PRIVACY-PRESERVING PUBLIC AUDITING FOR SHARED DATA IN THE CLOUD

D. Mohan, M.Tech¹, S. Kalaiselvi²

¹ Department of Computer Science & Engineering, Pallavan college of Engineering, Kanchipuram, Tamil Nadu, Indiadm_mohan@yahoo.co.in

² Department of Computer Science & Engineering, Pallavan College of Engineering, Kanchipuram, Tamil Nadu, India kaviyazhini2@gmail.com

ABSTRACT
A novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our 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, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. A privacy-preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphism authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire file, 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. There are two interesting problems we 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.

I INTRODUCTION
CLOUD service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approaches. The shared file is divided into a number of small blocks, where each block is independently signed by one of the two users with existing public auditing solutions. Once a block in this shared file is modified by a user, this user needs to sign the new block using his/her private key. Eventually, different blocks are signed by different users due to the modification introduced by these two different users. Then, in order to correctly audit the integrity of the entire data, a public verifier needs to choose the appropriate public key for each block (e.g., a block signed by Alice can only be correctly verified by Alice’s public key). As a result, this public verifier will inevitably learn the identity of the signer on each block due to the unique binding between an identity and a public key via digital certificates under public key infrastructure (PKI).

To solve the above privacy issue on shared data, we propose Oruta, I a novel privacy-
preserving public auditing mechanism, public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier, cloud and DOs may distrust each other before or during their cooperation. Meanwhile, global verification gives the cloud confidence against owners who are not always behaving appropriately and eliminate the fear of being wrongly accused.

The rapid development of cloud computing, cloud storage has been accepted by an increasing number of organizations and individuals, therein serving as a convenient and on-demand outsourcing application. However, upon losing local control of data, it becomes an urgent need for users to verify whether cloud service providers have stored their data securely. In this paper, we propose an efficient public auditing protocol with global and sampling blockless verification as well as batch auditing, where data dynamics are substantially more efficiently supported than is the case with the state of the art. Note that the novel dynamic structure in our protocol consists of a doubly linked info table and a location array. Moreover, with such a structure, computational and communication overheads can be reduced substantially. Security analysis indicates that our protocol can achieve the desired properties. Moreover, numerical analysis and real-world experimental results demonstrate that the proposed protocol achieves a given efficiency in practice.

## Preliminaries

This section introduces some preliminaries to facilitate the reader’s understanding, including the bilinear pairing, computational assumption and BLS-based homomorphic verifiable authenticator.

### Bilinear Pairing and Computational Assumption

Bilinear pairing is a relatively mature technique and has been widely employed in cryptography since 2001, when an identity-based encryption was designed by Boneh et al. [33], [34], [35], [36], [37]. Certainly, such an efficient technique has been utilized in cloud auditing as well. The detailed definition is as follows.

**Definition 1 (Bilinear Pairing):** Let \( m1, m2, \) and \( n T \) be multiplicative cyclic groups of a large prime order \( p \). A map function \( e : n1 \times n2 \rightarrow nT \) is a bilinear pairing only when it satisfies the three properties below:

1. **Bilinear:** For \( \forall g1 \in n1, \forall g2 \in n2, \) and \( \forall a, b \in Zp \), there is \( e(g1a, g2b) = e(g1, g2)ab \).
2. **Non-degeneracy:** For \( \exists g1 \in n1 \) and \( \exists g2 \in n2 \), there is \( e(g1, g2)\neq 1 \).
3. **Computability:** For \( \forall g1 \in n1 \) and \( \forall g2 \in n2 \), there exists an efficient algorithm to compute \( e(g1, g2) \).

Then, the Computational Diffie-Hellman (CDH) problem is described as follows, the hardness of which is the basis of the security of our protocol.

**Definition 2 (CDH Problem):** Given \( ga \) and \( gb \), where \( g \) is a generator of \( n1 \) and \( a, b \in R Zq \), compute \( g^{ab} \).

#### BLS-based Homomorphic Verifiable Authenticator

The final preliminary is the BLS-based Homomorphic Verifiable Authenticator (BLS-HVA), which has been widely employed by a large number of cloud auditing protocols [38], [39], [40]. The following is the definition.

