IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 3, May 2014.
www.ijiset.com
ISSN 2348 – 7968

Third Party Auditor for Secure Cloud storage

Tinku Abey Koshy, S Prema(Ph.D)
Department of Computer Science, Mahendra Institute of Technology,
Mallasamudaram, Tiruchengode ,Tamil nadu, India

Abstract: **The cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor to check the integrity of outsourced data and be worry-free. Auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our 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. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.**

Keywords—Data Storage, Data Intgrity, Privacy Preserving, cryptography, batch auditing

I. INTRODUCTION

The cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion.

Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse.

An important part of gene regulation that is mediated by specific proteins are called transcription factors that influence the transcription of a particular gene by binding to specific sites on DNA sequences, called transcription factor binding sites. The binding sites often emerge as a combination of two or more Public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud data without learning the data content. To the best of our knowledge, our scheme is the first to support scalable and efficient privacy preserving public storage auditing in Cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a Privacy-preserving manner.

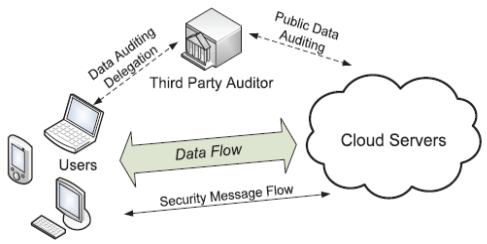

Fig. 1. Architecture of Cloud Data Storages.

ISSN 2348 – 7968

We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

II. PROPOSED SYSTEM

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. 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.

A.*THE 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 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. 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.

B. *Privacy-Preserving Public Auditing Scheme*

To achieve privacy-preserving public auditing, we propose to 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. On the other hand, the correctness validation of the block authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key based HLA, to equip the auditing protocol with public auditability. Specifically, we use the HLA proposed in, which is based on the short signature.

Scheme Details. Let G1, G2 and GT be multiplicative cyclic groups of prime order p, and $e : G1 \times G2 \rightarrow GT$ be a bilinear map as introduced in preliminaries. Let g be a generator of G2. H(・) is a secure map-to pointhash function: $\{0, 1\}* \rightarrow G1$, which maps strings uniformly to G1. Another hash function h(・) : GT →Zp maps group element of GT uniformly to Zp. 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 ← Zp, a random element u ← G1, and computes v ← gx. The secret parameter is sk = (x, ssk) and the public parameters are pk =(spk, v, g, u, e(u, v)). Given a data file F = (m1, . . . ,mn), the user runs SigGen to compute authenticator _i for each block mi: _i ← (H(Wi) ・ umi)x ∈ G1. Here Wi = name||I and name is chosen by the user uniformly at random from Zp as the identifier of file F. Denote the set of authenticators by _ ={_i}1≤i≤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||SSigssk(name) as the file tag for F, where SSigssk(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 (_, 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 we describe in the Setup phase, the TPA verifies the signature SSigssk (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 = {s1, . . . , sc} of set [1, n]. For each element i ∈ I, the TPA also chooses a random value _i (of bit length that can be shorter than |p|, as explained in the message "chal" specifies the positions of the blocks that are required to be checked. The TPA sends chal = {(i, _i)}i∈I to the server.

C. *Support for Batch Auditing*

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case, and achieves the aggregation of K verification equations (for K auditing tasks) into a single one, as shown in Equation As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

III. DISCUSSION

Discussion gives an asymptotic efficiency analysis on the batch auditing, by considering only the total number of pairing operations. However, on the practical side, there are additional less expensive operations required for batching, such as modular exponentiations and multiplications. Meanwhile, the different sampling strategies, i.e., different number of sampled blocks c, is also a variable factor that affects the batching efficiency. Thus, whether the benefits of removing pairings significantly outweighs these additional operations is remained to be verified. To get a complete view of batching efficiency, we conduct a timed batch auditing test, where the number of auditing tasks is increased from 1 to approximately 200 with intervals of 8. The performance of the corresponding non-batched (individual) auditing is provided as a baseline for the measurement. Following the same experimental settings c = 300 and c = 460, the average per task auditing time, which is computed by dividing total auditing time by the number of tasks.

IV.CONCLUSION

In this paper we have worked to facilitate the client in getting a proof of integrity of the data which he wishes to store in the cloud storage servers with bare minimum costs and efforts. Our scheme was developed to reduce the computational and storage overhead of the client as well as to minimize the computational overhead of the cloud storage server. We also minimized the size of the proof of data integrity so as to reduce the network bandwidth consumption. Many of the schemes proposed earlier require the archive to perform tasks that need a lot of computational power to generate the proof of data integrity. But in our scheme the archive just need to fetch and send few bits of data to the client

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM.

[2] Sravan Kumar R, Ashutosh Saxena, "Data Integrity Proofs in Cloud Storage".

[3] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Proc. of Crypto'96, volume 1109 of LNCS. Springer-Verlag, 1996..

[4] K.HariPriya #1 (Member – IEEE), P.Krishnamoorthy,An Efficient Cloud Storage with Secure

[5] T.Brindha, R.S.Shaji, G.P.Rajesh, A Survey on the Architectures of Data Security
in Cloud Storage Infrastructure..

[6] Susan B. Davidson Univers+ity of Pennsylvania,Sanjeev Khanna University
of Pennsylvania,Enabling Privacy in Provenance-Aware Workflow Systems.

[7] Mark Lillibridge Sameh Elnikety Andrew Birrell Mike Burrows Michael Isard
HP Systems Research Center,"A Cooperative Internet Backup Scheme"

[8] 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

ISSN 2348 – 7968

(CCS ’07), pp. 598-609, 2007.

[9] Ayad F.Barsoum and M.Anwar Hasan,"Provable Possession and Replication

[10] 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.

[11] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient
Provable Data Possession,” Proc. Int’lConf. Security and Privacy in Comm. Networks
(SecureComm ’08),pp. 1-10, 2008.

[12] Kevin D. Bowers, Ari Juels, and Alina Oprea RSA Laboratories, Bedford, MA,
Proofs of Retrievability: Theory and Implementation.

[13] Kevin D. Bowers RSA Laboratories,Ari Juels,RSA Laboratories Alina Oprea
RSA Laboratories,"HAIL: A High-Availability and Integrity Layer for Cloud Storage".

[14] Ensuring Data Storage Security in Cloud Computing,"Ensuring Data Storage Security in Cloud Computing".