ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

# Privacy Preserving Public Auditing for Shared Data in the Cloud

T.Magesh Kumar[1], G.Karthik Krishnan[2], Dr .V.Anjana Devi[3]

B.E, Dept. of Computer Science and Engineering, St.Joseph's College of Engineering, Chennai, Tamil Nadu, India[1]

B.E, Dept. of Computer Science and Engineering, St.Joseph's College of Engineering, Chennai, Tamil Nadu, India[2]

Associate Professor, Dept. of Computer Science and Engineering, St.Joseph's College of Engineering, Chennai, Tamil Nadu, India[3]

**ABSTRACT:** The novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, this system exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. Within mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, the mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. The propose system a privacy-preserving public auditing mechanism for shared data in the cloud. Utilize ring signatures to construct shared data integrity without retrieving the entire data, it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, they further extend our mechanism to support batch auditing. There are two interesting problems the will continue to study for our future work. One of them is traceability, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations.

**KEYWORDS:** Cloud storage, public auditing, Ring signature, Quantum computer attacks.

## I. INTRODUCTION

Cloud storage is an important service of cloud computing, which allows data owners to move data from their local computing systems to the Cloud server (CS). By hosting their data in the cloud server, data owners can avoid the initial investment of expensive infrastructure setup, large equipment's, and daily maintenance cost. Moreover, data owners can rely on the cloud to provide more reliable services, so that they can access data from anywhere and at any time. While cloud storage service makes these advantages more appealing than ever before, it also brings new and challenging security threats towards users' outsourced data. For example, the Cloud service provider (CSP) may hide data blocks loss accidents to maintain the reputation, or discard the data blocks which have not been rarely accessed to save storage space [1]. Therefore, it is desirable to have data integrity verification service to assure data owners that their data blocks are correctly stored in the cloud server.

A Third party auditor (TPA) is a natural choice for the public auditing of cloud storage data. A third party auditor who has expertise and capabilities can efficiently verify the integrity of the data stored in the cloud server on behalf of the data owner. Recently, some novel and efficient public auditing protocols [2–7] have been proposed to ensure the integrity of the data blocks stored in the cloud server. With the advent of the post-quantum cryptographic era, lattice- based cryptography has been regarded as the most attractive option for resisting quantum attacks. Some lattice-based linearly homomorphic signature schemes have been proposed [8, 9] so far. These schemes can be applied to construct the public proof protocol of cloud storage data. In 2014, Liu *et al.* proposed public proof of cloud storage from lattice assumption [10], their protocol employs the additive homomorphic hash function which is defined in Ref. [11]. They claimed that their protocol satisfies the security properties: unforge ability, public verifiability and privacy preserving in their secure model. These properties ensure that their protocol can successfully execute the public proof for cloud storage data.

DOI: 10.15680/IJIRCCE.2018.0603005

## II. RELATED WORK

The most challenging to this topic of cloud data sharing. They limited people ability will be there in this method. Shared date to encrypt to public auor there will be seeing in the data owner and user and then administrator process also cover. To any vulnerable access are attack they immediately action to remove the particular member are user. However the data will be encrypted to 128 bit of converted to store in the cloud server. They have split to suppurated private key send to the particular user they while be access to the random key enter then using the particular file. Any one cannot been attack outside hacker. The proposed system to use this privacy-preserving mechanism will be used and then reduced to outside attack. Before that process to outside attack is possible in this method. But current process outside and inside it’s also not possible in this concept. The novel privacy process also used in this mechanism. It is more secure to this data in this shared process. They MD-5 techniques and then ring signature also used in this concept. Administrator was control to overall process. They will approved to the data owner, auditor also permission apply to the admin. New user is also approved to send the random key. To access public access to the user in the particular group. Any can misuser this file are change at the same time stop the overall process then inform to the admin. Auditor re-upload the file.

## III. OVERVIEW OF OUR APPROACH

Before introducing the underlying concepts and algorithms, the overview of our dictionary based provenance scheme is discussed below. The first briefly introduce public cloud storage auditing system model as is illustrated in Fig.1. It consists of four different entities, they are cloud user, Data owner, Auditor and cloud service provider respectively, which are well defined in Ref.[10]
Theorems about the AES algorithms from lattice assumption as introduced in Ref.[12]. And refer the readers to the detailed definitions of lattice, hard problems on lattice in Ref.[10].To represented in fig 1.1.

