American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-5, Issue-2, pp-254-264
www.ajer.org
Research Paper Open Access

Secure and Reliable Sharing of Multi-Owner Data for Dynamic Groups in the Cloud

B.Nalini[1], Prof.Dr.D.J.Pete[2]
[1](Electronics and Telecommunications, Datta Meghe College of Engineering/ University of Mumbai, India)
[2](Electronics, Datta Meghe College of Engineering/ University of Mumbai, India)

ABSTRACT : Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the internet. It provides an economical and efficient solution for sharing group resource among cloud users. Data sharing in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue because of dynamic nature of multi-owner group. The solution of this issue is addressed in recently best efficient scheme MONA: secure multi-owner data sharing for dynamic groups in the cloud but it has some limitations. In this paper we are further extending the basic MONA by adding the reliability by increasing the number of group managers dynamically and also we are using efficient revocation for dynamic groups by leveraging group signature and dynamic broadcast encryption techniques and for overall security HMAC_SHA1 algorithm. Meanwhile, the storage overhead and encryption computation cost are independent with the number of revoked users.

Keywords -Cloud computing, data sharing, dynamic groups, identity privacy, multi-owner.

I. INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services, scalable services like data sharing, accessing etc., over the web on demand basis. It uses the web and central remote servers to maintain data and applications. 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 the migration of local data management systems into cloud servers, users can enjoy high-quality services and save significant investments on their local infrastructures. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with web access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Cloud computing is broken down into three segments: "application" "storage" and "connectivity". Each segment serves a different purpose and offers different products for businesses and individuals around the world.

Cloud computing is one of the greatest platform which provides storage of data in very lower cost and available for all time over the internet. Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and devices on demand. Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. Cloud Computing means more than simply saving on IT implementation costs. One of the most fundamental services offered by cloud providers is data storage. A company allows its staffs in the same group or department to store and share files in the cloud. By utilizing the cloud, the staffs can be completely released from the troublesome local data storage and maintenance. However, it also poses a significant risk to the confidentiality of those 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. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud. 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, multiple-owner manner 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. 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.

Third, 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 [8], [9], and [10]. 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. [5] proposed a secure provenance scheme based on the cipher text-policy attribute-based encryption technique, 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. [4] presented a scalable and fine-grained data access control scheme in cloud computing based on the key policy attribute based encryption (KP-ABE) technique [7]. This scheme unfortunately, the single owner manner hinders the adoption of their scheme into the case, where any user is granted to store and share data.

The solution of the above challenges presented in recently best efficient scheme Mona, a secure multi-owner data sharing scheme for dynamic groups in the cloud. The main contributions of Mona:

1. In Mona, any user in the group can securely share data with others by the untrusted cloud
2. It 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. It provided 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.

But it has some limitations. We further extending the basic MONA by adding the reliability as well as improving the scalability by increasing the number of group managers dynamically.

The remainder of this paper is organized as follows: Section II overviews the related work. In Section III, we describe the system model and our design goals. In Section IV, we presented some modules and algorithm used in implementation of the system. In Section V, shows some results of our implemented system. Finally, we conclude the paper in Section VI.

II. RELATED WORK

M. Armbrust et al. [2] presented a security one of the most often-cited objections to cloud computing; analysts and skeptical companies ask “who would trust their essential data, out there‟ somewhere?” There are also requirements for auditability, in the sense of Sarbanes-Oxley azon spying on the contents of virtual machine memory; it’s easy to imagine a hard disk being disposed of without being wiped, or a permissions bug making data visible improperly. There’s an obvious defense, namely user-level encryption of storage. This is already common for high-value data outside the cloud, and both tools and expertise are readily available. This approach was successfully used by TC3, a healthcare company with access to sensitive patient records and healthcare claims, when moving their HIPAA-compliant application to AWS. Similarly, auditability could be added as an additional layer beyond the reach of the virtualized guest OS, providing facilities arguably more secure than those built into the applications themselves and centralizing the software responsibilities related to confidentiality and auditability into a single logical layer. Such a new feature reinforces the Cloud Computing perspective of changing our focus from specific hardware to the virtualized capabilities being provided

S. Kamara et al. [3] proposed a security for customers to store and share their sensitive data in the cryptographic cloud storage. It provides a basic encryption and decryption for providing the security. However, the revocation operation is a sure performance killer in the cryptographic access control system. To optimize the revocation procedure, they present a new efficient revocation scheme which is efficient, secure, and unassisted. In this scheme, the original data are first divided into a number of slices, and then published to the cloud storage. When a revocation occurs, the data owner needs only to retrieve one slice, and re-encrypt and republish it. Thus, the revocation process is accelerated by affecting only one slice instead of the whole data.

