Preserving Privacy of Cloud Data in Public Auditing (IJLTEMAS)

Shefali Arora/University of Mumbai
Saraswati College of Engineering
Kharghar, Navi Mumbai
Shefali.cse@gmail.com

Prof. Dipti Patil/ University of Mumbai
Pillais Institute of Information Technology
Panvel, Navi Mumbai
Dypatil75@gmail.com

Absract- **Cloud computing technology has been looked upon as the next-generation architecture of IT solution. It enables the users to move their data and application software to the network which is different from traditional solutions. The distributed architecture of cloud data storage facilitates the customer to get benefits from the greater quality of storage and minimized the operating cost. Due to this IT services are not under logical, physical and users' controls, it brings many new different security challenges. Ensuring data storage security is one more urgent of them. The representative network architecture for cloud data storage includes a third party auditor which affords trustful authentication for user to operate their data security in cloud. This technology also brought numerous possible threats including data confidentiality, integrity and availability. To address these problems a homomorphic based model of storage is proposed, which enable the customer and a third party auditor to perform the authentication of data stored on the cloud storage. This model performs the verification of huge file's integrity and availability with less consumption of computation, storage and communication resources.**

Keywords-Cloud Computing; TPA; Data Integrity; Homomorphic authenticator

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet based development and use of computer technology. The cheaper and more powerful processors, together with the software as a service (SaaS) [3] computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable, flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside exclusively on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. Storing data remotely into the cloud in a flexible on demand manner brings appealing benefits: 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.

Cloud computing is gaining popularity due to its cost effectiveness in service provisioning. But it poses many challenges on integrity and privacy of users' data though it brings an easy, cost-effective and reliable way of data management. Hence, secure and efficient methods are needed to ensure integrity and privacy of data stored at the cloud.We propose a Third party auditor(TPA) between data owner and cloud service provider(CSP) which reduce the burden of data owner to audit the data in the cloud and it also make the data owner free from worrying about the data lose in cloud storage .

The obvious advantage of our scheme is the cloud service provider can offer the functions which were provided by the traditional third party auditor and make it trustful. So it indeed reduces the user's complexity in Cloud Computing.

We utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique and automatic blocker to achieve a privacy-preserving public auditing system for cloud data storage security. We are using automatic blocker to the cloud environment, which particularly blocks the auditing protocols from unauthorized access from the external user for privacy preserving for data.

II. RELATED WORK

The problem of remote data integrity checking is first introduced in [6], which independently propose RSA based methods for solving this problem. After that Shah et al. [7] propose a remote storage auditing method based on pre-computed challenge-response pairs. These protocols all provide good efficiency and security. However, none of them provides public verifiability or data dynamics.

Recently, Ateniese et al. [4] propose two provable data possession (S-PDP, E-PDP) schemes to provide integrity protection for remote data. The S-PDP and E-PDP support data block append operation, and a variant of their main PDP scheme has public verifiability. Seb_e et al. [1] propose a remote data possession checking protocol for critical information infrastructures. Their protocol supports unlimited times of file integrity verifications and has a trade off between the running time and the storage cost at the verifier. Their protocol can be easily adapted to support data dynamics, but it doesn't support public verifiability. Erway et al. propose two efficient PDP constructions with data dynamics by using rank-based skip lists and RSA trees . Wang et al. propose a method which uses merkle hash tree [8] to support fully data dynamics and uses BLS signature [9] to support public verifiability. Wang and Sherman *et al.* [5] have proposed a public auditing system of data storage security by developing a privacy preserving auditing protocol. By which auditor can audit without having knowledge of user's data contents. Wang and Sherman also proposed a batch auditing protocol where multiple auditing tasks from different users can be performed simultaneously by a TPA. Zhu et al. [10] propose a formal framework for interactive provable data possession (IPDP) and a zero-knowledge IPDP solution for private clouds. Their ZK-IPDP protocol achieves probabilistic data possession guarantee, supports fully data dynamics, public verifiability and is also private against the verifiers.