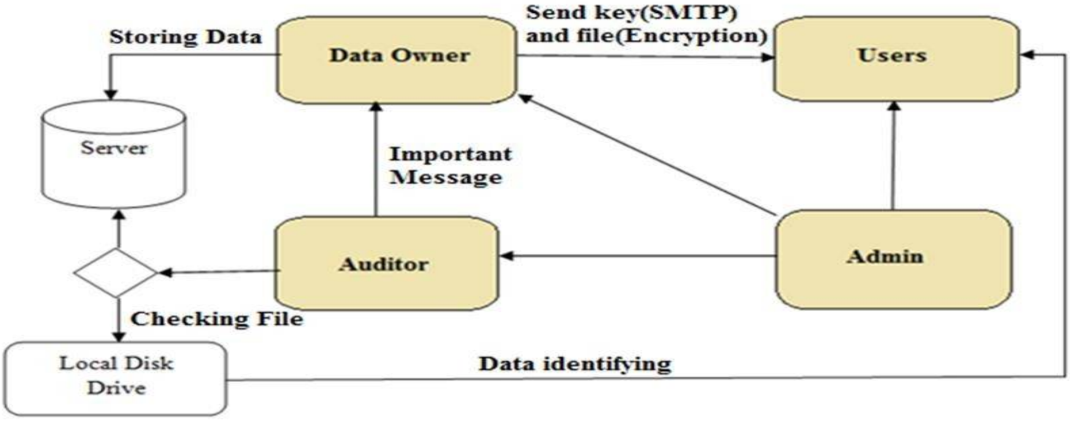


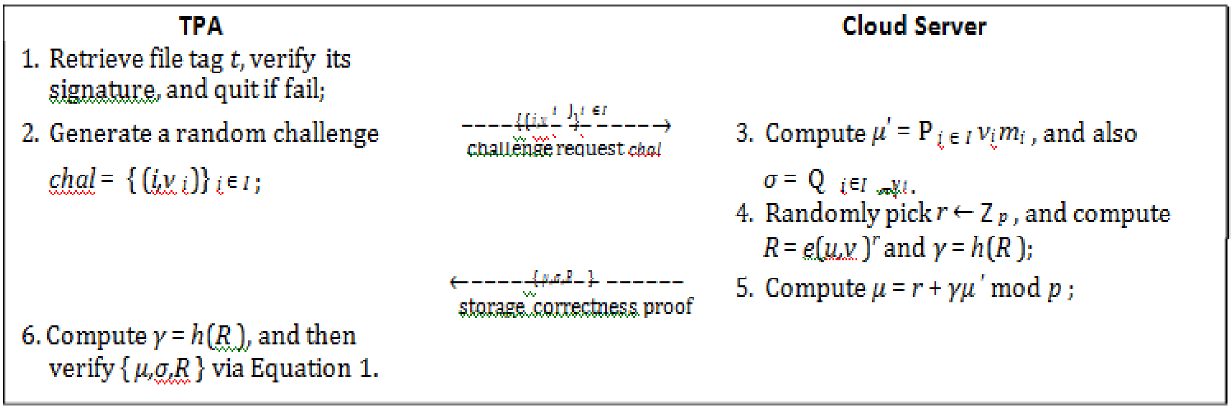
**Figure:1.1** Shared data in the cloud

**Explanation--**Cloud storage enables users to remotely store their data and enjoy the on demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing user's physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, The propose in this project a flexible distributed storage integrity auditing mechanism, utilizing the homomorphism token and distributed erasure-coded data.

## IV. THE PRIVACY-PRESERVING PUBLIC AUDITING SCHEME

To achieve privacy-preserving public auditing, the uniquely integrate the homomorphic linear authenticator with random masking technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated the server. 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. . Specifically, they use the HLA proposed in [13], which is based on the short signature scheme proposed by Boned, Lynn and Sachem (hereinafter referred as BLS signature) [17].**4.1Scheme Details.** Let $G_1$, $G_2$ and $G_T$ be multiplicative cyclic groups of prime order $p$, and $e : G_1 \times G_2 \rightarrow G_T$ be a bilinear map as introduced in preliminaries. Let $g$ be a generator of $G_2$. $H(\cdot)$ is a secure map-to point hash function: $\{0,1\}^* \rightarrow G_1$, which maps strings uniformly to $G_1$. Another hash function $h(\cdot) : G_T \rightarrow Z_p$ maps group element of $G_T$ uniformly to $Z_p$. The proposed scheme is as follows:

Setup Phase: The cloud user runs KeyGen to generate the public and secret parameters. Specifically, the user chooses a random signing key pair $(spk,ssk)$, a random $x \leftarrow Z_p$, a random element $u \leftarrow G_1$, and computes $v \leftarrow g^x$. The secret parameter is $sk = (x,ssk)$ and the public parameters are $pk = ((spk,v,g,u,e(u,v))$.



