Achieve High Verifiability using Proxy Resignature and TPA in User Revocation within the Cloud

Ms. Megha D. Bochare
Department Of Computer Engg,
SAE, Kondhwa (BK), Pune
Email:bochare.megha@gmail.com

Prof. L. J. Sankpal
Department Of Computer Engg,
SAE, Kondhwa (BK), Pune
Email: ljsankpal.sae@sinhgad.edu

Abstract— Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from cloud. User can get relaxation from the burden of local data storage and maintenance. In addition, we have an efficient probabilistic query and audit services to improve the performance of approach based on periodic confirmation. So that the users existing blocks by themselves do not need to sign up and download the proxy by using the idea of re-signatures, we block the user revocation on behalf of existing users to the cloud, the signing in again to allow for..In addition, a public Verifier always without retrieving all of the data shared data is able to audit the integrity of Cloud, even if part of the shared data has been signed by the cloud again. Moreover, our system by multiple auditing functions with batch verification audit is able to support. Experimental results show that our system fairly can improve the efficiency of user cancellation. Data storage and sharing services in the cloud, users can easily modify and share data in a group. Shared data to ensure unity in public, group users shared data to calculate signatures on all blocks need to be verified. Shared data by different users in different blocks are usually due to data revisions have been signed by individual users. The proposed system considers proxy resign, if the user from group get revoked. Cloud is able to resign block, which was created previously by the revoked user with existing users private kye. As a result, user revocation can be greatly improved, and capacity of computing and communications resources of existing users can be saved.

Keywords—*Public auditing, shared data, user revocation, cloud computing.*

*****

I. INTRODUCTION

In cloud 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. The proposed system considers proxy resign ,if the user from group get revoked. Cloud is able to resign block, which was created previously by the revoked user with existing users private kye. As a result, user revocation can be greatly improved, and capacity of computing and communications resources of existing users can easily be 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. Nice property with a new proxy signature scheme by designing the traditional re-proxy don't tend to re signatures, our system always check the integrity of the shared data without retrieving all of the data from the cloud is capable.

If the cloud have each the user's private key then, it is simple for cloud to resign the block with existing users private key without requesting blocks to be download. But such method introduces security. Another important issue is that we need to consider any of the signature computation should not be affected during user revocation The most attractive qualities of public audit-data integrity auditing publicly without retrieving all of the data. Therefore, efficiency is important to reduce load the For existing users load introduced by user revocation, and still a verifier without downloading the entire data of shared data allow to check the integrity of Cloud, is a daunting task.

The number of mechanisms has been proposed to protect the integrity of the data. In these mechanisms, signature is attached to each block of data, and data integrity depends on the accuracy of all signatures. One of the most important and common features of these mechanisms to allow a user without downloading the entire data integration in the cloud to check public Verifier referred to as public audit (or provable data possession is marked as. This public Verifier a client who mostly want to use the data for special purposes or which users data integrity is able to provide verification services on a third party Auditor (TPA) can be. Most of the previous work focused on auditing the integrity of personal data. These works, shared data from audit integrity of the verifiers during the public how to recognize privacy protected on many recent works from consideration. A user within the cloud will modify a block in shared data by activity associate degree insert, delete or update operation on the block.

Public Verifier: The public verifier is able to correctly check the integrity of shared data. That means it checks the correctness of the shared data that is share by the user.

User: User is the person who shares the data in the group or as a group.

Cloud: This is an entity that provides data storage service.

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.

II. RELATED WORK

The number of mechanisms [1]–[15] have been proposed To protect the integrity of data in the cloud. 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.

In [1]. B. Wang, B. Li, and H. Li, 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.

In [2] the model a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it is introduced. 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.

In [4] a proof-of -retrievability system, a data storage center convinces a verifier that he is actually storing all of a client's data. The central challenge is to build systems that are both efficient and provably secure – that is, it should be possible to extract the client's data from any prove that passes a verification check. In this paper the first proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski is given. The first scheme is built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-retrievability with public verifiability. And second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-retrievability scheme with private verifiability (but a longer query). Both schemes rely on homomorphicproperties to aggregate a proof into one small authenticator value.

[5] Now a day's Cloud computing is emerging field because of its Performance, high availability, at low cost. Cloud is kind of Centralized database where many organizations store their data, retrieve data and possibly modify data. In the cloud many Services are provided to the client by cloud. Data store is main future that cloud service provides to the big organization to store huge amount of data. But still many organizations are not ready to implement cloud computing technology because of following reason. That is Lack of security, Data redundancy, Misbehavior of the server. So the main objective of this paper is to solve the above reasons that are To prevent unauthorized access, it can be done with the help of a distributed scheme by using homomorphism token to provide security of the data in cloud. The cloud is support for data redundancy means clients can insert, delete or can update data so there should be security mechanism which ensure integrity of data. This paper also secures the data while the misbehaving of the server arise.

[6] Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, 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, we 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 dynamics via 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..

