An Efficient Public Integrity Checking for Cloud Data Sharing with Proof Verification

E. Saranya
PG/CSE
MARCET, Viralimalai

Abstract **– With information stockpiling and imparting administration in the cloud, client can without much of the stretch change and offer information as a gathering. To guarantee imparted information respectability can be confirmed freely, clients in the gathering need to process mark all the squares in imparted information are for the most part marked by distinctive clients because of information alterations performed by distinctive clients. For security reasons, once a client is disavowed from the gathering, the squares which were beforehand marked this renounced client to download the relating piece of imparted information and re-sign it and amid client. In this paper we propose a novel open examining system for the respectability of imparted information to productive client renouncement as a main priority. By using the thought of intermediary re-marks we permit the cloud to re-sign pieces in the interest of existing client amid client denial, so that current clients don't have to download and re-sign squares independent from anyone else. Moreover, an open verifier is constantly ready to review the honesty of imparted information without recovering the whole information from the cloud, regardless of the fact that some pieces of imparted information has been re-marked by the cloud. Also, our component has the capacity help cluster inspecting by confirming various examining undertaking at the same time. Test results demonstrate that our system can altogether enhance the proficiency of client renouncement.**

Index terms—Public auditing, shared data, user revocation, cloud computing.

1. INTRODUCTION

Cloud Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history:on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage based pricing and transference of risk [1].

To protect the integrity of data in the cloud, a number of mechanisms [3]-[15] 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 Posession[3]). 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 [3]-[13] focus on auditing the integrity of personal data. Different from these works, several recent works[14],[15] focus on how to preserve identity privacy from public verifiers when auditing the integrity of shared data. Unfortunately, none of the above mechanisms, considers the efficiency of user revocation when auditing the correctness of shared data in the cloud.

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 must be revoked from the group. As a result this revoked user should no longer b able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group[16]. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously 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.

Since shared data is outsourced to the cloud and users no longer store it on local devices, a straightforward method to re-compute these signatures during user revocation (as shown in Fig.1) is to ask an existing 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.

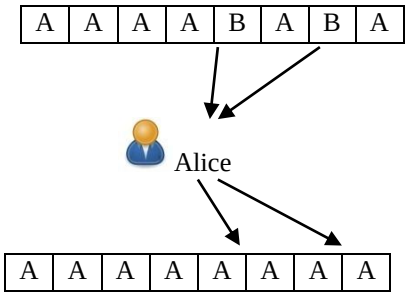

Fig.1. Alice and Bob share data in the cloud. When Bob is revoked, Alice re-signs the blocks that were previously signed by Bob with her private key.

In Fig.1. Alice and Bob share data in the cloud. When Bob is revoked, Alice re-signing task for existing users without asking them to download and re-sign blocks. 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 issues. Another important problem we need to consider is that the re-computation of any signature during user revocation should not affect the most attractive property of public auditing – auditing data integrity publicly without retrieving the entire data. Therefore, how to efficiently reduce the significant burden to existing users introduced by user revocation, and still allow a public verifier to check the integrity of shared data without downloading the entire data from the cloud, is a challenging task.

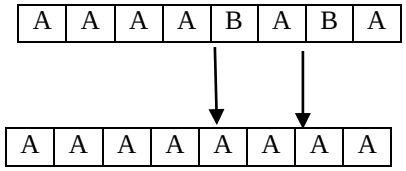

Fig.2. When Bob is revoked, the cloud re-signs the blocks that were previously signed by Bob with the re-signing key.

In this paper, we propose, a novel open examining system for the integrity of shared data with efficient user revocation in the cloud. In our mechanism, by utilizing the idea of proxy re-signatures [17], once a user in the group is revoked, the cloud is able to resign the blocks, which were signed by the revoked user, with a re-signing key (as presented in Fig.2). As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, who is not in the same trusted domain with each user, is only able to convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user. By designing a new proxy re–signature scheme with nice properties which traditional proxy re signatures do no have, our mechanism is always able to check the integrity of shared data without retrieving the entire data from the cloud.

Moreover, our proposed mechanism is scalable, which indicates it is not only able to efficiently support a large number of users to share data and but also able to handle multiple auditing tasks simultaneously with batch auditing. In addition, by taking advantages of Shamir Secret Sharing[18], we can also extend our mechanism into the multi-proxy model to minimize the chance of the misuse on re-signing keys in the cloud and improve the reliability of the entire mechanism.

2. PROBLEM STATEMENT

In this section, We describe the system and security model, and illustrate the design objectives of our proposed mechanism.

A. System and Security Model

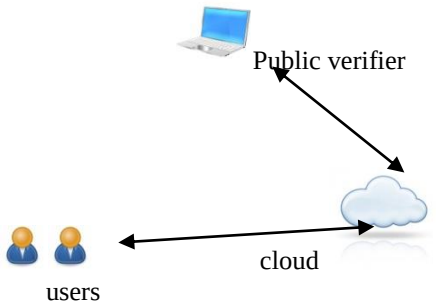

Fig.3.The system model includes the cloud, the public verifier, and users.

As illustrated in Fig.3, the system model in this paper includes three entities: the cloud, the public verifier, and users (who share data as a group). The cloud offers dta storage and sharing services to the group. The public verifier, such as 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 can provide verification services on data integrity of shared data via a challenge- and – response protocol with the cloud. In the group, there is one 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 divided into a number of blocks. A user in the group can modify a block in shared data by performing an insert, delete or update operation on the block.

In this paper, we assume the cloud itself is semi-trusted, which means it follows protocols and does not pollute data integrity actively as a malicious adversary, but it 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.