They have applied the efficient revocation scheme to the cipher text-policy attribute-based encryption based cryptographic cloud storage. The security analysis shows that the scheme is computationally secure.

In [4] 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 cipher-text 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.

Lu et al. [5] proposed a secure provenance scheme, which is built upon group signatures and cipher-text-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.

Ateniese et al. [6] leveraged proxy reencryptions 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.

In [9], 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.

E. Goh et al. [10] presented a SiRiUS, a secure file system designed to be layered over insecure network and P2P file systems such as NFS, CIFS, Ocean Store, and Yahoo! Briefcase. SiRiUS assumes the network storage is untrusted and provides its own read-write cryptographic access control for file level sharing. Key management and revocation is simple with minimal out-of-band communication. File system freshness guarantees are supported by SiRiUS using hash tree constructions. SiRiUS contains a novel method of performing file random access in a cryptographic file system without the use of a block server. Extensions to SiRiUS include large scale group sharing using the NNL key revocation construction. Our implementation of SiRiUS performs Ill relative to the underlying file system despite using cryptographic operations. SiRiUS contains a novel method of performing file random access in a cryptographic file system without the use of a block server. Using cryptographic operations implementation of SiRiUS is also possible. It only uses the own read write cryptographic access control. File level sharing are only done by using cryptographic access.

In [10] 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 [11] 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.

D. Boneh et al.[12] focused on a Hierarchical Identity Based Encryption (HIBE) system where the cipher-text consists of just three group elements and decryption requires only two bilinear map computations, regardless of the hierarchy depth. Encryption is as efficient as in other HIBE systems. They prove that the scheme is selective-ID secure in the standard model and fully secure in the random oracle model. The system has a number of applications: it gives very efficient forward secure public key and identity based cryptosystems (with short cipher-texts), it converts the NNL broadcast encryption system into an efficient public key broadcast system, and it provides an efficient mechanism for encrypting to the future. The system also supports limited delegation where users can be given restricted private keys that only allow delegation to bounded depth. The HIBE system can be modified to support sub linear size private keys at the cost of some cipher-text expansion.

A.Fiat et al.[16] proposed a system on multicast communication framework, various types of security threat occurs. As a result construction of secure group communication that protects users from intrusion and eavesdropping are very important. In this Dissertation (First Stage), they propose an efficient key distribution method for a secure group communication over multicast communication framework. In this method, they use IP multicast mechanism to shortest rekeying time to minimize adverse effect on communication. In addition, they introduce proxy mechanism for replies from group members to the group manager to reduce traffic generated by rekeying. They define a new type of batching technique for rekeying in which new key is generated for both leaving and joining member. The rekeying assumption waits for 30 sec so that number time's key generation will be reduced.

III. SYEM MODEL & DESIGN GOALS

To achieve a secure and reliable multi-owner data sharing for dynamic groups in the cloud, in this paper we are presenting the system model, which keeps the sharing of multi-owner data is secure and reliable. In this method we are further presenting how we are managing the risks like failure of group manager by increasing the number of backup group manager, hanging of group manager in case number of requests more by sharing the workload in multiple group managers. This method claims required efficiency, scalability and most importantly reliability. We 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.

