ISSN 2395-1621

Review: Public Auditing Securing Multi Owner Data In Cloud Using Multiple Third Parties With Efficient User Revocation

[#1]Atul S Rakate, [#2]Dr. D. M. Yadav

[1]atulrakate93@gmail.com
[2]dineshyadav800@gmail.com

[#1]Department of Computer Engineering, University of Pune, JSPM's Rajarshi Shahu School of Engineering & Research, Narhe, Pune, Maharashtra, India.

[#2]Director, JSPM's Rajarshi Shahu School of Engineering & Research, Narhe, Pune, Maharashtra, India.

ABSTRACT

Cloud registering is one of the greatest development in today's computing world, due to services of cloud it is not only possible to store but also to share with multiple users. When user have large size of shared data, to maintain integrity of such data in challenging. TPA (Third Party Auditor) is used to store and share data in cloud computing. To verify the integrity of shared data, data owners in the group needs to compute signatures on all shared data blocks. In the shared data different blocks are signed by different users due to data modifications performed by different users. Efficient user revocation is important threat in data sharing in groups. This paper gives privacy preserving public auditing system used to guarantee that the third party auditor would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process

Keywords— **Cloud Computing, Data Integrity, Public auditing, Efficient user revocation.**

ARTICLE INFO

Article History

Received :28th December 2015

Received in revised form :

29th December 2015

Accepted : 30th December , 2015

Published online :

4th January 2016

I. INTRODUCTION

Cloud computing is the fastest growing internet-based computing in today's world. It have also improved the ability of data sharing and data storing. Cloud is nothing but the collection of computers which are interconnected and run applications. Cloud computing provides us shared pool of computer resources, on-demand networks access and on-demand services[1]. The main benefit of cloud computing is cost efficiency. But there is also disadvantages with respect to data security. Cloud computing contains set of rules, methods and powers deployed to save data, applications and infrastructure of cloud computing. Cloud computing provides us security but there are some issues needs to be considered, the issues such as data integrity, data privacy and data accessed by third party.

Integrity is nothing but consist flow. It is an important fact that affects the performance of the cloud. Data integrity contains rules for writing the data in a reliable manner to the data storages which can be retrieved in the same format without any changes later. The task of maintaining data integrity is quite difficult. Various techniques have been proposed [2]-[15] to protect integrity of data. Attaching the Signature to each block of data is one of those. After storing the data on cloud storage user have no control on data so the correctness of data is being put at the risk.

On cloud we can able to store data as a group and share it or modify it within a group. In cloud data storage contains two entities as cloud user (group members) and cloud service provider/ cloud server. Cloud user is a person who stores large amount of data on cloud server which is managed by the cloud service provider. User can upload their data on cloud and share it within a group. A cloud service provider will provide services to cloud user. The major issue in cloud data storage is to obtain correctness and integrity of data stored on the cloud. Cloud Service Provider (CSP) has to provide some form of mechanism through which user will get the confirmation that cloud data is

secure or is stored as it is. No data loss or modification is done by unauthenticated member.

Centralized data management system is performed ny the cloud computing, but to provide the security to this centralized data is challenging. TPA is used for providing data security for centralized data. The reliability is increased as data is handled by TPA but data integrity is not achieved. TPA uses encryption to encrypt the contents of the file. It checks data integrity but there is threat of TPA itself leaks user's data.

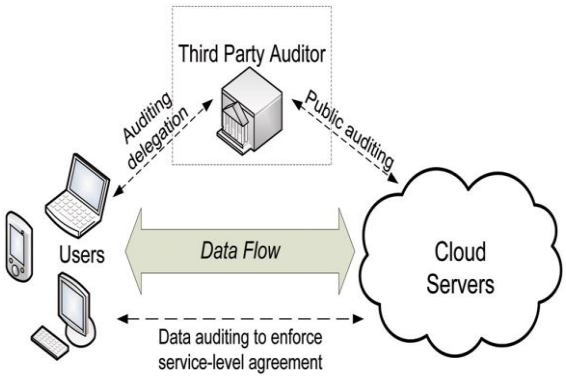

Fig.1 Architecture of Cloud Data Storage Service

Authors of [3] specify way to achieve storage correctness without Trusted Third Party (TTP). They achieve this by using secure key management, Flexible access right managements and light weight integrity verification process for checking the unauthorized change in the original data without requesting a local copy of the data.

II. RELATED WORK

Provable Data Possession at Untrusted Stores [8]

The concept of public auditability was given by Ateniese et al. [8]. They have described this concept in their defined provable data possession (PDP) model for making sure the ownership of data files on no trust worthy storage and used Rivest Shamir Adleman based homomorphic linear authenticators for auditing of outsourced data. Provable data possession model allows client (who has stored data on untrusted server) to verify, that the server possesses the original data without retrieving it. PDP model creates probabilistic proofs of possession by sampling random sets of blocks from the server. This significantly minimizes I/O costs. The client maintains a constant amount of metadata to verify the proof.