Further Nandeesh et al [5]. carried out future work on the physical possession of the outsourced data in cloud computing storage creates new security risk. To secure TPA based storage using homomorphic tokens and distributed erasure coded data, which allow to audit the cloud storage with minimum computation cost. To achieve efficient data dynamic operations, we improve the storage on outsourced data including data modification, deletion and updation. Mururalikrishnan Ramane et al. [4] studied further about the public auditing schemes are used efficiently in auditing the data stored in cloud, it solves the issue of restricting TPA to access of the data openly. This scheme verifies the metadata rather than actual data which provides secure cloud storage that supports privacy preserving public auditing. Dalia Attas et al [11]. studied further on cloud computing to ensure the integrity of the data stored in the cloud storage, TPA supported with digital signature is used for efficient auditing. This doesn"t affect the original data and also audits without demanding local copy of data. Checking is done in the cloud service provider(CSP) and TPA. The digital signature first performs hash function using message-digest algorithm(MD5). Compute encryption with private key on the other hand decryption by using public key with hash value containing reverse order of its original data.

Our proposed system enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee:

1. Public Auditability
to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users.
2. Storage Correctness
to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact.
3. Privacy-Preserving
to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process;
4. Batch Auditing
to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simul-taneously.
5. Lightweight
to allow TPA to perform auditing with minimum communication and computation overhead.

III. PROPOSED MODEL

This section presents our public auditing scheme for cloud data storage security.

The System and Threat Model: We consider a cloud data storage service involving three different entities, as illustrated in fig. 1: the cloud user (U) or data owner, who has large amount of data files to be stored in the cloud; the Cloud Server , which is managed by Cloud Service Provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.); the Third Party Auditor

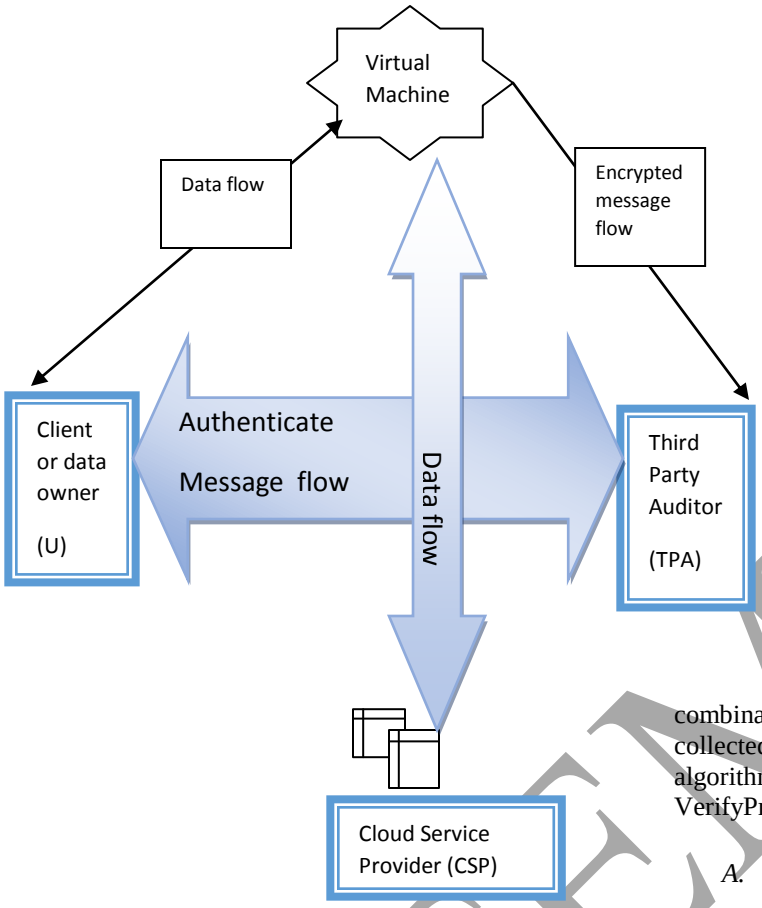

Fig 1. How data flows between CSP, TPA and Client

(TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request.[1]

By integrating the homomorphic authenticator with random mask technique, our protocol guarantees that 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 benefit our design for the batch auditing.

With Random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected.. A public auditing scheme consists of five algorithms (KeyGen, SigGen, GenProof, VerifyProof, protocol verifier).

A. Algorithms:

1. KeyGen Algorithm

KeyGen is a key generation algorithm that is run by the user to setup the scheme.

2. SigGen Algorithm

SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing.

3. GenProof Algorithm

GenProof is run by the cloud server to generate a proof of data storage correctness.

