Enabling Public Verification and Prove Shared Data Consistency in Cloud

[1]R. Jayaraghavan, [2]A. Selvakumar
[1]P.G. Student, [2]Assistant Professor,
Department of Computer Science and Engineering,
SRM University, Chennai, India

***Abstract* - The cloud service allows users to store and share their data with relative ease. Unfortunately these data are stored in untrusted servers and maintaining integrity of these shared data is skeptical, also the system hardware and software failure also affect the data integrity. There are several mechanisms defined to allow the public auditor to verify the integrity of data, but these existing mechanisms reveal identity of data owners. By using ring signature mechanism, the public auditor can audit the shared data without compromising user identity. The athos (AuTHenticated Outsourced Storage) is a platform independent architecture that allows authentication of outsourced data in untrusted storage. In this paper we utilize both ring signature and athos architecture for efficient public auditing and ensuring data freshness (latest version of stored data) of shared data.**

***Index Terms* - Ring signature, athos (AuTHenticated Outsourced Storage), Data freshness, public auditing.**

I. INTRODUCTION

Cloud service offers a scalable, secure and reliable environment for users to share their data at a much lower marginal cost due to the sharing nature of resources. As data stored in an untrusted cloud it can easily be lost or corrupted, due to hardware failures and human errors [6]. Operating on remotely managed data entails security risks: when the storage provider is not trusted by the data source, verifying the integrity of the stored data and the correctness of the computations performed on this data is necessary to ensure the trustworthiness of the storage system. The integrity can be verified by using Third party auditor (TPA) [1], who offers its auditing service with more powerful computation and communication abilities than regular users. The identities of signers on shared data may indicate which user in the group or block in shared data is a higher valuable target than others. Such information is confidential to the group and should not be revealed to any third party

Through the use of homomorphic ring signatures [4, 7], the third party auditor is able to verify the integrity of shared data for a group of users without retrieving the entire data there by allowing user confidentiality. The use random masking [9] allows data privacy during public auditing, and leverage index hash tables [7] to support fully dynamic operations on shared data. i,e.,When a user (either the original user or a group user) wishes to check the integrity of shared data, she first sends an auditing request to the TPA. After receiving the auditing request, the TPA generates an auditing message to the cloud server, and retrieves an auditing proof of shared data from the cloud server. Then the TPA verifies the correctness of the auditing proof. Finally, the TPA sends an auditing report to the user based on the result of the verification.

The Authentication service [10] allows data freshness verification by allow the client to efficiently verify the integrity of a dynamically evolving file system, namely to verify that its status is consistent with the exact history of file-system operations requested by the client, and to correctly detect any malicious data-update or data-retrieval patterns produced by the server. and thus prevents rollback attacks reverting the file system state to a previous version [11].

There are two kinds of threats persisting in untrusted store that must be addressed.
Integrity Threat: There are two ways data integrity can be compromised, one way is an adversary may try to corrupt the integrity of shared data and prevent users from using data correctly. Second, the cloud service provider may inadvertently corrupt data.
Privacy Threat: The identity of the signer on each block in shared data is private and confidential to the group. During the process of auditing, a semi-trusted TPA, who is only responsible for auditing the integrity of shared data, may try to reveal the identity of the signer on each block in shared data based on verification information.

This Paper is organized as follows. Section II describes related works regarding public verification of shared data. Section III describes proposed system model and design objectives. Section IV describes in depth details of system architecture. Finally we conclude our work and provide scope for future work in Section V.

II. RELATED WORK

There are several mechanism through which public auditor can audit the stored data. One of the mechanisms is Provable Data Possession [2]. It allows the auditor to audit the data by downloading entire data block. This will invariably reveal the identity privacy of owner. Alternatively group signatures can be utilized using linear ring signature. This method preserves identity privacy of data owner. This method will not support blockless auditing (batch auditing), so it will increase processing overhead.

To offset inefficiency of the conventional methods several other approaches are defined. One of the designs is Global private key [3]. This method allows every user to sign the data block using global private key, thereby preserving identity of data user. One of the disadvantages of this method is if one of the group members is compromised or left the group then new private key must be generated and securely shared among group user which will introduce new overhead.

