ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Securing Shared Data in Public Cloud with User Revocation

Ms. Yarramathi Mounika
M.Tech (CSE)
Department of CSE,
N.B.K.R Institute of Science & Technology,
Vidyanagar.

Mr. V. VeeraRaghavulu, M.Tech, [Ph.D]
Assistant Professor,
Department of CSE,
N.B.K.R Institute of Science & Technology,
Vidyanagar.

ABSTRACT

With data storage and sharing services in the cloud, users can easily modify and share data as a group. To ensure share 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. The straight forward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism. For the integrity of shared data with efficient user revocation in mind. By utilizing the idea of proxy re-signatures, we allow the cloud tore-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, 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. Moreover, our mechanism is able to support batch auditing by verifying multiple auditing tasks simultaneously. Experimental results show that our mechanism can significantly improve the efficiency of user revocation.

INTRODUCTION

WITH data storage and sharing services (such as Drop box and Google Drive) provided by the cloud, people can easily work together as a group by sharing data with each other. More specifically, once a user creates shared data in the cloud, every user in the group is able to not only access and modify shared data, but also share the latest version of the shared data with the rest of the group. Although cloud providers promise a more secure and reliable environment to the users, the integrity of data in the cloud may still be compromised, due to the existence of hardware/software failures and human errors. To protect the integrity of data in the cloud, number of mechanisms have been proposed. In these mechanisms, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signatures. One of the most significant and common features of these mechanisms is to allow a public verifier to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing (or denoted as Provable Data Possession). This public verifier could be a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third party auditor (TPA) who is able to provide verification services on data integrity to users. Most of the previous works focus on auditing the integrity of personal data. Different from these works, several recent works focus on how to preserve identity privacy from public verifiers when auditing the integrity of

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

shared data. Unfortunately, none of the above mechanisms, considers the efficiency of user revocation when auditing the correctness of shared data in the cloud.

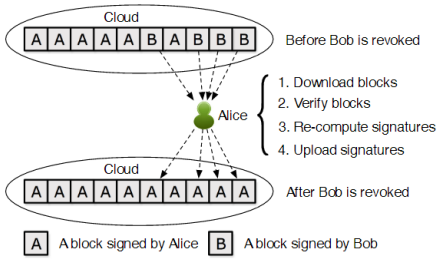


Fig. 1. Alice and Bob share data in the cloud. When Bob is revoked, Alice re-signs the blocks that were previously signed by Bob with her private key.

With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users. For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only.

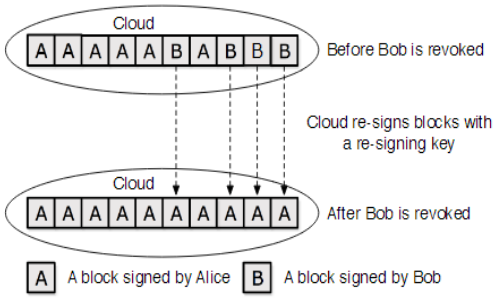


Fig. 2. When Bob is revoked, the cloud re-signs the blocks that were previously signed by Bob with a re-signing key.

Since shared data is outsourced to the cloud and users no longer store it on local devices, a straight forward method to re-compute these signatures during user revocation (as shown in Fig. 1) is to ask an existing user (i.e., Alice) to first download the blocks previously signed by the revoked user (i.e., Bob), verify the correctness of these blocks, then re-sign these blocks, and finally upload the new signatures to the cloud. However, this straightforward method may cost the existing user a huge amount of communication and computation resources by downloading and verifying blocks, and by re-computing and uploading signatures, especially when the number of re-signed blocks is quite large or the membership of the group is frequently changing. To make this matter even worse, existing users may access their data sharing services provided by the cloud with resource limited devices, such as mobile phones, which further prevents existing users from maintaining the correctness of shared data efficiently during user revocation.

EXISTING SYSTEM

An existing system the file uploaded in cloud which not signed by user in each time of upload. So that integrity of shared data is not possible in existing system. However, since the cloud is not in the same trusted domain with each user in the group, outsourcing every user's private key to the cloud would introduce significant security issue.