4. VerifyProof Algorithm

VerifyProof is run by the TPA to audit the proof from the cloud server

5. Protocol Verifier Algorithm

Protocol verifier is used by the cloud server.

B. Setup

The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate

the verification metadata. The user then stores the data file F at the cloud server, delete its local copy, and publish the verification metadata to TPA for later audit. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

C. Audit

The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F by executing GenProof. Using the verification metadata, the TPA verifies the response via VerifyProof.

The BLS signature scheme uses a cryptographic primitive called pairing, is defined as a map over two cyclic groups G1 and G2. The BLS signature scheme consists of three phases:
a) In the key generation phase, a sender chooses a random integer $x \in Zp$ and computes $y = g1^x \in G1$. The private key is x and public key is y.
b) Given a message $m \in \{0.1\}$ in the signing phase, the sender first computes $h = h(m) \in G1$, where h () is a hash function, and then computes $\sigma = hx \in G1$.The signature of m is σ.
c) In the verification phase, the receiver first computes $h = h(m) \in G1$ and then check whether $e(h,y) = e(\sigma,g1)$. If the verification succeeds, then the message m is authentic.

So the benefit we get is generation of a very short signature and also can resolve communication overhead.

D. PBlocker

Once the user initializes the parameters the system checks all the specified parameters and validates the protocol for proper users, it blocks the unauthorized users -if the user newly access the cloud servers, the system prompts for security parameters, previously assigned by the system during the user creation.

E. Batch Auditing Module

With the establishment of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different users' requests. The individual auditing of these tasks for TPA can be tedious and very inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side. Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. We can adopt this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics.

IV. UTILIZING HOMOMORPHIC AUTHENTICATORS

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers. [12] To significantly reduce the arbitrarily large communication overhead for public auditability without introducing any online burden on the data owner, we resort to the homomorphic authenticator technique as shown in fig. 3.[2] Suppose, for instance, that the task you've outsourced to the cloud is to search a huge encrypted database for the handful of records that match an encrypted search term. Homomorphic encryption ensures that the server has no idea what the search term is or which records match it. As a consequence, however, it has no choice but to send back information on every record in the database. The user's computer can decrypt that information to see which records matched and which didn't, but then it's assuming

much of the computational burden that it was trying to offload to the cloud in the first place.

Homomorphic Authenticators.: A homomorphic authenticator scheme consists of the probabilistic-polynomial time algorithms (KeyGen; Auth; Ver; Eval) with the following syntax:
-KeyGen(1^n) -> (evk, sk): Outputs the secret key sk and an evaluation key evk.
-$Auth_{sk}$(b, r) -> σ: Creates a tag _ that authenticates the bit b 2 {0, 1} under the label r ε {0, 1}*. (Equivalently, we say that σ authenticates b as the output of the identity program Ir .)
-$Eval_{evk}$ (f,σ) -> ψ : The deterministic evaluation procedure takes a vector of tags σ = (σ_1, ….,σ_k) and a circuit f : $\{0, 1\}^k$ -> {0, 1}. It outputs a tag ψ . If each σi authenticates a bit bi as the output of some labeledprogram Pi (possibly the identity program), then ψ should authenticate b* = f(b1, …., bk) as the output of the composed program P* = f(P1, …., Pk).
- Ver_{sk}(e,P,ψ) -> (accept, reject): The deterministic verification procedure uses the tag to check that e ε {0, 1} is the output of the program P on previously authenticated labeled data.