Another method, Trusted Proxy [4] allows the user to sign and upload their data to this trusted proxy. The public auditor can verify and find only its proxy sign of data, but this method is vulnerable to single point failure of proxy. Alternatively Direct Anonymous Attestation [5] which will protect their identity but its disadvantage is requires both client and service provider to migrate to Trusted Computing Platform which is costly and impractical

III. SYSTEM MODEL

The System Model consists of four components: the Group Users, a public auditor, a cloud server and Authentication System. There are two types of users in group, one is data owner who created the data and share it. Another is group of user who can access and modify the shared data. The Public Auditor is a third party verifier who will audit the integrity of shared data. These shared data and its metadata are stored in cloud servers. The Authentication System is store and verifies metadata of stored data.

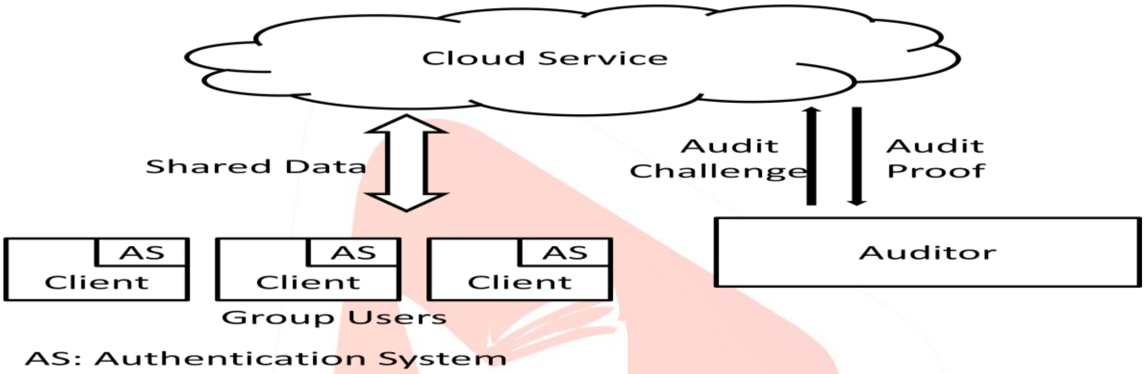

Fig 1 System Model

The proposed system mainly focused to satisfy following objectives:

1. Public Verification: Enable auditor to publicly audit data without compromising privacy
2. Efficiency: Ability to verify multiple data block (Batch auditing)
3. Unforgeability: only a user in a group can generate verification meta data
4. Identity Privacy: Auditor cannot identify the identity of signer of data block during auditing process.
5. Data freshness: Ensure cloud possess latest version of data.

IV. SYSTEM ARCHITECTURE

4.1 Auditing Service:

Auditing service utilize homomorphic ring signatures to hide the identity of the signer on each block, so that private and sensitive information of the group is not disclosed to the TPA This mechanisms is based on two properties:

Blockless Verification: Given two block b1 and b2, two random values v1,v 2 and a block m = v1m1 + v2m2 , A verifier is able to check the correctness of block m without knowing block m1 and m2.

Non-malleability: Given two block b1 and b2, two random values v1, v2 and a block m = v1m1 + v2m2 2, A user, who does not have private key **sk**, is not able to generate a valid signature b on block m by linearly combining signature b1 and b2