TPA

1. Retrieve file tag $t$, verify its signature, and quit if fail;
2. Generate a random challenge $chal = \{(i,v_i)\}_{i \in I}$;

Cloud Server

3. Compute $\mu' = P_{i \in I} v_i m_i$, and also $\sigma = Q_{i \in I}$ $\sigma_i^{v_i}$.
4. Randomly pick $r \leftarrow Z_p$, and compute $R = e(u,v)^r$ and $\gamma = h(R)$;
5. Compute $\mu = r + \gamma\mu' \bmod p$;

6. Compute $\gamma = h(R)$, and then verify $\{\mu,\sigma,R\}$ via Equation 1.

Fig. 2: The privacy-preserving public auditing protocol

Given a data file $F = (m_1,...,m_n)$, the user runs SigGen to compute authenticator $\sigma_i$ for each block $m_i$: $\sigma_i \leftarrow (H(W_i) \cdot u^{m_i})^x \in$
$G_1$. Here $W_i = name||i$ and $name$ is chosen by the user uniformly at random from $Z_p$ as the identifier of file $F$. Denote the set of authenticators by $\Phi = \{\sigma_i\}_{1 \le i \le n}$. The last part of SigGen is for ensuring the integrity of the unique file identifier $name$. One simple way to do this is to compute $t = name||\mathrm{SSig}_{ssk}(name)$ as the file tag for $F$, where $\mathrm{SSig}_{ssk}(name)$ is the signature on $name$ under the private key $ssk$. For simplicity, we assume the TPA knows the number of blocks $n$. The user then sends $F$ along with the verification metadata $(\Phi,t)$ to the server and deletes them from local storage.

**Audit Phase:** The TPA first retrieves the file tag $t$. With respect to the mechanism they describe in the Setup phase, the TPA verifies the signature $\mathrm{SSig}_{ssk}(name)$ via $spk$, and quits by emitting FALSE if the verification fails. Otherwise, the TPA recovers $name$. Now it comes to the "core" part of the auditing process. To generate the challenge message for the audit "$chal$", the TPA picks a random $c$-element subset $I = \{s_1,...,s_c\}$ of set $[1,n]$. For each element $i \in I$, the TPA also chooses a random value $v_i$ (of bit length that can be shorter than $|p|$, as explained in [13]). The message "$chal$" specifies the positions of the blocks that are required to be checked. The TPA sends $chal = \{(i,v_i)\}_{i \in I}$ to the server.

Upon receiving challenge $chal = \{(i,v_i)\}_{i \in I}$, the server runs GenProof to generate a response proof of data storage correctness. Specifically, the server chooses a random element $r \leftarrow Z_p$, and calculates $R = e(u,v)^r \in G_T$. Let $\mu'$ denote the linear combination of sampled blocks specified in $chal$: $\mu' = P_{i \in I}\, v_i m_i$. To blind $\mu'$ with $r$, the server computes: $\mu = r + \gamma\mu'$ mod $p$, where $\gamma = h(R) \in Z_p$. Meanwhile, the server also calculates an aggregated authenticator $\sigma = Q_{i \in I}\, \sigma_i^{v_i} \in G_1$. It then sends $\{\mu,\sigma,R\}$ as the response proof of storage correctness to the TPA. With the response from the server, the TPA runs VerifyProof to validate the response by first computing $\gamma = h(R)$ and then checking the verification equation

$$R \cdot e(\sigma^{\gamma}, g) \stackrel{?}{=} e\Big(\big(\prod_{i=s_1}^{s_c} H(W_i)^{v_i}\big)^{\gamma} \cdot u^{\mu}, v\Big) \qquad (1)$$

$$R \cdot e(\sigma^{\gamma}, g) = e(u,v)^{r} \cdot e\Big(\big(\prod_{i=s_1}^{s_c} (H(W_i) \cdot u^{m_i})^{x \cdot v_i}\big)^{\gamma}, g\Big)$$

$$= e(u^{r}, v) \cdot e\Big(\big(\prod_{i=s_1}^{s_c} H(W_i)^{v_i} \cdot u^{v_i m_i}\big)^{\gamma}, g\Big)^{x}$$

$$= e(u^{r}, v) \cdot e\Big(\big(\prod_{i=s_1}^{s_c} H(W_i)^{v_i}\big)^{\gamma} \cdot u^{\mu'\gamma}, v\Big)$$

$$= e\Big(\big(\prod_{i=s_1}^{s_c} H(W_i)^{v_i}\big)^{\gamma} \cdot u^{\mu'\gamma + r}, v\Big)$$

