4th International Conference on Recent Innovations in Science Engineering and Management
India International Centre, New Delhi
(ICRISEM-16)
20th March 2016, www.conferenceworld.in
ISBN: 978-81-932074-6-8

ENHANCED CENTRALIZED ACCESS CONTROL FOR USER REVOCATION

Mrs. I. MettildhaMary[1], Gayathri S[2], PavithraP[3], PavithraT.R[4], SoundariyaM[5]

[1,2,3,4,5]*Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, (India)*

ABSTRACT

To develop a web based application in an IT organization to create a decentralized network with attribute based encryption for file access control with user revocation functionalities. Data storage and sharing services in the cloud, users can easily modify and share data as a group. The data owner will have the group key to access the cloud data.in order to access the file systems the group members need to use their group key and personal keyto protect the integrity of data in the cloud, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signature.If user is revoked from the group, the block which were previously signed by this revoked user must be resigned .The key is automatically regenerated. A third party auditor is able to audit the integrity of shared data without retrieving the entire data from the cloud.

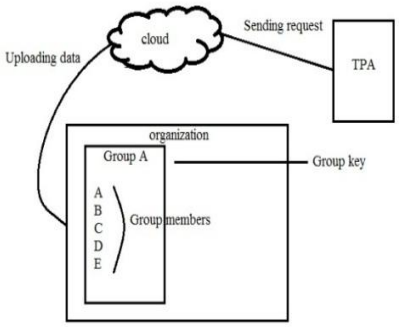
Index Terms: public auditing, shared data, and user revocation.

I.INTRODUCTION

Cloud computing has been envisioned as the next generation information technology (IT) architecture for enterprises, cloud computing is an internet-based development and use of computer technology. The public cloud environment is the IaaS/PaaS Infrastructure or Platform as a Service that we rent from Linux (IaaS) or Microsoft (PaaS). Both are enabled for web hosting. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing services. The cloud computing vendors, Amazon simple storageServices (S3) and amazon elastic compute cloud (EC2) [8].while these internet-based online services do provide huge amounts of storage space and customizable computing resources.

Cloud services providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data to protect the integrity of data in the cloud, a number of mechanisms[3],[4],[5],[6],[7],[9],[10],[11]have been proposed in this paper, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signature.

ISBN: 978-81-932074-6-8



Fig, 1. System model with cloud and group key generation

Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources.Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data. To securely introduce an effective third party auditor (TPA) should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user. Following steps are used 1) We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes.

- We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.
- We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons.

II. EXISTING SYSTEM

Without group key and personal key management many IT companies are facing various data based problems. In the existing method revocation will not be introduced. So that in case users in a group made any changes means, the admin could not able to identify the modified data.In the existing system without the group key management each user of the group will not be identified. So that they can't able to receives a update message and computes a group key by using data contained in the group key update message and its own key.

In case a user was created, they will be provided only with a use name and password and even if they allocated in a group, they will still remain with the same user name and password. Due to this method some data may easily modified by the user or the data bay be get leaked by the user. Sometimes the user may be get relieved from the group in case the admin don't wants the user's modification means the admin cant able to retrieve the data back. The user needs to do more modification in the project again to revert the data back.

III. DRAWBACKS IN THE EXISTING SYSTEM

- User provided only with user name and password, to specified key allotted.
- In case of creating group the user need to assess the data using their password only, no group key provided
- User work can't able to monitor.
- Could not able to do revocation process in case any user changed the important data or an important code.
- No security for the important information.
- User could not able to communicate with the admin

IV. PROPOSED SYSTEM

Revocation re-signatures process, allow the cloud to re-sign blocks.So existing users do not need to download and re-sign blocks by themselves. A public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud. Auditing process also able to support batch auditingby verifying multiple auditing tasks simultaneously

Each and every user will be provided with a user name and password along with the personal key. In case of creating a group, a group key will be created for the user. So that in case of accessing any data means the user need to give their personal key as well the group key for proper data retrieval.

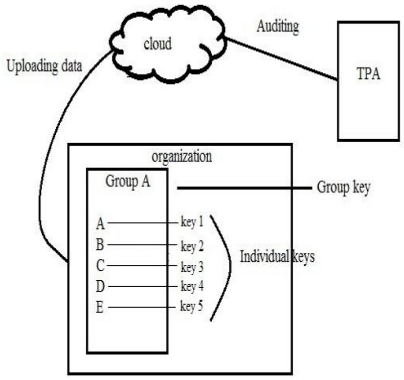


Fig. 2.Individual keys generation

In this proposed system is the user done some mischief activities like changing the code or modifying the important data, the whole document will be affected. If the admin do the revocation process for the particular user means, the data will be modified and the document will not affected by the revocated user. Also in case the admin done revocation process for a single user means the file system will generate a new group key for all the members in the group. This can be implemented using architectural design model.