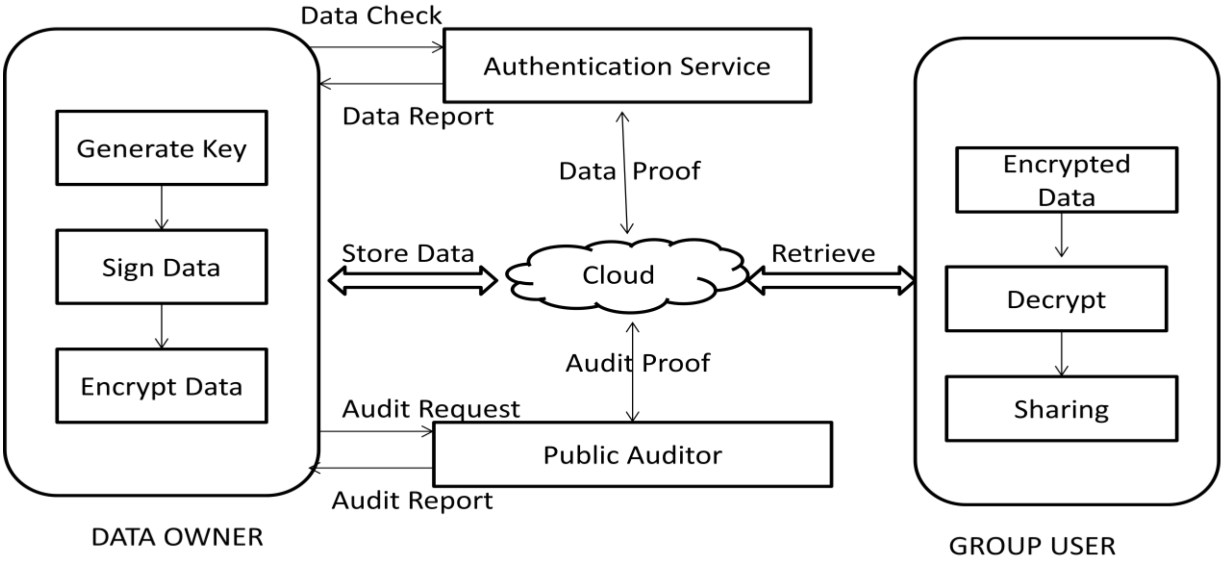

Fig. 2 System Architecture

This ring signature mechanism consists of four algorithms: KeyGen, RingSign**,** RingVerify, AuditVerify. Let G1, G2 and GT be multiplicative cyclic groups of order p, g1 and g2 be generators of G1 and G2 respectively. Let e: G1 × G2 ->GT be a bilinear map, and D: G2 ->G1 be a computable isomorphism with D(g2) = g1. There is a public map-to-point hash function H1: {0, 1}->G1. The global parameters are
(e,D,p,G1,G2,GT,g1,g2,H1). The total number of users in the group is d. Let U denote the group that includes all the d users.

KeyGen*:* Each user in the group generates her public key(Pk) and private key(Sk)

RingSign: A user in the group is able to sign a block with her private key and all the group members’ public keys.

RingVerify**:** A verifier is allowed to check whether a given block is signed by a group member.
Given all the d users’ public keys

AuditVerify: To audit the integrity of shared data, a user first sends an auditing request to the auditor. After receiving an auditing request, the Auditor generates an auditing message and sends to cloud server. The cloud server sends audit proof (proof of possession + aggregate public key for selected blocks). On receiving audit proof, auditor again recalculates it using public key of group members. If calculated result and audit proof are same then data integrity is preserved, else data block is corrupted.

4.2 Authentication Service:

A client C owns an outsources a file system FS to an untrusted server S. In additional to the file system, Server S hosts and controls an authentication service module A that stores authentication information about FS. The file system is generated and queried through a series of update and query operations issued by the client C. At any time, C keeps some state information s that encodes information about the current state of FS. If P is the set of operations supported over the file system, then the communication protocol is as follows:

1. Client C keeps state information s and issues a query or update operation o ∈ P to the server S.

2. Server S runs a certification algorithm, which performs operation o and accordingly answers the query or updates FS to a new version FS′, and, by using A, also generates a verification or respectively consistency proof π which is returned to C, along with the result ρ of the operation; ρ is the Corresponding answer if o is a query operation or else the empty string ⊥. We write π ← certify (o, FS, FS′, ρ).

3. Client C runs a verification algorithm, which takes as input the current state s, the operation o along with its result ρ, and the corresponding (consistency or verification) proof π and either accepts or rejects the input. If the input is accepted the state s is appropriately updated to state s′, where s′ = s if o is a query operation or else s′ 6= s. We write {(yes, s′), (no, ⊥)} ← verify(s, ρ, π).

