International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Efficient User Revocation in the Cloud

Supriya Gade[1], P. B. Kumbharkar[2]

[1, 2] Pune university, Department of Computer Engineering, Siddhant College Of Engineering,Sudumbare,Pune

Abstract: ***In cloud computing user can easily share and modify data inside group. Here data integrity can be easily ensured by using the public signature of existing users in the cloud group. By using data sharing services in cloud user can able modify data as a group on cloud. Integrity of these services can be improved using signature of public existing users on group. Shared data is divided into blocks. All user are responsible for modifying data on different blocks. When any user shows malicious activity in group then that user must be revoked for security purpose. So the block modified by that user must be re-sign by existing users in group. Traditional way to re-sign this is to down load part of data and re-sign it at the time of revocation of user. But this approach is not efficient for large amount of data being shared. This paper gives novel method for auditing integrity of shared data and provide effective way for user revocation. It uses the concept of proxy re-signature to avoid the downloading of re-signed block by existing user.***

Keywords: Public auditing, shared data, user revocation, cloud computing.

1. Introduction

There are many services for data storage and sharing services which shares bunch of data with each other like Google Drive. These mechanisms provides several facilities of modifying data and allow to share latest version of modified data with remaining group. The cloud service provider's issues a good quality service with sufficient security but integrity of this data can be reduced due to human errors and software or hardware failure.

For maintaining integrity in shared data space many mechanisms have been proposed. Each data block in group data is attached with signature which is attached by user responsible for modification of data. Data integrity is depend upon the correctness of signature. Here, a signature is attached to each block in data, and also the integrity of knowledge depends on the correctness of all the signatures. Safe and efficient approach to check integrity of data without downloading complete data on group. This is done by means of public verifier which is utilizes cloud data or a third party auditor (TPA) having ability of verification on integrity of data. Many existing works describes auditing on the integrity of personal knowledge. Some recent works focus on preserving identity of user from global cloud verifiers during maintaining integrity of group data. But none of existing method provide methodology for efficiency and correctness of data in cloud.

As shared Data is exported to the cloud and existing users no longer store it on native devices, simplest method to re-calculate user signature during revocation is to increase associate degree of existing user for preliminary transfer the blocks maliciously signed by the revoked user, It first examines the efficiency of these blocks, and then blocks are re-sign with uploading data on cloud. As compare to this

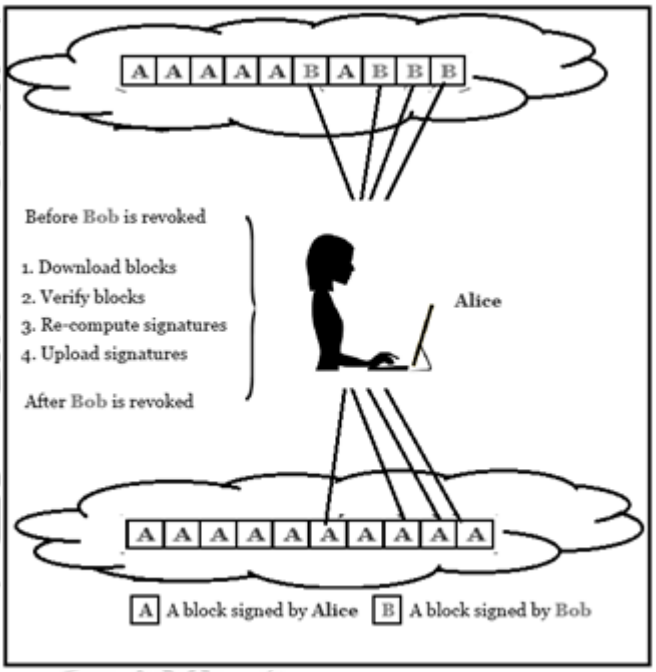

Figure 1: Straight forward Approach

existing method may have a huge amount of cost of communication and calculation resources by downloading blocks and re-examining attached signatures.

2. Literature Survey

In this Survey relative mechanisms and the methods which are employed earlier to attain a public auditing are discussed. And also the advantages and disadvantages of each technique are discussed. According to the survey of the earlier mechanism, it finds that the current system implemented has more advantages.

A. Boyang Wang, et.al.(2014),Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud;

With data storage and sharing services in the cloud, users can easily modify and share data as a group. To ensure shared data integrity can be verified publicly, users in the group need to compute signatures on all the blocks in shared data.

