Privacy Preserving Public Auditing For Personal and Shared Data with Efficient User Revocation

Aseema Jana
Department of Computer Engineering, Dhole Patil College of Engineering, Savitribai Phule Pune University, Maharashtra.

Abstract **- Cloud provides services like data storage and data sharing in a group. Users can remotely store their data on cloud and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. But the management of the data and services may not be fully trustworthy on cloud, as users no longer have physical possession of the outsourced personal data so data integrity protection becomes a difficult task. Maintaining the integrity of shared data services where data is shared among number of cloud user, is also a challenging task. This paper gives privacy preserving public auditing system for data storage security in cloud computing and for that it uses homomorphic linear authenticator with random masking technique. 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.**

Index Terms **- Data storage, privacy-preserving, public auditing, shared data, user revocation, cloud computing.**

I. INTRODUCTION

CLOUD Computing provides characteristics as on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk, these characteristic makes cloud computing suitable for enterprises. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From user's perspective, including both individuals and IT enterprises, remotely storing data to the cloud provide advantages as a relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. As user don't have control over data after storing it in cloud so the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices but there is threat of data integrity. Secondly, Cloud service provider (CSP) might by discard data that has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation.

To address these problems, public key based homomorphic linear authenticator (HLA) technique can be used for auditing and by integrating the HLA with random masking, protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefits the design for batch auditing.

In share data services, as data is modified by different users that's why different blocks in shared data is signed by different users .Each block is attached with a signature and integrity of data relies on the correctness of all the signatures. Once a user is revoked from group, at that time the block signed by the revoked user must be resigned by the existing user for security reasons. In basic method, first data blocks are downloaded by existing user and then upload process is done after verifying the correctness and resigning of block by existing user, which results in large amount of communication and computation cost due to large size of shared data in cloud.

II. LITERATURE SURVEY

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.

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.

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.

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.

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.

H. Wang et al [3] Proposed proxy provable data possession protocol for remote data checking as PPDP is major concern in public cloud when client cannot perform the remote data possession checking. This proposed protocol is based on bilinear pairing technique and through security analysis and performance analysis author has proved that the protocol is provable secure and efficient.

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.

B. Wang et al [1] proposed public auditing mechanism for shared data using homomorphic authenticator and 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.

III. APPROACH AND DESIGN

A. PROBLEM DEFINITION

While using cloud services as data storage and data sharing in a group, Integrity of personal and shared data on cloud and user revocation are major concerns. This paper uses the concept of homomorphic linear authenticator with random masking technique for personal data and Homomorphic authenticable proxy resignature scheme with Panda public auditing mechanism for shared data and user revocation.

B. METHODOLODY

Diagram shows the architecture of cloud data storage and methodology is explained in detail below:

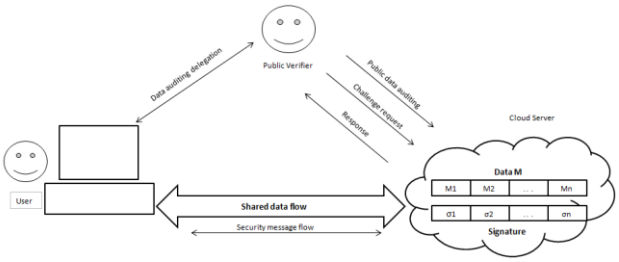

Fig. 1. Architechture of cloud data storage service

1) 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. This scheme uses below 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: is executed by the TPA to audit the verification from the CS.

This Public auditing technique works in two phases:

- Setup: Setup 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.

Flow of scheme: 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 γ, 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.

2) 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 $m_K \in Z_p$ and its block identifier idk. User ui outputs the signature on block mk as

$$\sigma_k = (H(id_k)w^{mk})^{\pi i} \epsilon G_1$$

- ReSign: This algorithm is used for re-signing the blocks by cloud which were previously signed by revoked users. Re-signing key, Public key, signature, block, block identifier, cloud checks that $e(\sigma_k, g) = e(H(id_k)w^{mk}, pk_i)$. 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 $(l, n_l)_{l\epsilon L}$ and send it to cloud and in second part cloud generates a proof of possession $\{\alpha, \beta, (id_l, s_l)_{l\epsilon L})\}$ of shared data M, after receiving the auditing message.

ProofVerify: By using ProofVerify algorithm public verifier is able to check the correctness responded by cloud. Here verification of shared data is done by using challenge and response protocol between the cloud and public verifier. Given an auditing message $(1, n_l)_{l\epsilon L}$, auditing proof $\{\alpha, \beta, (id_l, s_l)_{l\epsilon L})\}$ and all existing users public key $(pk_1, \cdots, pk_d)$ and public verifier checks the correctness of this auditing proof as below, if the result is 1, verifier believes that integrity in all the blocks in shared data M is correct otherwise public verifier outputs 0.

$$e\left(\pi_{i=1}^{d}\beta_i, g\right) = \pi_{i=1}^{d} e(\pi_{l\epsilon L} H(id_l)^{nl}.w^{\alpha i}, pk_i)$$

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.

ACKNOWLEDGEMENT

I would like to thanks my college"Dhole Patil College of Engineering" for providing a strong platform to develop the skills and capabilities.

REFERENCES

[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.*