## V. COST OF PRIVACY-PRESERVING PROTOCOL

There begin by estimating the cost in terms of basic cryptographic operations, as notated in Table 1. Suppose there are *c* random blocks specified in the challenge

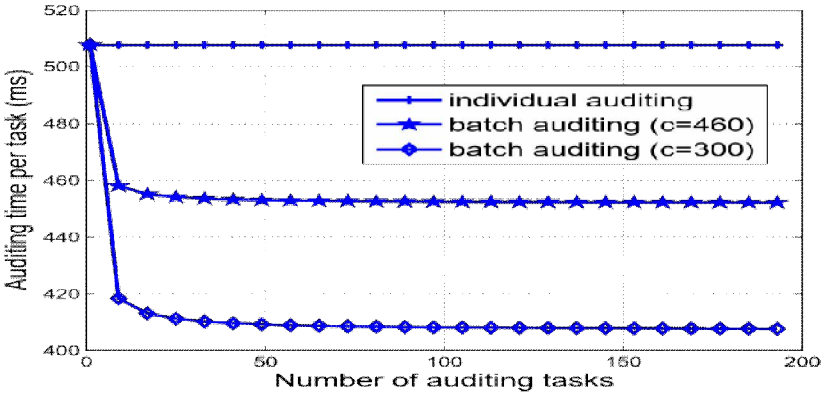


Fig. 4: Comparison on auditing time between batch and individual auditing.

Per task auditing time denotes the total auditing time divided by the number of tasks. For clarity reasons, we omit the straight curve for individual auditing when *c*=300.message *chal* during the Audit phase. Under this setting, we quantify the cost introduced of the privacy-preserving auditing in terms of server computation, auditor computation as well as communication overhead.On the server side, the generated response includes an aggregated authenticator *σ* = Q*i*∈*I σivi* ∈ G1, a random factor *R*

= *e*(*u*,*v*) ∈G$_T$, and a blinded linear combination of sampled blocks *μ* = *γ* P$i_{\in I}$ $v_i m_i$ +*r* ∈ Z$_p$, where *γ* = *h*(*R*) ∈ Z$_p$. The corresponding computation cost is$_i$ *c-MultExp*$^1$ (|*v* |), *Exp*$^1$ (|*p*|), $Hash^{1}_{\mathbb{Z}_p}$ + $Add^{c}_{\mathbb{Z}_p}$ + $Mult^{c+1}_{\mathbb{Z}_p}$ respectively. Compared to the existing HLA-based solution for ensuring remote data integrity [13]$^1$, the extra cost for protecting the user privacy, resulted from the random mask *R*, is only a constant: $Exp^{1}_{\mathbb{G}_T}(|p|) + Mult^{1}_{\mathbb{Z}_p} + Hash^{1}_{\mathbb{Z}_p} +$

, $Add^{1}_{\mathbb{Z}_p}$

which has nothing to do with the number of sampled blocks *c*. When *c* is set to be 300 to 460 for high assurance of auditing, as discussed in Section 3.4, the extra cost for privacy-preserving guarantee on the server side would be negligible against the total server computation for response generation.

Similarly, on the auditor side, upon receiving the response {*σ*,*R*,*μ*}, the corresponding computation cost for response validation is *Hash*$^1_{Zp}$ + *cMultExp* $_i$ $^1$ (|*v* |) + *Hash*$^c$ + *Mult*$^1$ + *Mult*$^1$ +

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

$_3$ $\mathbb{G}_1$ $Pair^2$ $\mathbb{G}_1$, among which only $Hash^{1}_{\mathbb{Z}_p}$ $\mathbb{G}_1$ $\mathbb{G}_T$ +

$Exp^2$ ($|p|$) + $Mult^1$ $\mathbb{G}_1$ account for the additional constant computation cost. For $c$ = 460 or 300, and considering the relatively expensive pairing operations, this extra cost imposes little overhead on the overall cost of response validation, and thus can be ignored. For the sake of completeness, Table 2 gives the experiment result on performance comparison between our scheme and the state-of-the-art [13]. It can be