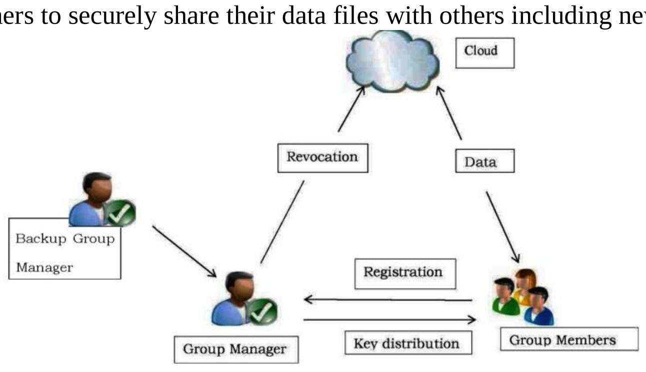


Figure 1: System Model

3.1 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 their files. The system model consists of four different entities: the cloud, a group manager (i.e., the company manager), backup group manager and a large number of group members (i.e., the staffs) as illustrated in Figure 1.

3.1.1 Cloud

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 , but will try to learn the content of the stored data and the identities of cloud users.

3.1.2 Group manager

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.

3.1.3 Backup group manager

Backup group manger takes charge all the responsibilities of group manger if group manger unable to perform his duties.

3.1.4 Group members

Group members are a set of registered users they 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.

3.2 Design Goals

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

3.2.1 Access Control

The requirement of access control is twofold. First, group members are able to use the cloud resource for data operations. Second, unauthorized users cannot access the cloud resource at any time, and revoked users will be incapable of using the cloud again once they are revoked.

3.2.2 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. 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.

3.2.3 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. For example, an inside attacker may store and share a mendacious information to derive substantial benefit. Thus, to tackle the inside attack, the group manager should have the ability to reveal the real identities of data owners.

3.2.4 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 users. That is, the remaining users do not need to update their private keys or reencryption operations. New granted users can learn all the content data files stored before his participation without contacting with the data owner.

IV. SYSTEM IMPLIMENTATION

4.1 Modules

4.1.1 Cloud Module:

In this module, we create a local Cloud and provide priced abundant storage services. The users can upload their data in the cloud. We develop this module, where the cloud storage can be made secure. 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 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.

4.1.2 Group Manager Module:

Group manager takes charge of followings,

1. Key generation,
2. User registration,
3. User revocation, and
4. Revealing the real identity of a dispute data owner.

Therefore, we assume that the group manager is fully trusted by the other parties. The Group manager is the admin. The group manager has the logs of each and every process in the cloud. The group manager is responsible for user registration and also user revocation too.

4.1.3 Group Member Module:

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. Note that, the group membership is dynamically changed, due to the staff resignation and new employee participation in the company. The group member has the ownership of changing the files in the group. Whoever in the group can view the files which are uploaded in their group and also modify it.

4.1.4 File Security Module:

- ➢ Encrypting the data file.
- ➢ 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).

4.1.5 Group Signature Module:

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.

4.1.6 User Revocation Module:

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.
We used HMAC_SHA-1 algorithm to implement the System for secure sharing of data between the group members.

4.2 HMAC_SHA1_ ALGORITHAM:

A keyed-hash message authentication code (**HMAC**) is an algorithm for applications requiring message authentication. Message authentication is achieved via the construction of a message authentication code (MAC). MACs based on cryptographic hash functions are known as HMACs. The purpose of a MAC is to authenticate both the source of a message and its integrity without the use of any additional mechanisms. HMACs have two functionally distinct parameters, a message input and a secret key known only to the message originator and intended receiver(s). Additional applications of keyed hash functions include their use in challenge-response identification protocols for computing responses, which are a function of both a secret key and a challenge message. An HMAC function is used by the message sender to produce a value (the MAC) that is formed by condensing the secret key and the message input. The MAC is typically sent to the message receiver along with the message. The receiver computes the MAC on the received message using the same key and HMAC function as was used by the sender, and compares the result computed with the received MAC. If the two values match, the message has been correctly received, and the receiver is assured that the sender is a member of the community of users that share the key. The HMAC specification in this standard is a generalization of HMAC as specified in Internet RFC 2104, HMAC, Keyed-Hashing for Message Authentication, and ANSI X9.71, Keyed Hash Message Authentication Code. Authenticate packets with HMAC using message digest algorithm (The default is SHA1). HMAC is commonly used message authentication algorithm (MAC) that uses a data sharing, a secure hash algorithm, and a key, to produce a digital signature.