V. ADVANTAGES OF THE PROPOSED SYSTEM

- User will be provided with a personal key
- Prior group key will be created will creating group for allocating work
- User can be monitored anytime
- Revocation can be done successfully, while the revocation process executing the particular user's data only will be getting changed.
- Can confirm the security at any level during the data transfer
- User can communicate with admin at anytime

VI. DESIGN GOALS

Our proposed mechanism should achieve the following properties:

- **Correctness**: the public verifier is able to correctly check the integrity of shared data.
- **Efficient and Secure User Revocation**: On one hand, once a user is revoked from the group, the blocks signed by the revoked user can be efficiently re-signed. On the other hand, only existing users in the group can generate valid signatures on shared data, and the
  Revoked user can no longer compute valid signatures on shared data.
- **Public Auditing**: The TPA can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have been re-signed by the cloud.
- **Scalability**: cloud data can be efficiently shared among a large number of users, and the public verifier is able to handle a large number of auditing tasks simultaneously and efficiently.

VII. PROXY RE-SIGNATURES

Proxy re-signatures, first proposed by Blaze et al [11] allow a semi-trustedproxy to act as a translator of signatures between two users, for example, Alice and Bob. More specifically, the proxy is able to convert a signature of Alice into a signature of Bob on the same block. Meanwhile, the proxy is not able to learn any private keys of the two users, which means it cannot sign any block on behalf of either Alice or Bob. In this paper, to improve the efficiency of user revocation, we propose to let the cloud to act as the proxy and convert signatures for users.

VIII. THEORY

8.1 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is based on a design principle known as a substitution-permutation network. It is efficient in both software and hardware. AES has a fixed block size of 128 bit and a key size of 128, 192, or 256 bit, whereas Rijndael has specified with block and key sizes in multiples of 32 bit, with a minimum of 128 bit. The block size has a maximum of 256 bit but the key size has no theoretical maximum AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger

block size have additional columns in the state). Most AES calculations are down in a special finite field.The distinctive features of this algorithm are as follows:

• Key length: the key length can be varied from 16 up to any larger value depending on the security level required.

• Word length: the block size can be varied between 1 to 16 bit and 1 to 32 and so on. That is, encryption can be performed on 16 or 32 or 64 bit blocks. This, in turn, can be used on different processor architectures employing 16, 32, or 64 bit registers.

• The algorithm, therefore, provides variable degrees of security. However, this increased security level will be at the cost of increased size of the cipher-text.

• The number of rounds is variable: the whole process can be repeated r times using the same key.

8.2 Encryption Process

The method is reasonably simple. We have a key matrix KLx2 where,

$$k_{ij} \in \{1,2,3,4,5,6,7,8\} \begin{cases} \forall i = 1,...,L\,; \mathrm{L} \geq 16 \\ \forall j = 1,2 \end{cases}$$

This key is known only to the sender and receiver. When the first party wants to send a message M to the second party, he/she determines the key 2 L K and every character from the message it is replaced by a binary value.
The aim of the algorithm is to hiding a number of bits from plain text message(M)into a random vector (v)of bits.the location of the hidden bits are determined by the key KLx2

Input:

M(plain text) and KLx2 (key array)

Algorithm Body:

First: in a plain text file, each character In sequentially replaced by its binary value

I=0

m:=first digit in M file

while(m≠EOF)

i:=I mod L

generate 8 bits randomly and set them in V vector

if(k[I,1]<k[I,2]) then

for j=k[I,1] to k[I,2]

if(m≠EOF)

then do

v[j]=m

m:=next m in M file

end do

next j

else

for j=k[I,1] down to k[I,2]

if(m≠EOF)

then do

v[j]=m

m:=next m in M file

save V in output file

i=i+1

end while

output: encrypted file

An eight-bit octet is generated randomly and set in a temporary vector V. the bits in the vector V from position K [1,1] to position K[1,2] are replaced by bits from the secret message. Then the resulting vector V is stored in a file. As long as the message file has not reached its end yet, we move to the next row of the key matrix and another octet is generated randomly and the replacement is performed repeatedly and the resulting vector is stored in the file. The previous procedure is repeated over and over again pending the end the message. The resulting file is sent to the receiver who beforehand has the key matrix. If the key length is not enough to cover the whole message during the encryption process, the key will be reapplied over and over again until the encryption of the whole message is completed.