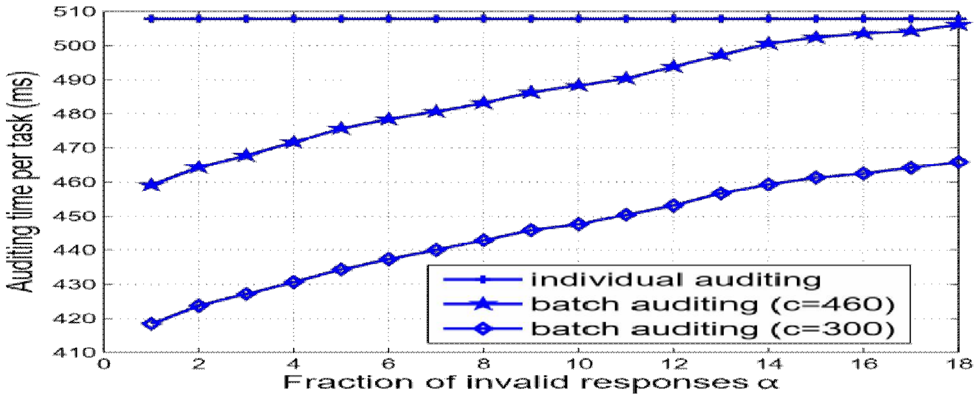


Fig. 5: Comparison on auditing time between batch and individual auditing, when $\alpha$-fraction of 256 responses are invalid.

Per task auditing time denotes the total auditing time divided by the number of tasks.

## VI. PROPOSED SCHEMAS

The propose system, a privacy-preserving public auditing mechanism for shared data in the cloud. The utilize ring signatures to construct homomorphism authenticators. So that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. The proposed system can perform multiple auditing tasks simultaneously they improve the efficiency of verification for multiple auditing tasks. High security provide for file sharing.

## VII. BENEFICIAL FEATURES

The propose system, a privacy-preserving public auditing mechanism for shared data in the cloud. The utilize ring signatures to construct homomorphism authenticators. So that a public verifier is able to audit shared data integrity without retrieving the entire data, it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, the further extend our mechanism to support batch auditing.

- The proposed system can perform multiple auditing tasks simultaneously
- They improve the efficiency of verification for multiple auditing tasks.
- High security provide for file sharing.

## VIII. CONCLUSION and FUTURE WORK

The first privacy-preserving public auditing mechanism for shared data in the cloud. With Oruta, the public verifier is able to efficiently audit the integrity of shared data, this system cannot distinguish who is the signer on each block, which can preserve identity privacy for users. An interesting problem in our future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

DOI: 10.15680/IJIRCCE.2018.0603005

**FUTURE WORK:** The system future work will be how to avoid this type of re-computation introduced by this dynamic groups while still preserving identity privacy from the public verifier during the process of public auditing on shared data.

## REFERENCES

1. Q. Wang, C. Wang, K. Ren, W. Lou and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing”, *IEEE Transactions on Parallel and Distributed Systems*, Vol.22, No.5, pp.847–859, 2011.
2. C. Wang, Q. Wang, K. Ren and W. Lou, “Towards secure and dependable storage services in cloud computing”, *IEEE Transactions on Service Computing*, Vol.5, No.2, pp.220–232, 2012.
3. C. Wang, S.M. Chow, Q. Wang, K. Ren and W. Lou, “Privacy preserving public auditing for secure cloud storage”, *IEEE Transactions on Computers*, Vol.62, No.2, pp.362–375, 2013.
4. K. Yang and X. Jia, “An efficient and secure dynamic auditing protocol for data storage in cloud computing”, *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, No.9, pp.1717–1726, 2013.
5. G.W. Solomon, C. Xu, J. Zhao and X. He, “Secure and efficient privacy-preserving public auditing scheme for cloud storage”, *Computers and Electrical Engineering*, Vol.40, No.5, pp.1703– 1713, 2014.
6. H. Wang, Q. Wu, B. Qin and J. Domingo-Ferrer, “Identity based remote data possession checking in public clouds”, *IET Information Security*, Vol.8, No.2, pp.114–121, 2014.
7. Z. Ren, L. Wang, Q. Wu and R. Deng, “Data dynamics enabled privacy-preserving public batch auditing in cloud storage”, *Chinese Journal.*