Correctness***:*** For any o ∈ P, when (FS′τ , ρ) ← operate(o, FSτ), it holds that (yes, s′) ← verify(s, ρ, certify(o, FSτ , FS′τ , ρ)). I.e., for any correctly per- formed operation, certify generates a proof that is always accepted by verify, which also computes a new, consistent with the new file system FS, state s′.
If metadata block of file system holds the above condition, then it possesses the latest version of shared data i.e., data correctness.

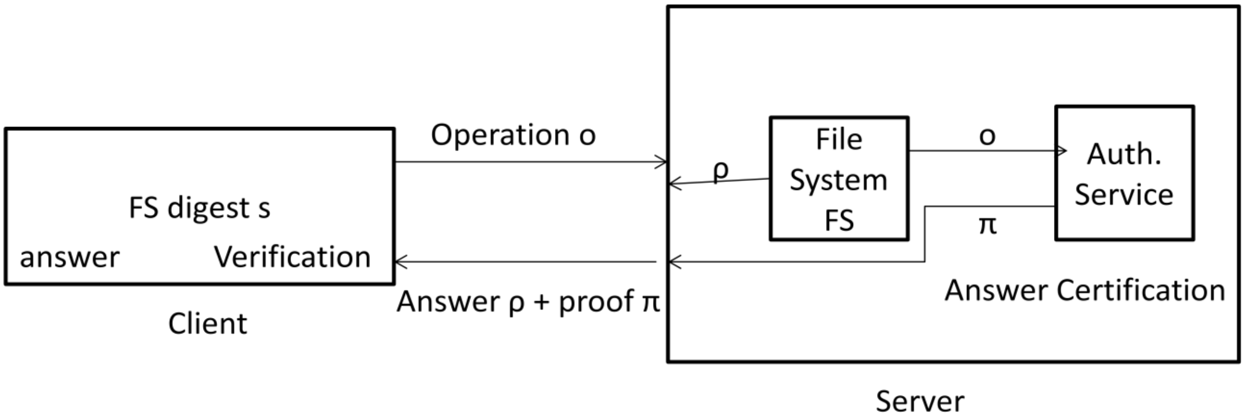

Fig. 3 Authentication Service Architecture

V. CONCLUSION AND FUTURE SCOPE

Our proposed auditing mechanism thus combines ring signature scheme and athos architecture to enable robust integrity verification and data freshness verification. The batch auditing due to blockless verification property of ring signature allows efficiency and reduce processing overhead. The Data freshness allows the cloud to avoid roll-back attacks. There is two issues that will be continued for future study. One of them is traceability i.e., ability of group manager to identify the data signer based on verification metadata. The another is ability to efficiently revocate user in a dynamic group without increasing computation and communication overhead

REFERENCES

[1] B. Wang, B. Li, and H. Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,” Proc. IEEE Fifth Int’l Conf. Cloud Computing, pp. 295-   302, 2012.

[2] 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), pp. 598-610, 2007.

[3] B. Wang, S.S. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” Proc. IEEE 33rd Int’l Conf. Distributed Computing Systems (ICDCS’13), pp. 124-133, 2013.

[4] D. Boneh, X. Boyen, and H. Shacham, “Short Group Signatures,” Proc. 24th Ann. Int’l Cryptology Conf. (CRYPTO’04), pp. 41-55, 2004

[5] E. Brickell, J. Camenisch, and L. Chen, “Direct Anonymous Attestation,” Proc. 11th ACM Conf. Computer and Comm. Security (CCS’04), pp. 132-145, 2004.

[6] 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, April 2010

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 525–533.

[8] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” in *Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*. Springer-Verlag, 2008, pp. 90–107.

[9] K. Ren, C. Wang, and Q. Wang, “Security Challenges for the Public Cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[10] R. C. Merkle. A certified digital signature. In Proc. CRYPTO, pp. 218–238, 1989.

[11] K. Fu, M. F. Kaashoek, and D. Mazi`eres. Fast and secure distributed read-only file system. ACM Trans. Computer Syst., 20(1):1–24, 2002

[12] C. Cachin, A. Shelat, and A. Shraer. Efficient fork-linearizable access to untrusted shared memory. In Proc. Principles of Distr. Computing, pp. 129–138, 2007.