[8]In this paper, Y. Zhu, H. Wang, Z. Hu, G.-J.Ahn, H. Hu, and S. S. Yau propose a dynamic audit service for verify- ing the integrity of untrusted and outsourced storage.The audit service, constructed based on the techniques, fragment

structure, random sampling and index-hash table, can sup- port provable updates to outsourced data, and timely abnor- mal detection. In addition, an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services.

In [13], Wang et al. consider dynamic data storage in distributed scenario, and the proposed challenge-response proto- col can both determine the data correctness and locate possible errors.

Shah et al. [10]-[15]propose introducing a TPA to keep online storage honest by first encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies the integrity of the data file and the server's possession of a previously committed decryption key.This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up. Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data.

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 shared data. Unfortunately, none of the above mechanisms, considers the efficiency of user revocation when auditing the correctness & of accuracy of shared data in the cloud over the course of the audit.

Design Objectives

Our proposed mechanism should achieve the follow- ing 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 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.
- 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

III. Proposed Approach Framework and Design

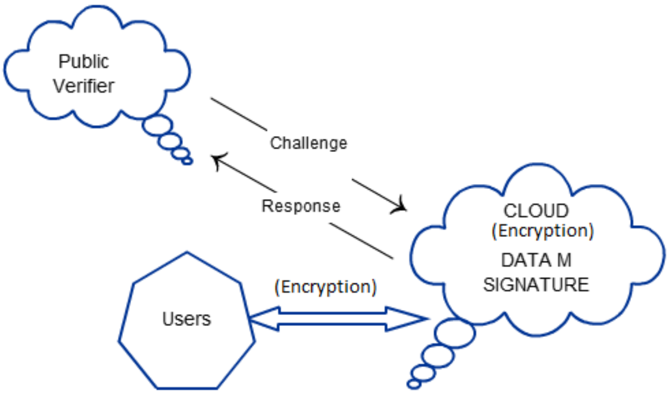

Fig. Architecture Design

Let G1 and G2 be two groups of order p, g be a generator of G1, e: G1×G1 → G2 be a bilinear map, w be another generator of G1. The global parameters are (e,p,G1,G2,g,w,H), where H is a hash function with

$H : \{0,1\}* \rightarrow G1.$

Homomorphic Authenticators:

By using Homomorphic authenticators a public verifier is able to check the integrity of data stored in the cloud without downloading the entire data. It has following properties

a)Blockless verifiability:

Given σ_1 and σ_2, any two random values α_1, α_2 in Z_p and a block $m' = \alpha_1 m_1 + \alpha_2 m_2$, without knowing m_1 and m_2 a verifier is able to check the correctness of block m'.

2)Non-malleability:

Given m1 and m2, α_1 and α_2, two random values α_1, α_2 in Z_p and a block $m' = \alpha_1 m_1 + \alpha_2 m_2$, a user without private key sk is unable to generate a valid signature α_1 on block m by combining α_1 and α_2.

The proxy re-signature scheme includes five algorithms: KeyGen, ReKey, Sign, ReSign and Verify.

KeyGen :

Each user in the of cloud generates his/her public key and private key using RSA.Given global parameters (e,p,G1,G2, g,w,H), a user uA selects a random a ∈ Z∗ p, and outputs public key $pk_A = g^a$ and private key $sk_A = a$.

Sign: Using own private key user encrypts the message digest with encryption algorithm & user u_A outputs the signature on block m as:

$\sigma = (H(id)\, w^m)a \in G1$

ReKey: The proxy generates a re-signing key $rk_{A\rightarrow B}$ as follows:

- The proxy generates a random r ∈ Z∗ p and sends it to user u_A;
- user u_A computes and sends r/a to user u_B, where sk_A= a;
- user u_B calculates and sends rb/a to the proxy, where sk_B = b;
- the proxy recovers rk $_{A\rightarrow B}$ = b/a ∈ Z∗p.

.

ReSign : Using the key generated by Reykey algorithm proxy is able to convert a signature of revoked user into signature of existing user 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 any user.

Verify. The TPA generates an audit message to the cloud server to make sure that the cloud server has retained the data file properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file and its verification metadata. TPA then verifies the response via Verify

IV. CONCLUSION

Experimental results show that the cloud can improve User revocation, efficiency and existing users to a group of users during the cancellation computation and communication can save a significant amount of resources. When a user in the group is canceled, users with proxy re-signed again allow to sign. We share it with quality users in the cloud revocation data for a proposed new public auditing mechanism. The existing users in the group can save a significant amount of computation and communication resources during user revocation & high verifiability can be achieved.

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] Cloud Security Alliance, “Security Guidance for Critical Areas of Focus in CloudComputing,” http://www.cloudsecurityalliance.org, 2009.

[11] 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.

[12] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring data storage security in cloud computing,” in Proc. of IWQoS’09, Charleston, South Carolina, USA, 2009.

[13] 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.

[14] 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.

[15] Networks,” in Proceedings of IEEE INFOCOM, 2011, pp. 2435-2443.