Different blocks in shared data are generally signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user.

B. Boyang Wang,et.al.(2014), Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud;

In this paper, They propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, they exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With those mechanisms, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, those mechanisms are able to perform multiple auditing tasks simultaneously instead of verifying them one by one and experimental results demonstrate the effectiveness and efficiency of those mechanisms when auditing shared data integrity.

C. CongWang,et.al.(2013),Privacy-Preserving Public Auditing for Secure Cloud Storage;

Using cloud storage, users can remotely store their data and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, they propose a secure cloud storage system supporting privacy-preserving public auditing. They further extend those result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Those preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

D. M. Armbrust,et.al.(2010),A View of Cloud Computing, Communications of the ACM;

cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under provisioning for one that becomes wildly popular, thus missing potential customers and revenue.

E. C.Wang,et.al.(2010),Privacy-Preserving Public Auditingfor Data Storage Security in Cloud Computing

Cloud Computing has been envisioned as the nextg eneration architecture of IT Enterprise. It moves the application software and databases to the centralized large data centres, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, they consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamic svia the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations, this paper achieves both. They first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in those protocol designs. In particular, to achieve efficient data dynamics, they improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

F. G. Ateniese,et,al.(2007),Provable Data Possession at Untrusted Stores, in the Proceedings of ACM CCS 2007

they introduce a model for provable data possession (PDP)that allows a client that has stored data at an un trusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems.

3. Proposed System

Our proposed system avoid losing money on data sharing services it may lie to examiners about the false shared data for saving the reputation of its data services. Here we consider an assumption to avoid collusion between user and the cloud.

This system consist of three modules,
User Module.
Auditor Module.
Admin Module.

User Module;
User module is responsible for following operations.

Registration: This module is responsible for user registration with details for using system and files. Only those users can login into cloud server who are registered user.

File Upload: User uploads file with blocks along with performing encryption using secrete key.

Download: In this module user can download the file and can decrypt data using his secret key.

Re-upload: Once user re-sign the blocks of file then this module allow user to re-upload the downloaded files.

Unblock: This module provide some questions and by answering these security user account will be unlocked.

Auditor Module:
Verification of Files: The public verifier is responsible for checking the scalability of shared data.

Files View: This module allow auditor to view the all details of file upload, file download, blocked user, re-upload.

Admin Module:
View Files: This module allow auditor to view the all details of file organization.

Block User: Admin have authority to block the misbehave user account.

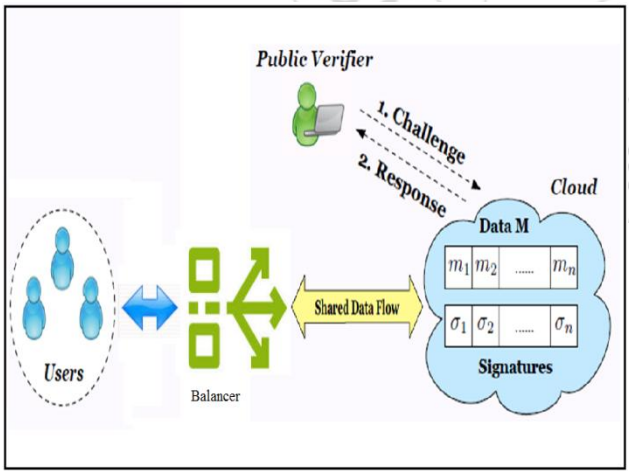

Figure 2: System Architecture

Public Verifier: It correctly check the integrity and correctness of the shared data.
User: Here user can able to share data as individual or as group.
Cloud: Cloud provides storage service for shared data.
Public Auditing: It is able to audit the integrity without gaining the complete data from the cloud.

4. Algorithm

A. Algorithm Panda:
This algorithm uses five main methods stated as follows.

1) KeyGen:
Here each user generates a random input to produce public key and private key. User is assume as creator of share data and he is responsible for key generation without violating generality.

2) ReKey:
It is a resigning key which is generated by cloud. This is generated for each user pair lies within group. Here we have assumed that existing private channels among each pair of users at the time of creation of re-signing keys, and collusion is avoided.