The response protocol sends a modest, constant quantity of information, which reduces network communication. Hence, the PDP model for distant information inspection supports large data sets in widely-distributed storage systems. Authors have presented two provably-secure PDP schemes that are more capable than prior solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments by execution confirm the practicality of PDP and tell that the performance of PDP is restricted by disk Input output and not by cryptographic computation. For auditors who are external, linear combination of sample blocks were required and when directly used, their protocol did not provided privacy preserving and thus may leak the user data to auditors.

Compact Proofs of Retrievability [7]

Shacham et al. [7] built proof of retrievability (PoR) model and constructed a random linear function based homomorphic authenticator which enables limitless number of inquiry and requires minimal communication overhead. Shacham et al.s first methods, built from BLS signatures and secure in the random oracle model, characteristics of a proof-of retrievability protocol in which the clients inquiry and servers response are both very short. This method allows public verifiability: anyone can act as a verifier, not only the file owner. Second method, which builds on pseudorandom functions (PRFs) and is protected in the regular model, allows only secret confirmation. It features a proof-of-retrievability protocol with a yet shorter servers response than the first method proposed, but the clients query is very long. Both methods depend on homomorphic characteristics to comprehensive evidence into one small authenticator value.

Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing [6]

Wang et al [6] projected a theory to combine BLS-based HLA with MHT to sustain equally public auditability and full data dynamics. Considered a like support for incomplete dynamic data storage in a disseminated situation with added quality of data error localization. To efficiently carry public auditability without having to recovering the data blocks themselves, resort to the homomorphic authenticator system.

Homomorphic authenticators are unforgeable metadata generated from individual data blocks, which can be strongly aggregated in such a way to reassure a verifier that a linear combination of data blocks is appropriately computed by verifying only the aggregated authenticator. In this design, here proposal is to use PKC based homomorphic authenticator (e.g. BLS signature or RSA signature based authenticator) to implement the verification protocol with public auditability.

In the following explanation, there is present the BLS-based method to illustrate the design with data dynamics support. As will be shown, the schemes designed under BLS construction can also be implemented in RSA construction.

Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing [5]

K.Ren et al [5] proposed privacy preserving system where public key based homomorphic authenticator is combined with random masking which fulfill the requirement of efficient audit without demanding the local copy of data and user data privacy. Explored the technique of bilinear aggregate signature for multi user setting which allow third party auditor execute multiple number of auditing task together.

Privacy-Preserving Public Auditing for Secure Cloud Storage Auditing [4]

C.Wang et al [4] proposed privacy-preserving public auditing system for data storage security in Cloud Computing. Homomorphic linear authenticator and random masking have been used to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which

not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, privacy-preserving public auditing protocol further extended into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that discussed schemes are provably secure and highly efficient.

Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud [10]

B. Li et al [2] has proposed a privacy preserving mechanism that supports public auditing on shared data stored in the cloud. He has used ring signature to compute verification metadata and identity of signer is kept private from public verifier, who are able to efficiently verify shared data integrity without retrieving the entire file. Additionally this mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one and experimental results demonstrate the effectiveness and efficiency of this mechanism when auditing shared data integrity.

III. PERSPECTIVE AND DESIGN

1. Problem Statement:

With relinquish trends in cloud, Data integrity is one of the critical issue, as there is lack of identity privacy, where the users are unacquainted with the auditor of the data, over geographically scattered datacentres. This features of cloud computing evolved various concerns related to user's identity, data integrity and users availability. Ultimately this influences to propose an enhanced model in order to audit the data integrity and keeping the identity privacy with efficient user revocation while sharing.

2. Methodology:

I)Privacy-Preserving Public Auditing For Secure Data Storage [4]: Homomorphic linear authenticator with random masking technique is used when there is a need of public auditability without retrieving the data blocks. HLAs are unforgeable verification metadata which are used to authenticate the integrity of a data block. HLAs can be aggregated. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks. To perform these tasks it uses following algorithms:

- KeyGen: KeyGen is a key generation algorithm that is executed by the user to setup the scheme.
- SigGen: SigGen is executed by the user to produce verification metadata, which may consist of signatures, or other linked information that will be used for executing audit.
- GenProof: GenProof is executed by the CS to produce a verification of data storage rightness.
- VerifyProof: VerifyProof is executed by the TPA to audit the verification from the CS.

This public auditing technique works as follows:

- Initialization: Initialization phase works with two algorithms, Key-Gen and SigGen. By running KeyGen algorithm, user initializes the public and secret parameters of the system and verification metadata for data file is generated using SigGen algorithm. Data file F and the verification metadata is stored on cloud server and user deletes its local copy. User may alter the data file F by expanding it or including additional metadata to be stored at server as a part of pre-processing.
- Audit: The Audit phase works with two algorithms, GenProof and VerifyProof. Whenever TPA wants to verify that the cloud server has retained the data file F properly or not, at that time TPA is sending audit message or challenge to cloud server. By running GenProof, cloud server will derive a response message from a function of the stored data file F and its verification metadata. Then TPA verifies the response by running algorithm VerifyProof.) K-Means: K points[11] are set in the space produced by the objects. These points serve as initial group centroids. Each object is assigned to the group that has the closest centroid. Recalculate the positions of the K centroids after all the objects have been assigned. This process continues unless and until the centroids no longer move. It divides of the objects into clusters.