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. Specifically, when shared data is initially crated 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 should be re-signed by an existing user's private key, so that the correctness of the entire data can still be verified with the public keys of existing users only.

2.2. Design objectives

Our proposed mechanism should achieve the following properties: (1) Correctness: The public verifier 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 complete valid signatures on shared data. (3) Public Auditing: The public verifier 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. (4) Scalability: Cloud data can be efficiently shared among a large number of users, and the public verifier is a able to handle a large number of auditing tasks simultaneously and efficiently.

3. OVERVIEW

Based on the new proxy re-signature scheme and its properties in the previous section, we now present panda a public auditing mechanism for shared data with efficient user revocation. In our mechanism, the original user acts as the group manager, who is able to revoke users from the group when it is necessary. Meanwhile, we allow the cloud to perform as the semi-trusted proxy and translate signatures for users in the group with re signing keys. As emphasized in recent work [23], for security reasons, it is necessary for the cloud service providers to storage data and keys separately on different servers inside the cloud in practice. Therefore, in our mechanism, we assume the cloud has a server to store shared data, and has another server to manager resigning keys. To ensure the privacy of cloud shared data at the same time, additional mechanisms, such as [24], can be utilized. The details of preserving data privacy are out of scope of this paper. The main focus of this paper is to audit the integrity of cloud shared data.

A. Support Dynamic Data

To Build the entire mechanism, another issue we need to consider is how to support dynamic data during public auditing. Because the computation of a signature includes the block identifier, conventional methods – which use the index of a block as the block identifier(i.e., block mj is indexed with j) – are not efficient for supporting dynamic data [8], [14]. Specifically, if a single block is inserted or deleted, the indices of blocks that after this modified block are all changed, and the change of those indices requires the user to re-compute signatures on those blocks, even though the content of those blocks are not changed.

by leveraging index hash tables [8],[14], we allow a user to modify a single block efficiently without changing block identifiers of other blocks The details of index hash tables are explained in Appendix A. Besides a block identifier and a signature, each block is also attached with a signer identifier to distinguish which key is required during verification, and the cloud can utilize it to determine which resigning key is needed during user revocation.

B. Construction

This 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 a re-signing key for each pair of users in the group. As argued in previous section, we still assume that private channels exist between each pair of entities during the generation of re-signing keys and there is no collusion. 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, which were previously signed by this revoked user, with a resigning key. The verification on data integrity is performed via a challenge and response protocol between the cloud and a public verifier. More specifically, the cloud is able to generate a proof of possession of shared data in ProofGen under the challenge of a public verifier. In ProofVerify, a public verifier is able to check the correctness of a proof responded by the cloud.

In ReSign, Without loss of generally, we assume that the cloud always converts signatures of a revoked user into signatures of the original user. The reason is that the original user acts as the group manager, and we assume he/she is secure in our mechanism. Another way to decide which re-signing key should be used when a user is revoked from the group, is to ask the original user to create a priority list (PL). Every existing user's id is in the PL and listed in the order of re-signing priority. When the cloud needs to decide which existing user the signatures should be converted into, the first user shown in the PL is selected. To ensure the correctness of the PL, it should be signed with the private key of the original user(i.e. the group manager).

C. Efficient and Secure User Revocation.

We argue that our mechanisms is efficient and secure during user revocation. It is efficient

because when a user is revoked from the group, the cloud can re-sign blocks that were previously signed by the revoked user with a re-signing key, while an existing user does not have to download those blocks, re-compute signatures on those blocks and upload new signatures to the cloud.

The resigning preformed by the cloud improves the efficiency of user revocation and saves communication and computation resources for existing users.

The user revocation is secure because only existing users are able to sign the blocks in shared data. As analyzed in Theorem 1, even with a resigning key, the cloud cannot generate a valid signature for an arbitrary block on behalf of an existing user. In addition, after being revoked from the group, a revoked user is no longer in the user list, and can no longer generate valid signatures on shared data.

4. 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 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 emputationand communication resources during user revocation.

REFERENCE

1. B.Wang, B.Li and H.Li, "Public Auditing for Shared Data with Efficent User Revocation in the Cloud," in te Proceedings of IEEE INFOCOM 2013, 2013, pp.2904-2912.
2. M.Armbrust, A.Fox, R.Grifith, 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 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.Huand 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", IEEE Transactions on Services Computing,vol.5,No.2,pp. 220-232,2011.
10. Y-Zhy,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 Services Computing, accepted.
12. J.Yuan and S.Yu, "Proofs of Retrievability with Public Verifiability 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, B.Li, and H.Li,"Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," in the Proceedigns of IEEE Cloud 2012,2012,pp.295-302.
15. S.R.Tate, R.Vishwanathan, and L.Everhart, "Multi-user Dynamic Proofs of Data Possession Using Trusted Hardware, " in Proceedings of ACM CODASPY'13,2013,pp.353-364.
16. B.Wang, B.Li, and H.Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in th Cloud", in the Proceedigns of ACNS 2012, June 2012, pp.507-525.
17. M.Blaze, G.Bleumer, and M.Strauss, "Divertible Protocols and Atomic Proxy Cryptography," in the Proceedings of Eurocrypt 98,Sringer-Verlag, 1998,pp.127-144.
18. A.Shamir, "How to share a secret," in Communication of ACM, vol.22, no.11, 1979, pp.612-613.
19. B.Wang,H.Li, and M.Li,. "Privacy=Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics," in the Proceedigns of IEEE ICC 2013,2013.
20. B.Wang, S.S.Chow, M.Li, and H.Li, "Storing Shared Data on the Cloud via Security – Medicator," in Proceedings of IEEE ICDCS 2013,2013.