3) ReSign:
At the time of creation of shared data original user computes a sign for each block like signature. During modification data block is get assigned with the signature of user who is modifying that data block. After revocation of user cloud re-signs the blocks modified by revoked user using a new key known as resigning key.

4) ProofGen:
The challenge-and-response protocol verifies data integrity among the cloud and a public verifier. The cloud is responsible for creation of a proof of volume of shared data which held under the monitoring of a public verifier.

5) Proof Verify:
In this, assurance and proof sign by cloud is checked by public verifier.

B. DSA Algorithm:
DSA is form of electronic text that can be used to authenticate the identity of the transmitter of a document or the signer of a message, and verify that the original blocks of the message that has been sent is remain non-modified. It has two main steps,

Steps:
1. Key Generation.
2. Signature Creation.
3. Signature Verification.

1. Key Generation:
Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system, while the second phase computes public and private keys for a single user.
Have shared global public key values (p, q, g):
a. choose 160-bit prime number q
b. choose a large prime p with $2L-1 < p < 2L$

i. where L= 512 to 1024 bits and is a multiple of 64
ii. Such that q is a 160 bit prime divisor of (p-1)

c. choose $g = h(p-1)/q$
i. where $1<h<p-1$ and $h(p-1)/q \bmod p > 1$
d. Users choose private & compute public key

e. choose random private key: x<q
f. compute public key: y = gx mod p

2. Signature Creation:
To sign a message M the sender:
a. generates a random signature key k, k<q
b. nb. k must be random, be destroyed after use, and never be reused
c. then computes signature pair:
r = (gk mod p) mod q
s = [k-1(H (M) + xr)] mod q
d. sends signature (r,s) with message M