IX. CONSTRUCTION OF PANDA

Panda includes six algorithms: KeyGen, Rekey, Sign, Resign, ProofGen, ProofVerify. In KeyGen, every user in the group generates his/her public key and private key. In Rekey, the cloud computes are-signing key for each pair of users in the group. When the original user creates shared data in the cloud, he/she computes a signature on each block as in Sign. After that, if a user in the group modifies a block in shared data, the signature on the modified block is also computed as in Sign. In Resign, a user is revoked from the group, and the cloud re-signs the blocks. In Proof-Verify, a public verifier is able to check the correctness of a proof responded by the cloud.

- **KeyGen**: This is key generation algorithm and here user generates their public and private key. The data owner creates a user list which contains Id's of all the users in the group. This user list (UL) is public.
- **Sign**: This algorithm is used for signing the block by data owner and if a user in the group modifies a block in shared data, the signature on the modified block is also computed.
- **Resign**: This algorithm is used for re-signing the blocks by cloud which were previously signed by revoked users.
- **ProofGen**: In ProofGen algorithm, cloud is able to generate proof of possession of shared data under the challenge of public verifier and this works in two parts. In first part public verifier generates audit message and send it to cloud and in second part cloud generates a proof of possession of shared data M, after receiving the auditing message.

X. SCALABILITY OF PANDA

The scalability of panda is by reducing the total number of re-signing keys in the cloud and enabling batch auditing for verifying multiple auditing tasks simultaneously. Reduce the number of re-signing keys, the cloud needs to establish and maintain a resigning key for each pair of two users in the group. Since the number of users in the group is denoted as d, the total number of re-signing keys for the group is d (d-1)/2. If the cloud data is shared by a very large number of users, e.g., d =200, then the total number of resigning keys that the cloud has to securely store and manage is 19900, which significantly increases the complexity of key management in cloud.More specifically, if the total number of users in the group is still d=200 and the size of a short PL is d= 5, which means the cloud is able to convert signatures of a revoked user only into one of these five users shown in the short PL, then the total number of re-signing keys required with the short PL of five users is 990. It is only 5 percent of the number of re-signing keys with the entire PL of all the 200 users.

XI. RELIABILITY OF PANDA

The reliability of panda is importance for the cloud is to securely store and manage the re-signing keys of the group, so that the cloud can correctly and successfully convert signatures from a revoked user to an existing user when it is necessary. In many cases, the public verifier may need to handle multiple auditing tasks in a very short time period. Therefore, to improve the scalability of public auditing mechanism, we can further extend Panda to support batch auditing [7]. With batch auditing, a public verifier can perform multiple auditing tasks simultaneously.

XII. CONCLUSIONS

In this paper, we proposed a new public auditing mechanism for shared data with efficient user revocation in the cloud. When a user in the group is revoked, we allow semi-trusted cloud to re-sign blocks by revocated user with proxy re-signature. The result show that the cloud can improve the efficiency of significant amount of computation and communication resources during user revocation.

REFERENCES

[1] B.Wang, B Li,and H.Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud,"Proc.IEEE INFOCOM, pp.2904-2912, 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," *Communications of the ACM*, vol. 53,no. 4, pp. 50–58, April 2010.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," in *theProceedings of ACM CCS 2007*, 2007, pp. 598–610.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," in *the Proceedings of ACM/IEEE IWQoS2009*, 2009, pp. 1–9

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing, "in *the Proceedings of ESORICS 2009*. Springer-Verlag, 2009, pp. 355–370.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *the Pro-ceedings 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] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," in *the Proceedings of IEEE Cloud 2012*, 2012, pp. 295–302.

[9] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "LT Codes-based Secure and Reliable Cloud Storage Service," in *the Proceedings of IEEEINFOCOM 2012*, 2012, pp. 693–701.

[10] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," in *the Proceedings ofACNS 2012*, June 2012, pp. 507–525.

[11] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Xidian University, Xi'an, China, Tech. Rep., 2012. [Online]. Available: http://ste.xidian.edu.cn/ lihui/cloud12.pdf