HMAC involves hash algorithms in combination with the secret key. HMAC can be used with any of the iterative hash functions and MD5, SHA-1 and SHA256 are such examples and the resulting MAC functions are referred as HMAC-MD5, HMAC-SHA-1 and HMAC-SHA256. Security 18 of HMAC depends on the underlying hash function, the size of the output and the quality of the secret key. HMAC-SHA1 does not suffer from a collision attack in the same way as SHA-1 because in the beginning of the HMAC-SHA1 the message to hash is based on the secret key not known by the attacker. To compute HMAC the following calculation is performed.

$$\mathrm{HMAC}(K, m) = H\big((K \oplus opad) \parallel H(K \oplus ipad) \parallel m)\big) \quad (1)$$

Where

H is a cryptographic hash function,
K stands for the secret key and if the key is shorter than the block size it is then padded with zeroes or if the key is longer than the block it is then truncated.
m is the message to be authenticated,
|| denotes concatenation,
⊕ denotes exclusive or (XOR),
opad is the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant),

and **ipad** is the inner padding (0x363636…3636, one-block-long hexadecimal constant).

Figure.2shows the principle of HMAC-SHA1. HMAC functions can be used in several ways. The secret key known by the HMAC clients can be given as a parameter or the key is not known by the sender and receiver but instead, the key is hidden and the user can choose the key by giving it as a parameter. Alternative way is to keep the key totally hidden and the HMAC function takes the message as a parameter and then uses the hidden secret key in the HMAC operation.

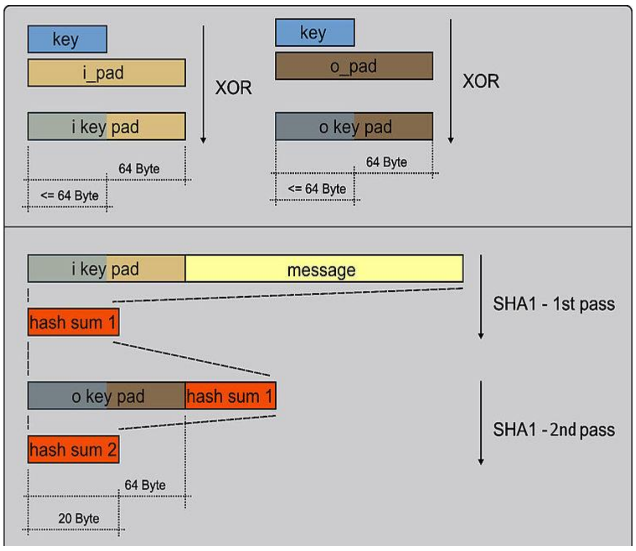


Figure2.HMAC SHA-1

HMAC SPECIFICATION
To compute a MAC over the data ‘text’ using the HMAC function, the following operation is performed:

MAC(text)t = HMAC(K, text)t = $H((K_0 \oplus opad) || H((K_0 \oplus ipad) || text))t$ (2)

A step by step process in the HMAC algorithm is illustrated below.
Step 1: If the length of K = B: set K_0 = K. Go to step 4.
Step 2: If the length of K > B: hash K to obtain an L byte string, then append (B-L) zeros to create a B-byte string K0 (i.e., K_0= H(K) || 00...00). Go to step 4.
Step 3: If the length of K < B: append zeros to the end of K to create a B-byte string K0 (e.g., if K is 20 bytes in length and B = 64, then K will be appended with 44 zero bytes 0x00).
Step 4: Exclusive-OrK_0 with ipad to produce a B-byte string: $K_0 \oplus$ipad.
Step 5: Append the stream of data 'text' to the string resulting from step 4: ($K_0 \oplus$ipad) || text.
Step 6: Apply H to the stream generated in step 5: H($K_0 \oplus$ipad) || text).
Step 7: Exclusive-Or K_0 with opad: $K_0 \oplus$opad.
Step 8: Append the result from step 6 to step 7: ($K_0 \oplus$opad) || H(($K_0 \oplus$ipad) || text).
Step 9: Apply H to the result from step 8: H(($K_0 \oplus$opad)|| H(($K_0 \oplus$ipad) || text)).
Step 10: Select the leftmost t bytes of the result of step 9 as the MAC.