PROPOSED SYSTEM

Proposed system may lie to verifiers about the incorrectness of shared data in order to save the reputation of its data services and avoid losing money on its data services. In addition, we also assume there is no collusion between the cloud and any user during the design of our mechanism. Generally, the incorrectness of share data under the above semi trusted model can be introduced by hardware/software failures or human errors happened in the cloud. Considering these factors, users do not fully trust the cloud with the integrity of shared data.

ADVANTAGES

- Blocking User account

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- Security question
- Login with secret key in each time

SYSTEM ARCHITECTURE

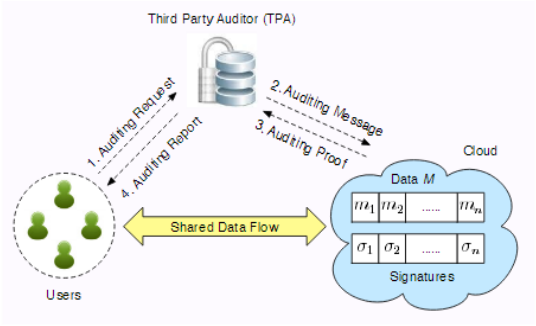


Fig. 3. The system model includes the cloud, the TPA, and users.

SYSTEM OVERVIEW:

The system model includes three entities: the cloud, the third party auditor (TPA), and users who share data as a group (as illustrated in Fig. 3). The cloud offers data storage and sharing services to users. The TPA is able to publicly audit the integrity of shared data in the cloud for users. In a group, there is one original user and a number of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data.

Shared data is further divided into a number of blocks. A user can modify a block in shared data by performing an insert, delete or update operation on the block. Generally, the integrity of shared data is threatened by three factors. First, the cloud service provider may inadvertently pollute shared data due to hardware/software failures and human errors. Second, an external adversary may try to corrupt shared data in the cloud, and prevent users from using shared data correctly. Third, a revoked user, who no longer has the right as existing users, may try to illegally modify shared data. Considering these threats, users do not fully trust the cloud with the integrity of shared data. To protect the integrity of shared data, each block in shared data is attached with a signature, which is computed by one of the users in the group. When shared data is initially created by the original user in the cloud, all the signatures on shared data are computed by the original user. After that, once a user modifies a block, this user also needs to sign the modified block with his/her own private key. By sharing data among a group of users, different blocks may be signed by different users due to modifications from different users. When a user in the group leaves or misbehaves, the group needs to revoke this user. Generally, as the creator of shared data, the original user acts as the group manager and is able to revoke users on behalf of the group. Once a user is revoked , the signatures computed by this revoked user become invalid to the group, and the blocks that were previously signed by this revoked user need to be re-signed by an existing user, so that the correctness of the entire data can still be verified with the public keys of existing users only. Note that allowing every user in the group to share acommon group private key and sign each block with it, is also a possible way to protect the integrity of shared data. However, when a user is revoked from the group, a new group private key needs to be securely distributed to every existing user andall the blocks in the shared data have to be re-signed with the new private key, which increases the complexity of key management and affects the efficiency of user revocation.

Design Goals

To correctly verify the integrity of shared data with efficient user revocation, our public auditing mechanism should achieve the following properties:

(1) Correctness: The TPA is able to correctly check the integrity of shared data.

(2)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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

(3)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 thecloud.

IMPLEMENTATION

MODULE:

- Data Owner (Group Member)
- Cloud Server
- ProxyServer
- Data Integrity
- Public Verifier
- DataConsumer(End-User/Group Member)

MODULES DESCRIPTION:

Data Owner(Group Member)

In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the data file and then store in the cloud. The Data owner can have capable of manipulating the encrypted data file.

Cloud Server

The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

ProxyServer

The Proxy Server manages all data forwards to cloud service provider and if there is any un matching key then it will sent to public Verifier to revoke the user details.

Data Integrity

Data Integrity is very important in database operations in particular and Data warehousing and Business intelligence in general. Because Data Integrity ensured that data is of high quality, correct, consistent and accessible.