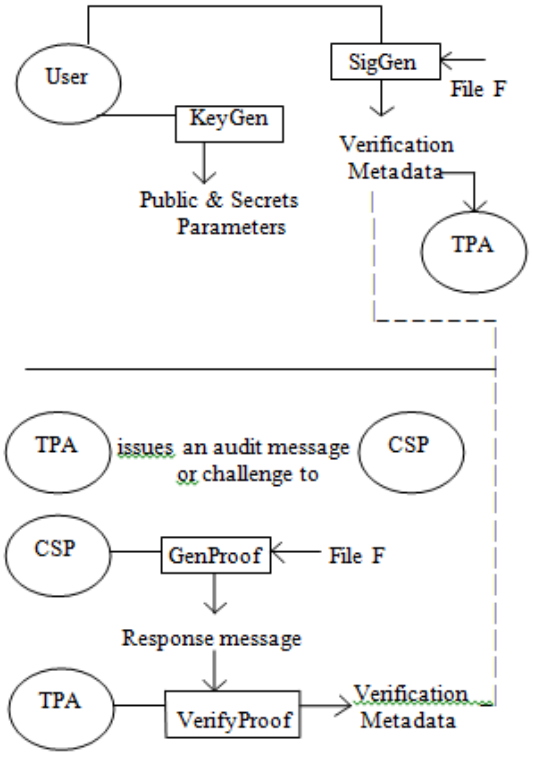

Fig.2 Set-up and Audit Phase

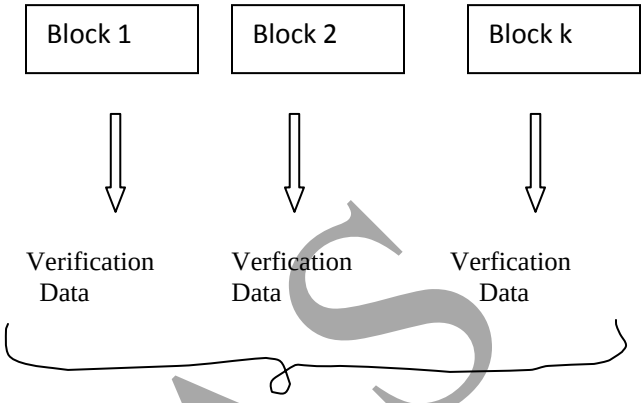

Fig.3 Homomorphic Authenticator

FUTURE WORK

Best practices in cloud computing architectures will continue to evolve and as researchers, we should focus not only on enhancing the cloud but also on building tools, technologies and processes that will make it easier for developers and architects to plug in applications to the cloud easily.

CONCLUSION

The public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the outsourced data when needed. This work studies the problem of ensuring the integrity of data storage in cloud client, to verify the integrity of the dynamic data stored in the cloud. We utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. This scheme is the first to support scalable and efficient public auditing in the Cloud Computing.
The technique of Bilinear Aggregate signature is used to achieve batch auditing, where TPA can perform multiple auditing tasks simultaneously.

ACKNOWLEDGMENT

I would like to show my gratitude to Prof. Dipti Patil who encourage me from the start to the final level to build this fine piece of paper. Her guidance has been the constant driving force behind my preparation to this Paper.

REFERENCES

[1] M.Vanitha, R.Raju, " Data Sharing: Efficient Distributed Accountability in Cloud Using Third Party Auditor", (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, April 2013

[2] Balakrishnan.S, 2Saranya.G, 3Shobana.S, 4Karthikeyan.S, "Introducing Effective Third Party Auditing (TPA) for Data Storage Security in Cloud", IJCST Vol. 2, Issue 2, June 2011

[3] Qian Wang and Cong Wang and Kui Ren, Wenjing Lou, Jin Li "Enabling Public Auditability And Data Dynamics For Storage Security in Cloud Computing" in IEEE transactions on parallel and distributed systems, 2011, vol. 22, no. 5

[4] Muralikrishnan Ramane and Bharath Elangovan, "A Metadata Verification Scheme for Data Auditing in Cloud Environment", International Journal on Cloud Computing: Services and Architecture(IJCCSA), Vol.2, no.4, August 2012.

[5] Wang, Sherman, Kui, Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage", INFOCOM, 2010 Proceedings IEEE, 14-19 March, 2010.

[6] Y. Deswarte and J.-J. Quisquater, "Remote Integrity Checking," in Sixth Working Conference on Integrity and Internal Control in Information Systems (IICIS) (S. J. L. Strous, ed.), pp. 1–11, Kluwer Academic Publishers, 1 2004.

[7] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS XI., Usenix, 2007.

[8] R. C. Merkle, "Protocols for public key cryptosystems," Security and Privacy, IEEE Symposium on, p. 122, 1980.

[9] R. Curtmola, O. Khan, and R. Burns, "Robust remote data checking," in StorageSS '08: Proceedings of the 4th ACM international workshop on Storage security and survivability, (New York, NY, USA), pp. 63–68, ACM, 2008.

[10] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Cooperative provable data possession." Cryptology ePrint Archive, Report 2010/234, 2010. http://eprint.iacr.org/.

[11] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in 14th European Symposium on Research in Computer Security, pp. 355–370, Springer Berlin / Heidelberg, September 2009.

[12]http://en.wikipedia.org/wiki/Homomorphic_encryption