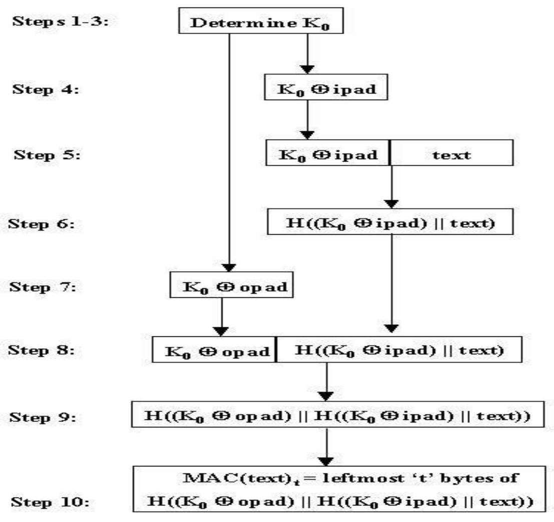
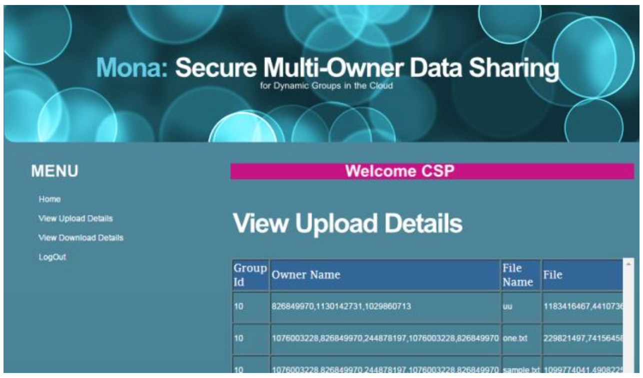

Steps 1-3:
Determine K_0
Step 4:
$K_0 \oplus$ ipad
Step 5:
$K_0 \oplus$ ipad
text
Step 6:
H(($K_0 \oplus$ ipad) || text)
Step 7:
$K_0 \oplus$ opad
Step 8:
$K_0 \oplus$ opad
H(($K_0 \oplus$ ipad) || text)
Step 9:
H(($K_0 \oplus$ opad) || H(($K_0 \oplus$ ipad) || text))
Step 10:
MAC(text)$_t$ = leftmost ‘t’ bytes of H(($K_0 \oplus$ opad) || H(($K_0 \oplus$ ipad) || text))

Figure3 Illustration the of HMAC construction

V. RESULTS

This Project is implemented using Java as programming language. Net beans usedas integrated development environment (IDE) for coding. Xamp server for combining Apache application server and MySQL for database purposes. The database stores the list of group members registered and also the revocation list.The group manger generate group key and distribute to all his group members. All group members upload their data and download data from the cloud which was uploaded by the other group member also modify their part of data and uploaded again in the cloud. Every group member can upload into the cloud and download from the cloud without revealing their real identity instead they use group signature. The following are screenshot of scheme.

Cloud is operated by Cloud Service Provider (csp) s 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, but will try to learn the content of the stored data and the identities of cloud users. The below Figure 4 shows the csp module the csp can view the dataupload in cloud and downloaded from cloud in encrypted form and data owner identity also encrypted form.

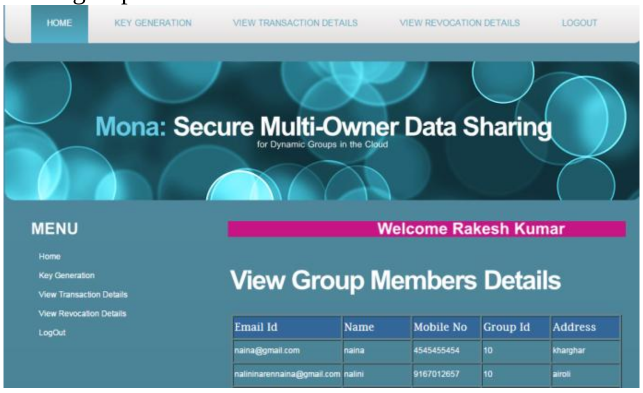

Mona: Secure Multi-Owner Data Sharing
for Dynamic Groups in the Cloud
MENU
Home
View Upload Details
View Download Details
LogOut
Welcome CSP

