access control is twofold. First, group members are able to use the cloud resource for data operations.
- **Data confidentiality:** Data confidentiality requires that unauthorized users including the cloud are incapable of learning the content of the stored data. An important and challenging issue for data confidentiality is to maintain its availability for dynamic groups.
- **Anonymity and traceability**: Anonymity guarantees that group members can access the cloud without revealing the real identity. Although anonymity represents an effective protection for user identity, it also poses a potential inside attack risk to the system.

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

- **Efficiency:** The efficiency is defined as follows: Any group member can store and share data files with others in the group by the cloud .User revocation can be achieved without involving the remaining.

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}	$sig(RL)$

4.2 THE PROPOSED SCHEME: MONA OVERVIEW

To achieve secure data sharing for dynamic 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.

4.2.1 Scheme Description

This section describes the details of Mona 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 H1,H2,H3 along with two random numbers q , and computing
- Randomly choosing two elements P;G 2 G1 and a number and computing respectively.
- Publishing the system parameters including S;P;H;H0;H1;H2; U; V ;W; Y ;Z; f; f1,

where f is a one-way hash function: f0 f1 is hash function.

function: f0; 1g,G1; and EnckðÞ is a secure symmetric encryption algorithm with secret key k.In the end, the parameter ð_; _1; _2;GÞ will be kept secret as the master key of the group manager.

- **User Registration:** For the registration of user i with identity IDi, the group manager randomly select a number q and computes. Then, the group manager adds into the group user list, which will be used in the traceability phase. After the registration, user i obtains a private key

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

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 ðRLÞ, 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 <; :::; tr). Let ID group denote the group identity. The tuple ðAi; xi; tiÞ represents that user i with the partial private key (Ai; xi) is revoked at time ti. P1; P2; :::; Pr and Zr are calculated by the group manager with the private secret key. 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.

$$\begin{cases} P_1 = \frac{1}{\gamma + x_1} \cdot P \in G_1 \\ P_2 = \frac{1}{(\gamma + x_1)(\gamma + x_2)} \cdot P \in G_1 \\ P_r = \frac{1}{(\gamma + x_1)(\gamma + x_2) \cdots (\gamma + x_r)} \cdot P \in G_1 \\ Z_r = Z^{\overline{(\gamma + x_1)(\gamma + x_2) \cdots (\gamma + x_r)}} \in G_2. \end{cases}$$

➢ **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. If the revocation list is invalid, the data owner stops this Revocation.

Case 1. There is no revoked user in the revocation list:

i. Selecting a unique data file identity ID data;

ii. Choosing a random number k 2 Z _q ;

iii. Computing the parameters C1; C2; K; C

➢ **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 _f1ðIDdataÞ and sends the signature along with ID data to the cloud. The cloud will delete the file. Traceability When a data dispute occurs, the tracing operation is performed by the group manager to identify the real identity of the data

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

owner. Given a signature a1,a2 T3; c; s_; s_; sx; s_1; s_2 Þ, the group manager employs his private key ð_1; _2Þ to compute Ai ¼ T3 _ ð_1 _ T1 þ _2 _ T2Þ. Given the parameter Ai, the group manager can look up the user list to find the corresponding identity.

- **Algorithm**

Parameters Computing

Input: The revoked user parameters $(P_1, x_1), \ldots, (P_r, x_r)$,

and the private key.

Output: Ar;r or NULL

begin

set temp ═ A

for1 to r

if x ═x

return NULL

else

set temp ¼ 1

x—x

temp

return temp

end

- **File Access:**

To learn the content of a shared file, a member does the following actions:1. Getting the data file and the revocation list from the cloud server. In this operation, the user first adopts its private key A; x Þ to compute a signature _u on the message ID group; ID data; tÞ by using Algorithm 1, where t denote the current time, and the ID data can be obtained from the local shared file list maintained by the manager. Then, the user sends a data request containing ID group; ID data; t; _up to the cloud server. Upon receiving the request, the cloud server employs Algorithm 2 to check the validity of the signature and performs revocation verification.

- **Traceability:**

When a data dispute occurs, the tracing operation will be performed by the group manager. The cloud the real identity of the data owner. Given a signature _ ¼ ðT1; T2;T3; c; s_; s_; sx; s_1; s_2 Þ, the group manager employs his private key .

5. PERFORMANCE EVALUATION

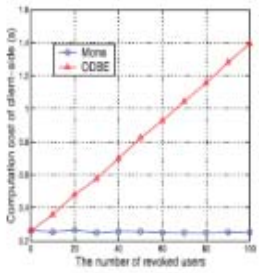
In this section, we first analyze the storage cost of Mona, and then perform experiments to test its computation cost.

5.1 STORAGE:

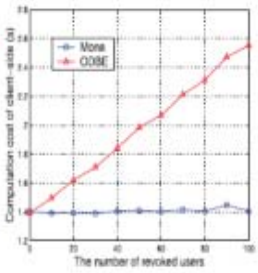
Without loss of generality, we set q ═160 and the elements in G1 and G2 to be 161 and 1,024 bit, respectively. In addition,

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

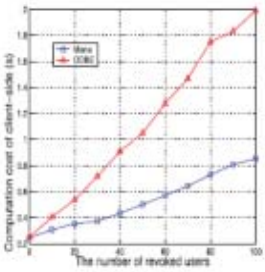
we assume the size of the data identity is 16 bits, which yield a group capacity of 216 data files. Similarly, the size of user and group identity are also set as 16 bits. Group manager. In Mona, the master private key of the group manager is G1,G2,G3 Additionally, the user list and the shared data list 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 .



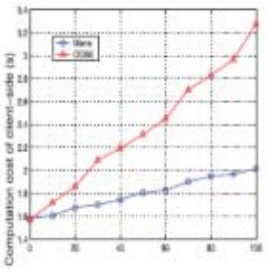
(a) Generating a 10 MB file



(b) Generating a 100 MB file



(a) Accessing a 10 MB file



(b) Accessing a 100 MB file

6. CONCLUSION

In this paper, we design a secure data sharing scheme, Mona, for dynamic groups in an untrusted cloud. In Mona, a user is able to share data with others in the group without revealing identity privacy to the cloud.

Additionally, Mona 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.

Computation Cost of the Cloud (s)

Request	The number of revoked users		
	0	50	100
File generation (100 MB)	0.065	0.154	0.271
File generation (10 MB)	0.045	0.125	0.226
File access (100 MB)	0.045	0.150	0.237
File access (10 MB)	0.045	0.151	0.240
File deletion (100 MB)	0.041	0.153	0.240
File deletion (10 MB)	0.042	0.156	0.238

Cloud Computation Cost

To evaluate the performance of the cloud in Mona, we test its computation cost to respond various client operation requests including file generation, file access, and file deletion. Assuming the sizes of requested files are 100 and 10 MB, the test results are given in Table 3. It can be seen that the computation cost of the cloud is deemed acceptable, even when the number of revoked users is large. This is because the cloud only involves group signature and revocation verifications to ensure the validity of the requestor for all

ISSN: 2348-4098
VOLUME 02 ISSUE 04 APRIL-MAY 2014

operations. In addition, it is worth noting that the computation cost is independent with the size of the requested file for access and deletion operations, since the size of signed message is constant, e.g., ðID group; IDdata; tÞ in file access and ðID data; _Þ in file deletion requests.

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 SecuritySymp. (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 Cloud Computing," , pp. 282-292, 2010Proc. ACM Symp. Information, Computer and Comm. Security pp. 29-43, 2006