PRIVACY PRESERVING PUBLIC AUDITING FOR SECURED DATA STORAGE IN CLOUD USING BLOCK AUTHENTICATION CODE

R.REVATHI [#]

PG Scholar[#],

Bharathiyar Institute Of Engineering for Women, Deviyakurichi, Salem(DT) India

***Abstract*— Using cloud storage, users can remotely store their data, on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. Users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing for users with constrained computing resources. Users should be able to use the cloud storage as if it is local, without worrying about the need to verify its integrity. To enable public auditability 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. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. To propose a secure cloud storage system supporting privacy-preserving public auditing. and further extend result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient**

***Keywords*— Batch verification, Data storage, privacy preserving, public auditability, zero knowledge**

I. 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 resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [2]. As a disruptive technology with profound implications, cloud computing is transforming the very nature of businesses use information technology. One fundamental aspect of this paradigm shifting is that data are 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 location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc., [3]. While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward 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 [4]. Examples of outages and security breaches of noteworthy cloud services appear from time to time [5], [6], [7]. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation [8], [9], [10]. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. 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 [12], [8]. Moreover, the overhead of using cloud storage should be minimized as much as possible, that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. There may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud 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. 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 [10].To 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. Recently, the notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different system and security models [9],[13], [11], [8]. Public auditability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. Most of these schemes [9], [13], [8] do not consider the privacy protection of users' data against external auditors. Indeed, they may potentially reveal user's data to auditors. This severe drawback (Placeholder1)greatly affects the security of these protocols in cloud computing. From the perspective of protecting data privacy, the users, who own the data and rely on TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage toward their data security [14], [15]. Moreover, there are legal regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA) [16],further demanding the outsourced data not to be leaked to external parties [10]. Simply exploiting data encryption before

outsourcing [15], [11] could be one way to mitigate this privacy concern of data auditing, but it could also be an overkill when employed in the case of unencrypted/public cloud data (e.g., outsourced libraries and scientific data sets), due to the unnecessary processing burden for cloud users. Besides, encryption does not completely solve the problem of protecting data privacy against third-party auditing but reduces it to the complex key management domain. Unauthorized data leakage still remains possible due to the potential exposure of decryption keys. The individual auditing of the growing tasks can be tedious and cumbersome, a natural demand is then to enable the TPA to efficiently perform multiple auditing tasks in a batch manner, i.e., simultaneously. To address these problems, utilizes the technique of public key-based homomorphic linear authenticator (or HLA) [9], [13], [8], enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, our protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit for the batch auditing.

II. 2 PROBLEM STATEMENT

A. *The System and Threat Model*

The cloud user has large amount of data files to be stored in the cloud; the cloud server, is managed by the cloud service provider to provide data storage service and has significant storage

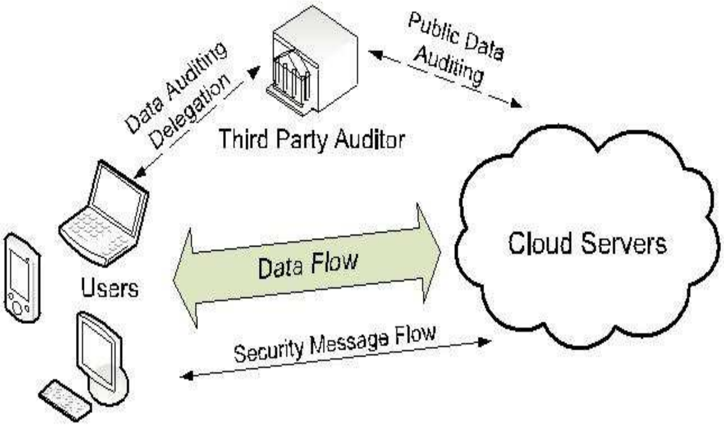

Fig. 1**. The architecture of cloud data storage service.**

the third-party auditor, has expertise and capabilities that cloud users 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 also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA. The data integrity threats toward users' data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For the benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. Entities will deviate from the prescribed protocol execution. To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate.