Public Verifier

The Public Verifier will perform the revocation and un revocation of the remote user if he is the attacker or malicious user over the cloud data.

Data Consumer (End User / Group Member)

In this module, the user can only access the data file with the encrypted combined key if the user has the privilege to access the file.

PERFORMANCE

We first discuss the communication and computation cost of our mechanism. Then we evaluate the performance of our mechanism in experiments.

A. Communication Cost

the size of an auditing message $\{(l, yl)\}l\in L$ is $c.(|n| +|q|)$bits, where c is the number of selected blocks, $|n|$is the size of an element of set $[1, n]$ and $|q|$is the size of an element of Zq. The size of an auditing proof $\{\alpha,\beta, \{idl\}l\in L\}$ *is* $2d \cdot |p|+c(|id|)$ bits, where d is the number of existing users in the group, |p|is the size of an element of *G1 or Zp*, $|id|$is the size of a block identifier. Therefore, the total communication cost of an auditing task is $2d \cdot |p| + c \cdot (|id| + |n| + |q|)$bits.

B. Computation Cost

As shown in **ReSign** of our mechanism, the cloud first verifies the correctness of the original signature on a block, and then computes a new signature on the same block with a re-signing key. The computation cost of re-signing a block in the cloud is *2ExpG1 +MulG+2Pair+HashG1*, where *ExpG*1 denotes one exponentiation in *G1*, *MulG1*denotes one multiplication in *G1*, Pair denotes one pairing operation one$G1\times G1 \rightarrow G2$, and *HashG*1 denotes one hashing operation in *G1*. The cloud can further reduce the computation cost of the re-signing on a block to ExpG1 by directly re-signing it without verification. The public auditing performed by the TPA ensures that the re-signed blocks are correct. Based on Equation, the computation cost of an auditing task in our mechanism is *(c+d)*

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

ExpG1+(c+2d)MulG1+(d+1)Pair+dMulG2 + cHashG1.

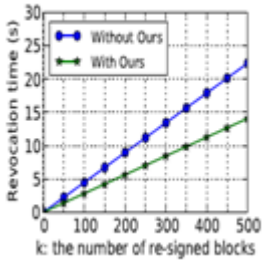
C. Experimental Results

We evaluate the performance of our mechanismin experiments. We utilize Pairing Based Cryptography Library (PBC)1 to implement cryptographic operations in our mechanism. All the experiments are tested under Ubuntu withan Intel Core i5 2.5GHz Processor and 4GB Memory over1, 000 times. In the following experiments, we assume the size of an element of *G1orZp is |p|=160bits*, the size of an element of *Zq* is |q|=80bits, the size of a block identifier is|id|=80 bits, and the total number of blocks in shared data is n=1,000,000. By utilizing aggregation methods from the size of each block can be set as 2KB, then the total size of shared data is 2GB.

1).Performance of User Revocation:

As introduced in Section I, the main purpose of our mechanism is to improve the efficiency of user revocation. Without our mechanism, to revoke a user in the group, an existing user needs to download the blocks were previously signed by the revoked user, verify the correctness of these blocks, re-compute signatures on these blocks and upload the new signatures. In this experiment, we assume the download speed and upload speed for the data storage and sharing services is 1Mbps and 500Kbps, respectively. We also assume the cloud and an existing user leverage the same type of machine (Intel Core i5 2.5GHz Processor and 4GB Memory) to perform user revocation. Let k denote the number of re-signed blocks during user revocation. The performance of our mechanism during user revocation is presented in Figure. The cloud is able to not only efficiently re-sign blocks but also save existing users' computation and communication resources. As shown in Figure, when the number of re-signed blocks is 500, which is only 0.05% of the total number of blocks, the cloud in our mechanism can re-sign these blocks within 15 seconds. In contrast, without our mechanism, an existing user needs about 22 seconds to re-sign the same number of blocks by herself.