**Definition 3 (BLS-HVA):** Given a data file that contains \( n \) blocks, \( F = \{ m1, m2, m3, \ldots, mi, \ldots, mn \} \), let \( nT \) and \( nT \) be two
multiplicative cyclic groups of a large prime order $p$, and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear pairing. Select a random secret key $sk = a \in \mathbb{Z}_p$, and the corresponding public key is $pk = \{g, v = ga\}$. Then, BLS-HVA for each data block $mi$ is $ai = (h(mi))a$, where $h(\cdot)$ is a hash function. For verification, the auditor will simply check whether $e(Qi \in Q h(mi), v) = e(\sigma, g)$ holds, where $Q$ is the set of challenged blocks and $\sigma$ is the aggregated authenticator of these blocks’ BLS-HVA.

With such a technique, blockless verifiability can be realized. In addition, two other properties, homomorphism and non-malleability, are both possessed by BLS-HVA [17].

### III Proposed Approach

The proposed system is a privacy-preserving public auditing mechanism for shared data in the cloud. We 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. There are two interesting problems we 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.

### IV Implementation

1. **User Registration:**

   For the registration of user with identity ID the group manager randomly selects a number. Then the group manager adds into the group user list which will be used in the traceability phase. After the registration, user obtains a private key which will be used for group signature generation and file decryption.

   ```plaintext
   Registration
   ```

2. **Public Auditing:**

   Homomorphic authenticators are unforgeable verification metadata generated from individual data blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. Overview to achieve
privacy-preserving public auditing, we propose to uniquely integrate the Homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server’s response is masked with randomness generated by a pseudo random function (PRF). The proposed scheme is as follows:

- Setup Phase
- Audit Phase

3. Sharing Data:

The canonical application is data sharing. The public auditing property is especially useful when we expect the delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key.

4. Integrity Checking:

Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how 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. The user download the particular file not download entire file.

RELATED WORK

Cloud storage has been studied in recent years as one of the hot spots in cloud computing. Many branches of cloud storage, such as data auditing, privacy preservation, and dynamic updating, are the subject of intense discussion as well. For data auditing in a cloud, many protocols have been proposed in the past few years and can be divided into private protocols and public protocols. In the model of private auditing protocols, the participating entities are the DO and the CSP. Only the DO possesses the private key, and the entire auditing process is executed by the owner. However, these solutions increase the burden on DOs, who are not equipped with sufficient computing resources. Moreover, the fatal flaw is that the auditing results are unconvincing because the DO and the CSP distrust each other, and the DO is the sole source of the verification results. To remove the above doubts, a trustworthy TPA is introduced into the system. In 2007, Ateniese et al. first proposed the idea of public auditing, which is widely accepted by researchers. After that, increasingly more auditing protocols were designed based on the mechanism of “challenge-proof-verify”. In 2013, Wang et al. noted that the proposed scheme with public auditability in might leak data information. As a result, the auditing protocol in was designed to be privacy-preserving through the combination of a homomorphic linear authenticator (HLA) and a random masking technique, which is extended to support multiple users as well. One year later, Worku et al. suggested that the privacy-preserving auditing protocol in could not preserve the identity privacy of signers. Therefore, the privacy-preserving public auditing scheme proposed by Worku et al. employed a ring signature to verify data integrity without exposing signers’ identities; this was found to perform better than the work Moreover, auditing protocols were extended to key-exposure-resistant. In 2015, they gave the first solution for the exposure of clients, where stacks and binary trees were utilized to facilitate key update. The security proof and asymmetric performance evaluation in that paper showed its efficiency and security. Subsequently, they improved and perfected their auditing protocol with key-exposure which performed better in the verifiable outsourcing of key up-dates.
VI CONCLUSION

In this paper, we propose a public auditing protocol with a novel dynamic structure composed of a doubly linked info table and a location array. Compared with the state of the art, an appropriate relationship between the DLIT and the LA makes our protocol perform better both in terms of efficient dynamic support and reduced overhead. Moreover, some basic challenges in cloud auditing, such as batch auditing, block less verification and lazy update, have been overcome by our protocol. Sufficient theoretical proof indicates the security of our protocol. Extensive numerical analysis and experimental comparison results could be used to validate the performance of our protocol, making it substantially more convincing.

VII FUTURE ENHANCEMENT

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

VIII REFERENCES

- proofs of retrievability,” in Proc. of ACM CCS 2013, Berlin,