3. Signature Verification:
Having received M & signature (r, s)
a. To verify a signature, recipient computes:

w = s-1 mod q
u1= [H (M) w] mod q
u2= (rw) mod q
v = [(gu1 yu2) modp] mod q

b. if v=r then signature is verified
c. see Appendix A for details of proof why

5. Results

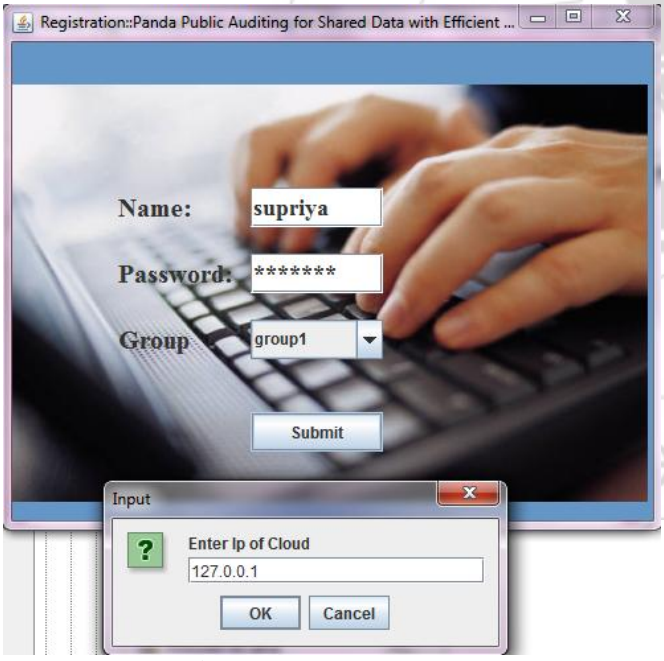

Figure 3: User Registration

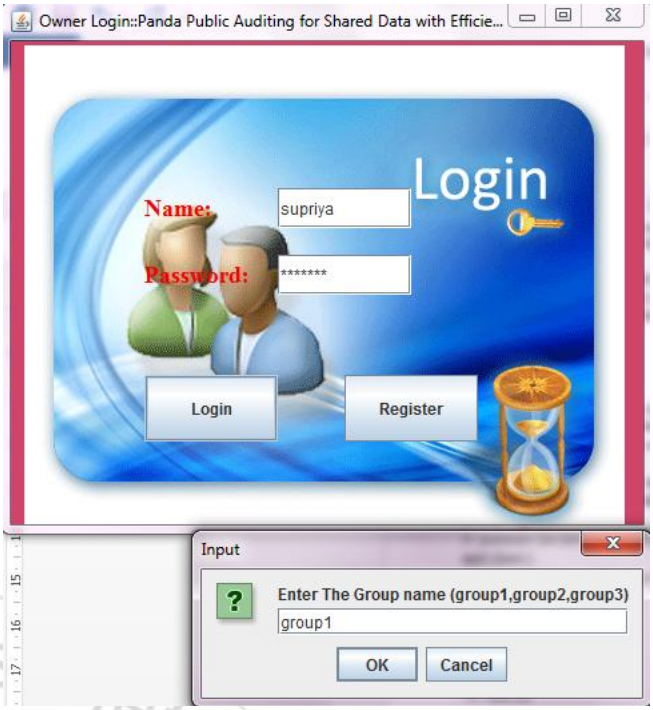

Figure 4: User Login

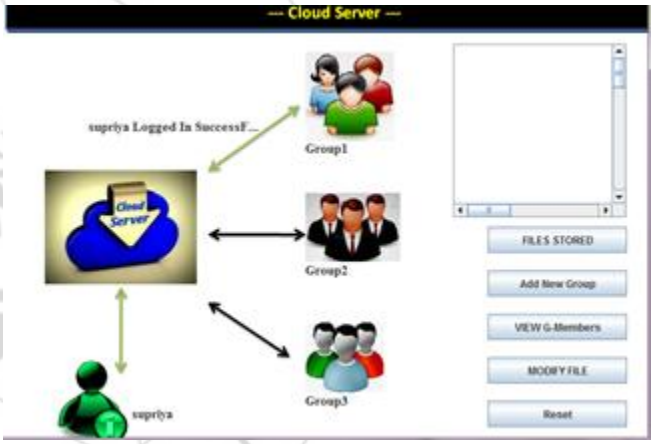

Figure 5: User login status on server

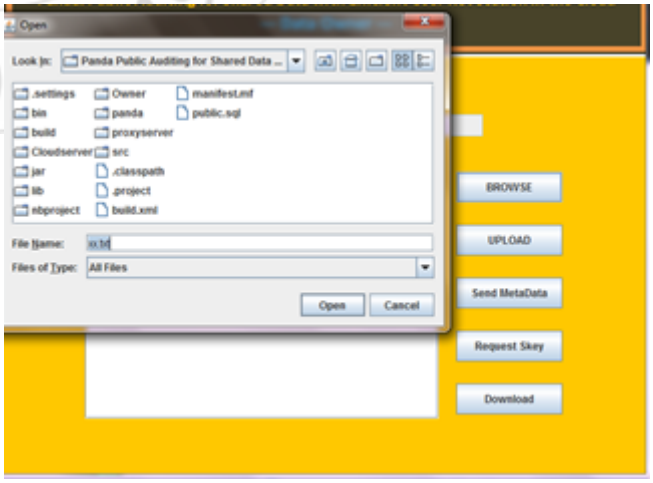

Figure 6: Browse file

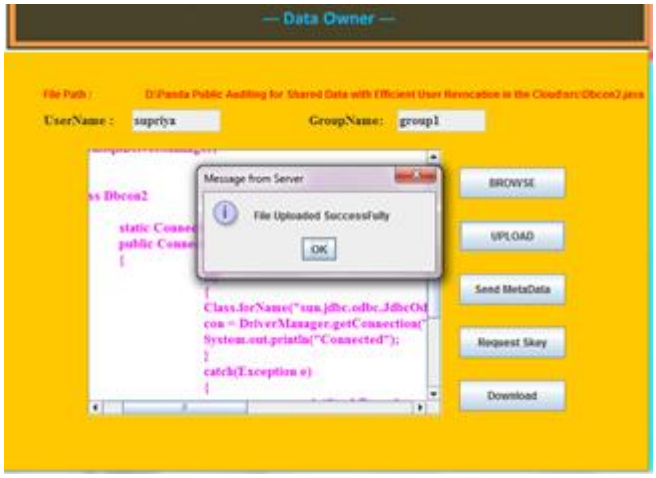

Figure 7: Upload file

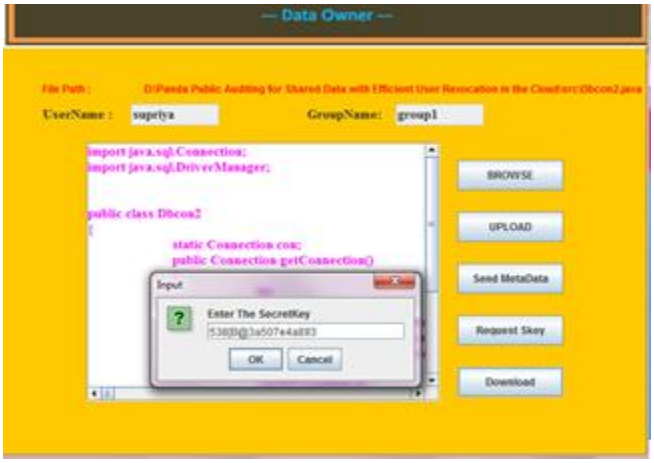

Figure 11: File download

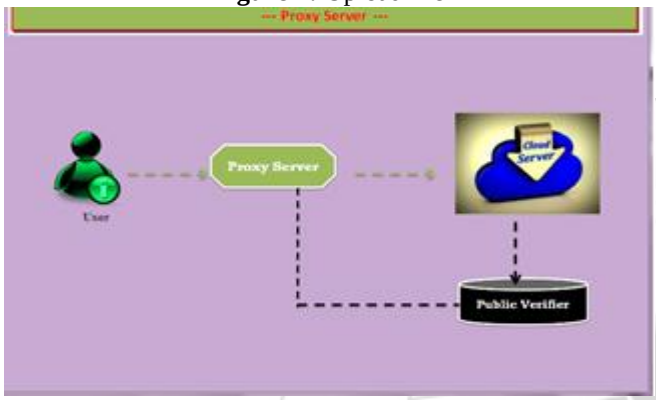

Figure 8: File Upload status on proxy server

Figure 12: View metadata

Figure 9: Send metadata

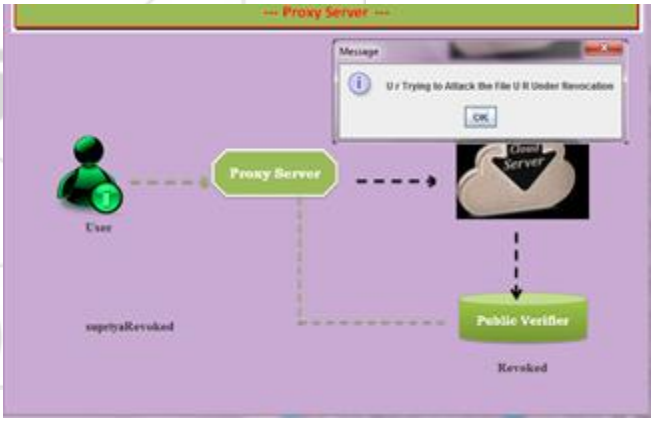

Figure 13: User revoked

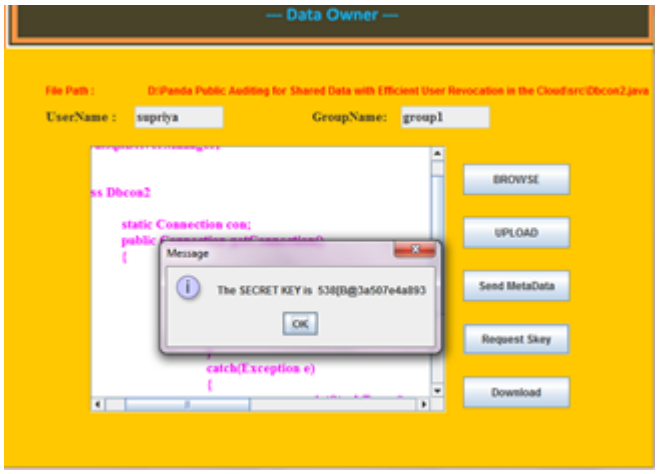

Figure 10: request secrete key

Figure 14: Revoked user on public verifier

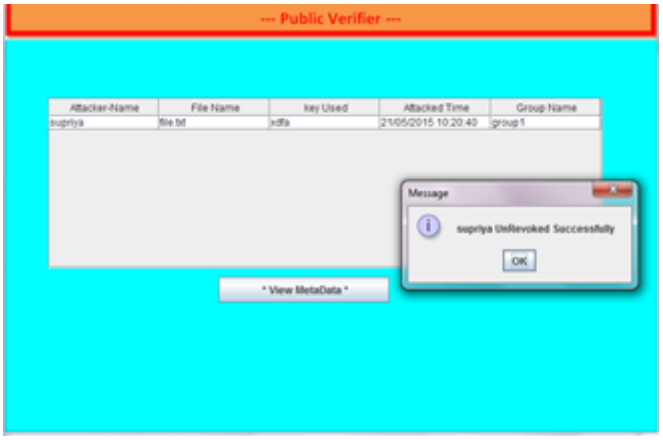

Figure 15: User unrevoked

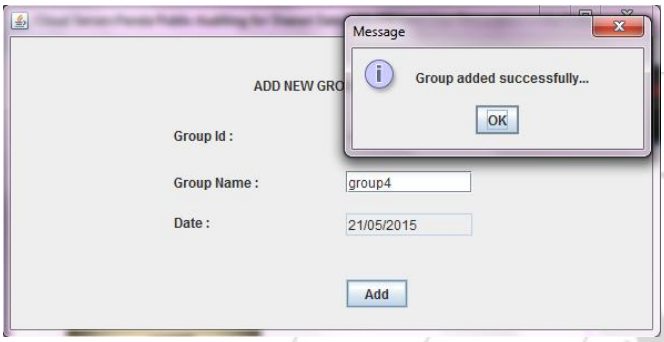