Besides, the 500 re-signed blocks that this existing user downloaded costs her extra bandwidth during user revocation. Both of the two revocation time are linearly increasing with an increase of k—the number of re-signed blocks. Since we assume the cloud and an existing user have the same level of computation resource in this experiment, it is easy to see that the gap in terms of revocation time between the two lines in Figure is mainly introduced by downloading the re-signed blocks. In a practical cloud environment, the cloud should have more powerful computation capabilities than personal devices, which allows the cloud to finish the re-signing on data even sooner.



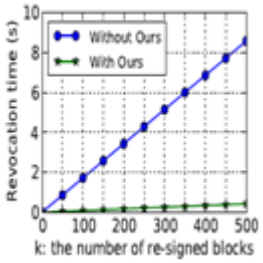


Fig.4.Impact of k on revocationFig.5.Impact of k on revocation time(s) time without verification(s).

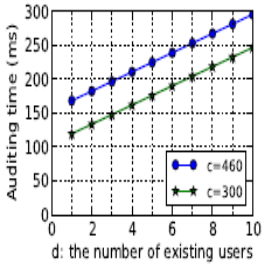
In addition, as we analysed before, the cloud can even directly re-sign data without verification, which can further improve the efficiency of re-signing about 100 times. More specifically, the re-signing time on one block with verification is 28.19 milliseconds while the one without verification is only 0.28 milliseconds. Note that due to the existence of transmission errors in networks, it is not a good idea to allow an existing user to re-sign the blocks without verifying them. Even if an existing user directly re-signs the blocks without verification, compared to our mechanism, this user still needs to spend some extra time to download the blocks. As illustrated in Fig.4.When the number of re-signed blocks is still 500, the cloud in our mechanism can re-sign these blocks in about 0.14 seconds; while an existing user needs about 8.43 seconds by herself. With the comparison between Fig.4 and Fig.5, we can see that the verification on original signatures before re-signing is one of the main factors that can slow down the entire user revocation process. Meanwhile,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

as shown in Fig.4 and Fig.5, the key advantage of our mechanism is that we can improve the efficiency of user revocation and release existing users from the communication and computation burden introduced by user revocation.

2) Performance of Auditing:

We can see from Fig.6 and Fig.7 that, in order to maintain a higher detection probability, a verifier needs more time and communication overhead to finish the auditing task on shared data. Meanwhile, the auditing time (the time that the TPA needs to verify the correctness of an auditing proof based on Equation is linearly increasing with the number of existing users in the group. Our mechanism allows a verifier to efficiently audit the correctness of shared data without retrieving the entire data from the cloud. More specifically, when c=460 and d=10, the communication cost of an auditing task (the communication cost that the TPA requires during an auditing task) is about 11.9KB, and the auditing time of the entire data is only about 300 milliseconds



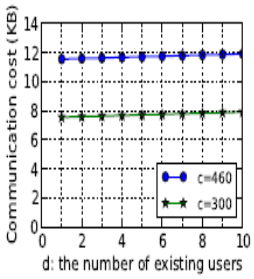


Fig.6.Impact of d on auditing communicationtime(ms).

Fig.7.Impact of d on cost (KB).

CONCLUSION

In this system, 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 the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures. Experimental results show that the cloud can improve the efficiency of user revocation, and existing users in the group can save a 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 Revoation in the Cloud," in the Proceedings of IEEE INFOCOM 2013, 2013, pp. 2904–2912.

[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, 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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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 IEEE INFOCOM 2012, 2012, pp. 693–701.

[12] J. Yuan and S. Yu, "Proofs of Retrievability with Public Verifiability and Constant Communication Cost in Cloud," in Proceedings of ACM ASIACCS-SCC'13, 2013.

[13] H. Wang, "Proxy Provable Data Possession in Public Clouds," IEEE Transactions on Services Computing, accepted.

[14] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving PublicAuditing for Shared Data in the Cloud," in the Proceedings of IEEECloud 2012, 2012, pp. 295–302.

[15] S. R. Tate, R. Vishwanathan, and L. Everhart, "Multi-user DynamicProofs of Data Possession Using Trusted Hardware," inProceedings of ACM CODASPY'13, 2013, pp. 353–364.

[16] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing forShared Data with Large Groups in the Cloud," in the Proceedingsof ACNS 2012, June 2012, pp. 507–525.