B. *Design Goals*

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, 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.
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.

III. RELATED WORK

"Provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for auditing outsourced data and suggest randomly sampling a few blocks of the file. Among their two proposed schemes, the one with public auditability exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the external 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 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 straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Later, Bowers et al. [18] propose an improved framework for POR protocols that generalizes Juels' work. Dodis et al. [29] also give a study on different variants of PoR with private auditability. Shacham and Waters [13] design an improved PoR scheme built from BLS signatures [19] with proofs of security in the security model defined in [11]. Similar to the construction in [9], 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 is not privacy preserving due to the same reason as [9]. Shah et al. [15], [10] propose introducing 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 the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up. Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. [21] is the first to propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [22], Wang et al. consider a similar support for partially dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [8] propose to combine BLS-based HLA with MHT to support fully data dynamics. Concurrently, Erway et al. [23] develop a skip list based scheme to also enable provable data possession with full dynamics support. However, the verification in both protocols requires the linear combination of sampled blocks as an input, like the designs in [9], [13], and thus does not support privacy-preserving auditing. In other related work, Sebe et al. [30] thoroughly study a set of requirements which ought to be satisfied for a remote data possession checking protocol to be of practical use. Their proposed protocol supports unlimited times of file integrity verifications and allows preset trade off between the protocol running time and the local storage burden at the user. Schwarz and Miller [31] propose the first study of checking the integrity of the remotely stored data a cross multiple distributed servers. Their approach is based on erasure-correcting code and efficient algebraic signatures, which also have the similar aggregation property as the homomorphic authenticator utilized in our approach. Curtmola et al. [32] aim to ensure data possession of multiple replicas across the distributed storage system. They extend the PDP scheme in [9] to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained. In [33], Bowers et al. utilize a two-layer erasure-correcting code structure on the remotely archived data and extend their

POR model [18] to distributed scenario with high-data availability assurance. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, almost none of them necessarily meet all the requirements for privacy-preserving public auditing of storage. Moreover, none of these schemes consider batch auditing, while our scheme can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations. Portions of the work presented in this paper have previously appeared as an extended abstract in [1]. We have revised the paper a lot and improved many technical details as compared to [1]. The primary improvements areas follows: First, we provide a new privacy-preserving public auditing protocol with enhanced security strength in For completeness, we also include an additional (but slightly less efficient) protocol design for provably secure zero-knowledge leakage public auditing scheme Second, based on the enhanced main auditing scheme, we provide a new provably secure batch auditing protocol. All the experiments in our performance evaluation for the newly designed protocol are completely redone. Third, we extend our main scheme to support data dynamics, and provide discussions on how to generalize our privacy-preserving public auditing scheme in , which are lacking in [1]. Finally, we provide formal analysis of privacy-preserving guarantee and storage correctness, while only heuristic arguments are sketched in [1].

IV. CONCLUSION

To 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 privacy-preserving public auditing protocol into a multiuser 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. Preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of our design on both the cloud and the auditor side. We leave the full-fledged implementation of the mechanism on commercial public cloud as an important future extension, which is expected to robustly cope with very large scale data and thus encourage users to adopt cloud storage services more confidently.

V. REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[2] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing,"http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, June 2009.

[3] 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," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

[4] Cloud Security Alliance, "Top Threats to Cloud Computing," http://www.cloudsecurityalliance.org, 2010.

[5] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof-mass-email-deletions/, 2006.

[6] J.Kincaid, "MediaMax/TheLinkup Closes Its Doors," http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits-doors/, July 2008.

[7] Amazon.com, "Amazon s3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, July 2008.

[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[9] 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. 598-609, 2007.

[10] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[11] A. Juels and J. Burton, S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, Oct. 2007.

[12] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing," http://www.cloudsecurityalliance. org, 2009.

[13] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107,Dec. 2008.