View Upload Details

Group Id	Owner Name	File Name	File
10	826849970,1130142731,1029860713	uu	1183416467,441073
10	1076003228,826849970,244878197,1076003228,826849970	one.txt	229821497,7415645
10	1076003228,826849970,244878197,1076003228,826849970	sample.txt	1099774041,490822

Figure 4 Upload Details of Data in Encrypted Form

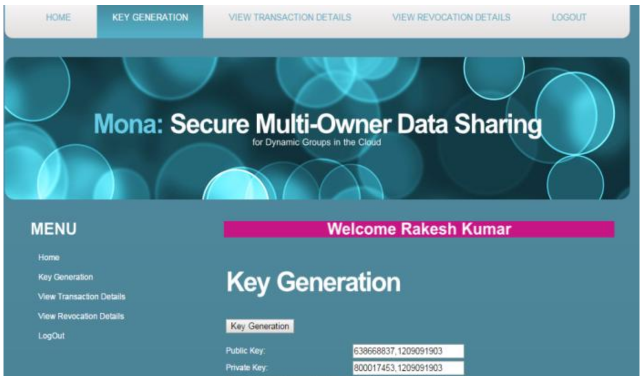
Group manager takes charge of key generation, user registration, user revocation, and revealing the real identity of a dispute data owner. In the given example, the group manager is acted by the administrator of the company. Therefore, we assume that the group manager is fully trusted by the other parties. The following figs gives some key duties of group manager as follows Figure 5 Group manager home page it gives the all the details of group members in that group.



View Group Members Details

Email Id	Name	Mobile No	Group Id	Address
naina@gmail.com	naina	4545455454	10	kharghar
nalininarennaina@gmail.com	nalini	9167012657	10	airoli

Figure 5 Group Manager Home page



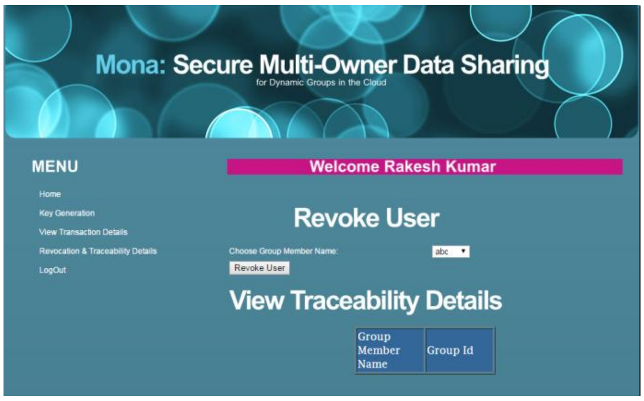
Welcome Rakesh Kumar

Public Key: 638668837,1209091903

Private Key: 800017453,1209091903

Figure 6 Key Generation

The Figure 6 gives the generation of private and public keys and distributes those keys to group members of the group. The group members can upload their files using public key distributed by group members. Group members can download using private key distributed to them.


Mona: Secure Multi-Owner Data Sharing
for Dynamic Groups in the Cloud
MENU
Home
Key Generation
View Transaction Details
Revocation & Traceability Details
LogOut
Revoke User
Revoke User
View Traceability Details

Welcome Rakesh Kumar

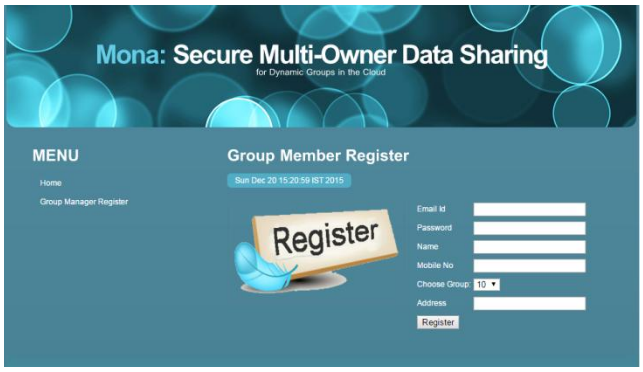
Choose Group Member Name: abc

Group Member Name	Group Id

Figure 7 User Revocation

Figure 7 shows the revocation and traceability of the group member. The revoked user can’t log into the system. If any disputes occurs among the group members on data ownership group manager will reveal the real identity of the group member i.e. traceability.