Figure 16: Add new group

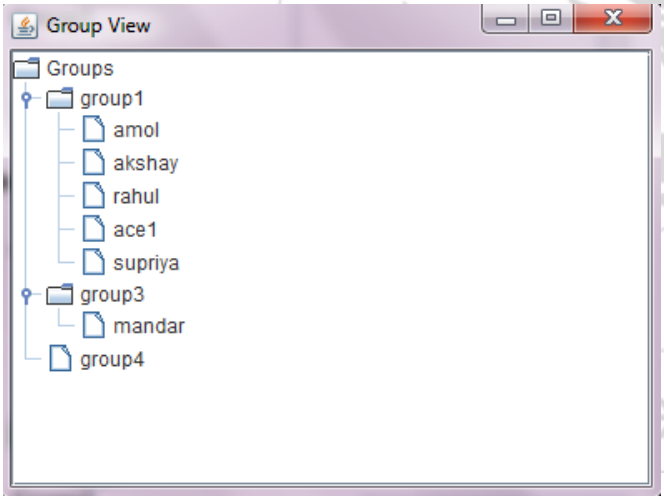

Figure 17: View group

6. Conclusion

Thus here we come to conclude that our system have an ability to generate a fully unique public auditing mechanism through revocation of economical user for integrity of shared data. Propose system aims to enable the cloud to automatically re-sign data blocks through existing users while creating the proxy re-signatures. There is no need of user to re-sign blocks manually. Public verifier is able to audit the integrity of data being shared and does not retrieve the complete data, but some part of data shared are re-signed by cloud itself. This system enables batch auditing by examining multiple tasks in synchronous way. Her we are allowing semi-trusted cloud to verify and re-sign blocks using proxy signatures at the time of user revocation.

References

[1] Boyang Wang, Baochun Li, Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee,A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ―A View of Cloud Computing, Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apirl 2010.
[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ―Provable Data Possession at Untrusted Stores, ―in the Proceedings of ACM CCS 2007, 2007, pp. 598–610.
[4] H. Shacham and B. Waters, ―Compact Proofs of Retrievability, in the Proceedings of ASIACRYPT 2008. Springer-Verlag, 2008,pp. 90–107.
[5] C. Wang, Q. Wang, K. Ren, and W. Lou, ―Ensuring Data Storage Security in Cloud Computing, in the Proceedings of ACM/IEEE IWQoS 2009, 2009, pp. 1–9.
[6] C. Wang, Q. Wang, K. Ren, and W. Lou, ―Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing, ―in the Proceedings of IEEE INFOCOM 2010, 2010, pp. 525–533.
[7] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, ―Dynamic Audit Services For Integrity Verification of Outsourced Storage in Clouds, in the Proceedings of ACM SAC 2011, 2011, pp. 1550–1557
[8] C. Wang, Q. Wang, K. Ren, and W. Lou, ―Towards Secure and Dependable Storage Services in Cloud Computing, IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2011.
[9] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and S. Chen, ―Dynamic Audit Services For Outsourced Storage in Clouds, IEEE Transactions on Services Computing, accepted.