

# Privacy Preserving Public Auditing for Secure Cloud

Nithyashree D<sup>1</sup> Swetha K R<sup>2</sup>  
<sup>1</sup>Mtech, <sup>2</sup>Asst. Prof., dept of CSE,  
 BGSIT BG Nagar

*Abstract: Using Cloud Storage, users can remotely store their data 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. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free.*

## I. 1 INTRODUCTION

Cloud Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. As a disruptive technology with profound implications,

Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to 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 [2]. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, 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, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [3]–[7]. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been

or is rarely accessed, or even hide data loss incidents so as to maintain a reputation [8]–[10].

In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted [11]. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users [10], [12]. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that user does not need to perform too many operations to use the data (in addition to retrieving the data). For example, it is desirable that users do not need to worry about the need to verify the integrity of the data before or after the data retrieval. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that the cloud server only entertains verification request from a single designated party. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes [9]. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established, where

users will need ways to assess risk and gain trust in the cloud.

## II. 2 PROBLEM STATEMENTS

### 2.1 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), who has large amount of data files to be stored in the cloud; the *cloud server* (CS), which is managed by the *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* (TPA), who has expertise and capabilities that cloud user do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request.

Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. To save the computation resource as well as the online burden, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA.

We consider the existence of a semi-trusted CS as does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. However, it harms the user if the TPA could learn the outsourced data after the audit. To authorize the CS to respond to the audit delegated to TPA's, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

## III. 3 DESIGN GOALS

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees.

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 online burden to the cloud users.

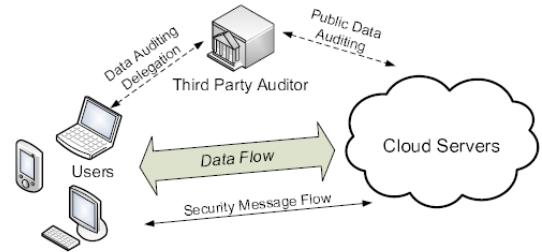


Fig. 1: The architecture of cloud data storage service

2) **Storage correctness:** to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact.

3) **Privacy-preserving:** to ensure that the TPA cannot 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 simultaneously.

5) **Lightweight:** to allow TPA to perform auditing with minimum communication and computation overhead.

## 4 THE PROPOSED SCHEMES

This section presents our public auditing scheme which provides a complete outsourcing solution of data – not only the data itself, but also its integrity checking. We start from an overview of our public auditing system and discuss two straightforward schemes and their demerits. Then we present our main scheme and show how to extend our main scheme to support batch auditing for the TPA upon delegations from multiple users. Finally, we discuss how to generalize our privacy-preserving public auditing scheme and its support of data dynamics.

## IV. RELATED WORK

Ateniese *et al.* [8] are the first to consider public audit ability in their defined “provable data possession” (PDP) model for ensuring possession of data files on untrusted storages. Their scheme utilizes the RSA-based homomorphic linear authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the auditor. Juels *et al.* [11] describe a “proof of retrievability” (PoR) model, where spot-checking and error-correcting codes are used to ensure both “possession” and “retrievability” of data files on re-remote archive service systems. However, the number of audit challenges a user can perform is fixed a priori, and public auditability is not supported in their main scheme. Although they describe a straight for-ward Merkle-tree

construction for public PoRs, this approach only works with encrypted data. Dodis *et al.* give a study on different variants of PoR with private audit ability. Shacham *et al.* [13] design an improved PoR scheme built from BLS signatures with full proofs of security in the security model defined in [11]. Similar to the construction in [8], they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach does not support privacy-preserving auditing for the same reason as [8]. Shah *et al.* [9], propose allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies both the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files and it suffers from the auditor statefulness and bounded usage, which may potentially bring in online burden to users when the keyed hashes are used up. In other related work, Ateniese *et al.* where each tuple is assigned to a polynomial-based authentication tag that ensures the verifiability of results for certain aggregation queries.

However, their scheme still did not consider the case when the CSP returned an empty result. To the best of our knowledge, there is no effective auditing scheme of outsourced database that can fully support and propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In Wang *et al.* consider a similar support for partial dynamic data storage in a distributed scenario with additional feature of data error localization.

## V. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking 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, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

In a subsequent work, Wang *et al.* [10] propose to combine BLS-based HLA with MHT to support both public auditability and full data dynamics. Almost

simultaneously, Erway *et al.* developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two protocols requires the linear combination of sampled blocks just as [8], and thus does not support privacy-preserving auditing. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, none of them meet all the requirements for privacy-preserving public auditing in cloud computing. More importantly, none of these schemes consider batch auditing, which can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations.

## VI. REFERENCES

- [1] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 [http://csrc.nist.gov/groups/SNS/cloud\\_computing/index.html](http://csrc.nist.gov/groups/SNS/cloud_computing/index.html), 2009.
- [2] 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, "Above the clouds: A Berkeley view of cloud computing," University of California, Berkeley, Tech. Rep.
- [3] M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at <http://www.techcrunch.com/2006/12/28/gmail-disaster-reports-of-mass-email-deletions/>, December 2006.
- [4] J. Kincaid, "Media Max/The Linkup Closes Its Doors," Online at <http://www.techcrunch.com/2008/07/10/Media-max-the-linkup-closes-its-doors/>, July 2008.
- [5] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at <http://status.aws.amazon.com/s3-20080720.html>, 2008.
- [6] S. Wilson, "Appengine outage," Online at <http://www.cio-weblog.com/50226711/appengine-outage.php>, June 2008.
- [7] B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at <http://voices.washingtonpost.com/securityfix/2009/01/payment-processor-breach-may-b.html>, Jan. 2009.
- [8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 598–609.
- [9] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
- [10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proc. of ESORICS'09, volume 5789 of LNCS*. Springer-Verlag, Sep. 2009, pp. 355–370.
- [11] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 584–597.
- [12] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, <http://www.Cloudsecurityalliance.org>.