Groupmemersare the registered users in the group Figure 8 shows group members registration form the user will register in the group by filling the required information in the fields.


Mona: Secure Multi-Owner Data Sharing
for Dynamic Groups in the Cloud
MENU
Home
Group Manager Register
Group Member Register
Register
Register

Sun Dec 20 15:20:59 IST 2015

Email Id

Password

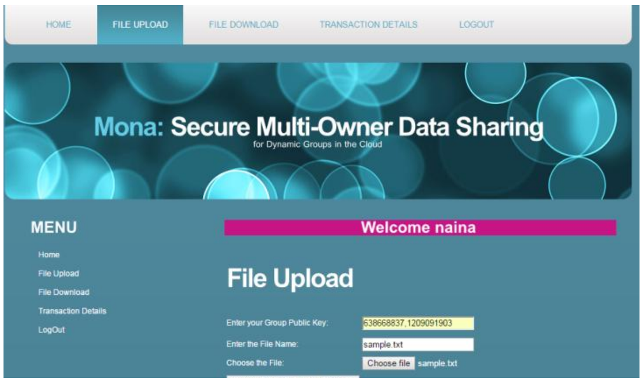
Name

Mobile No

Choose Group: 10

Address

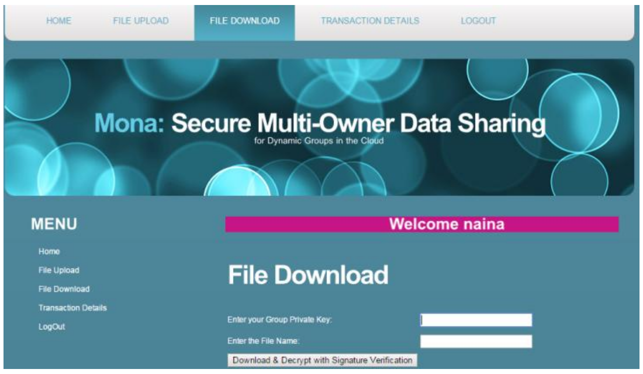
Figure 8 Group Member Registration Form



Enter your Group Public Key: 638668837,1209091903
Enter the File Name: sample.txt
Choose the File: sample.txt

Figure 9 File Upload

Group membersupload theirlocal data files into cloud using group signature. Theychoose theirfile from local system and click on encrypt & upload with signature button. The uploaded content stored in cloud in an encrypted form.



Enter your Group Private Key:
Enter the File Name:

Figure 10 File Download

Figure 10 shows the file downing activity of the .group members download file from cloud with group private key. The downloaded content available in decrypted form in our local system. If they want to do modifications and upload they can do.

VI. CONCLUSION

Cloud computing is very attractive environment for business world in term of providing required services in a very cost effective way. This project focuses on problem of secure data sharing scheme for dynamic groups in an untrusted cloud. In this scheme, a user is able to share data with others in the group without revealing identity privacy to the cloud. In this scheme, we are further presenting how we are managing the risks like failure of group manager by increasing the number of backup group manager, hanging of group manager in case number of requests more by sharing the workload in multiple group managers. This method claims required efficiency, scalability and most importantly reliability.

REFERENCES

[1] Xuefeng Liu, Yuqing Zhang, Boyang Wang, Jingbo Yan “Mona:Secure Multi-owner Data Sharing for Dynamic Groups in the Cloud,” Trans. IEEE Parallel and Distributed Systems, vol 24,No 6,June 2013.
[2] 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.
[3] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. Int’l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.

[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] C. Delerablee, P. Paillier, and D. Pointcheval, “Fully CollusionSecure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys,” Proc. First Int’l Conf. Pairing-Based Cryptography, pp. 39-59, 2007.
[15] 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.
[16] A. Fiat and M. Naor, “Broadcast Encryption,” Proc. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 480-491, 1993.