First third party auditor (TPA) retrieves file and verifies its signature, if signature verification occurs successfully then next step is being performed, else the process is terminated. In next step TPA generates a random challenge request and send is to server. After receiving the challenge request, server computes μ', σ, R. Here μ' is linear combination of sampled blocks, σ is aggregated authenticator and R is calculated for inserting the random masking so that by evaluating the linear equations, TPA cannot predict the data. Server finally computes μ by using Υ, μ' and R and send the calculated values μ, σ and R to TPA as a storage correctness proof. Then TPA verifies the response by running algorithm VerifyProof.

II) Public Auditing Scheme For Shared Data And User Data Revocation [1]: Homomorphic authenticable proxy resignature scheme with Panda public auditing mechanism is used for public auditing of shared data with efficient user revocation in cloud. Here semi trusted cloud re-signs the blocks which were signed by revoked user, using proxy re-signature and save a significant amount of computation and communication resources during user revocation. Additionally it support dynamic data and batch auditing for handling number of task simultaneously.

Scheme Details: Let G1 and G2 be two groups of order p, g and w be the generator of G1.e:G1 X G1 → G2 be a bilinear map (e,p,G1,G2,g,w,H) are the global parameters where H is the hash function. Total number of blocks in shared data is n, shared data is described as M = m1,…,mn) and total number of users in a group is d.

Flow of this mechanism is described below with the help of algorithms.

- KeyGen: This is key generation algorithm and here user generates their public and private key. Here original user creates a user list which contains ids

of all the users in the group. This user list (UL) is public and signed by the original user.

- ReKey: Through this algorithm cloud computes resigning key for each pair of user in group and it is assumed that private channels as SSL exist between each pair of entities and there is no collusion. For this cloud generates a random r and send it to user A, user A calculates some value and send it to user B then user B do same calculation and pass the value to cloud and by this value cloud recovers the Rekey.
- Sign: This algorithm is used for signing the block by original user i.e. creator of data and if a user in the group modifies a block in shared data, the signature on the modified block is also computed as in Sign. Given private key as ski = πi, block m_k € Zp its block identifier idk.
- ReSign: This algorithm is used for re-signing the blocks by cloud which were previously signed by revoked users. If the verification result is 0, the cloud outputs 1; otherwise, it outputs σ'k.
- 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 (l,nl)l€L. and send it to cloud and in second part cloud generates a proof of possession {α,β,((idl,sl),leL)} of shared data M, after receiving the auditing message.

In ReSign algorithm, Cloud always translates revoked users signature into signature of data creator (original user) because original user acts as group manager and assumed to be secure in this 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. Based on the properties of bilinear maps; the correctness of this mechanism in ProofVerify can be explained.

IV. CONCLUSION

This paper discusses Privacy preserving public auditing mechanisms, homomorphic linear authenticator with random masking have been used to guarantee that the third party auditor would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the user's fear of their outsourced data leakage. Homomorphic authenticable proxy resignature scheme with Panda public auditing mechanism checks shared data integrity along with efficient user revocation. Furthermore, these mechanisms are able to support batch auditing by verifying multiple auditing tasks simultaneously.

REFRENCES

[1] Boyang Wang, Baochun Li and Hui Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud", IEEE Transactions on services computing, vol. 8, no. 1, January/February 2015.

[2] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud", Proc. IEEE CLOUD, pp. 295-302, 2014.

[3] H. Wang, "Proxy Provable Data Possession in Public Clouds", IEEE Trans. Services Computing, vol. 6, no. 4, pp. 551-559, Oct.-Dec. 2013.

[4] C. Wang, Q. Wang, K. Ren ,"Privacy-Preserving Public Auditing for Secure Cloud Storage Auditing", IEEE transaction on computer, 2013.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data
Storage Security in Cloud Computing", Proc. IEEE INFOCOM, pp. 525-533, 2010.

[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing",Proc. 14th European Conf. Research in Computer Security (ESORICS09), pp. 355-370, 2009.

[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability", Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT08), pp. 90-107, 2008.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), pp. 598-610, 2007.

[9]Shamir, How to Share a Secret, Comm. ACM, vol. 22, no. 11,pp. 612-613, Nov. 1979.

[10] C.Wang, Q.Wang, K. Ren, andW. Lou,"Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.no. 5, pp. 847-859, May 2011.

[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li,"Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[12] C. Wang, K. Ren, W. Lou, and J. Li,"Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[13] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), 2007.

[14] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession,"Proc. ACM Conf. Computer and Comm. Security (CCS '09), 2009.

[15] H. Shacham and B. Waters,"Compact Proofs of Retrievability," Theory and Application of Cryptology and Information Security: Advances in Cryptology Dec. 2008.