Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secure Cloud Storage System Incorporating Privacy-Preserving Third Party Audit

Suchita Ghatage
RMD Sinhgad COE,
Computer Department, Pune University, India

D. N. Rewadkar
Associate Prof. RMD Sinhgad COE,
Computer Department, Pune University, India

Abstract— Cloud storage is actually made up of many distributed resources, but still acts as one. It is highly fault tolerant through redundancy and distribution of data also it is highly durable through the creation of versioned copies. Users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. Data integrity is a prime concern in cloud computing as users no longer have physical possession of the outsourced data. This article studies the problem of ensuring the integrity and security of data storage in Cloud Computing. Users should be able to use the cloud storage as if it is local, without the burden of the need to verify its integrity. So correctness of data and security is a prime concern. So, enabling public auditability for cloud storage is of critical importance. To ensure the correctness of data, an external auditor called as a third party auditor (TPA) should be securely introduced, on behalf of the cloud user, to verify the integrity of the data stored in the cloud. Users' data privacy should not be compromised by bringing in the auditing process, and introduce no additional online burden to them. In this paper, a secure cloud storage system is proposed which incorporates privacy-preservingThird Party Audit. The results can be further extended to enable the TPA to perform audits for multiple users simultaneously and efficiently in a batch manner.

Keywords—Data integrity, Cloud computing, Third Party Auditing, Homomorphic encryption.

I. INTRODUCTION

Cloud computing is the next natural step in the evolution of on-demand information technology services and products. Cloud computing is a paradigm that focuses on sharing data and computations over a scalable network of nodes which include end user computers, data centers, and Web Services. Cloud storage provides users with immediate access to a broad range of resources and applications hosted in the infrastructure of another organization via a web service interface. Companies need to pay only for the storage they actually use. Users do not need to check the integrity of data stored on cloud servers before using it, instead he can treat the storage as if it's local. Cloud computing has many advantages like less maintenance on user's side as hardware, applications are managed by cloud service provider, user needs to pay only for those resources which are used, the resources can be scaled according to user's requirements. In spite of these appealing advantages, cloud computing also brings new and challenging security threats toward user's outsourced data. Data is outsourced to separate administrative entities known as cloud service providers (CSP) which means user's loss of control over data integrity. The risk of incorrect data in the cloud increases due to the following reasons. Infrastructures under the cloud are powerful and reliable than personal computing devices still they are facing the broad range of both internal and external threats for data integrity [2]. Second, CSP may discard the data which have not been used or accessed rarely, or to just maintain reputation may hide data loss incidents and give the wrong status of outsourced data to the user [3, 4, 5]. Users do not have physical possession of data so cryptographic techniques cannot be used directly. Also downloading all the data only for integrity check is not a practical solution as I/O and transmission costs are expensive [3,7]. As the outsourced data is large in size and user has limited and constrained resources, he may not want to do the complex and expensive auditing task. Like one user there will be many users who would like to check their data for integrity, so it would be easier to manage verification request from a single designated party instead of requests from multiple users in the cloud.

To make user's task simpler, enabling public auditing service for cloud data storage is important. Users have constrained and limited resources so they may delegate the auditing task to third-party auditor (TPA) to audit the outsourced data when needed. Unlike users, the TPA has expertise and capabilities, can periodically check the integrity of all the data stored in the cloud. This will save user's resources, computation time and will ensure their storage correctness in the cloud.Cloud service providers can also get benefits from TPA to improve their cloud-based service platform, and even serve for independent arbitration purposes [5]. Therefore, to enable a privacy-preserving third party auditing protocol, independent of data encryption, is the problem we are going to tackle in this paper.

II. THE SYSTEM MODEL

Cloud data storage service involves three different entities as shown in Fig. 1[1]. The **cloud user**: needs to store large number of data files on cloud server; **the cloud server**: it is managed by cloud service provider, who is responsible for providing data storage service and has significant storage space and computation resources; **the third-party auditor:** has the expertise and capabilities which user does not have and on users request has access to cloud storage service.

The data is no longer in local possession of the user, so it is very important to ensure that the data is correctly stored and maintained on cloud servers. The user may delegate the task of checking data integrity to a third party auditor (TPA), this will save users resources and computation time.

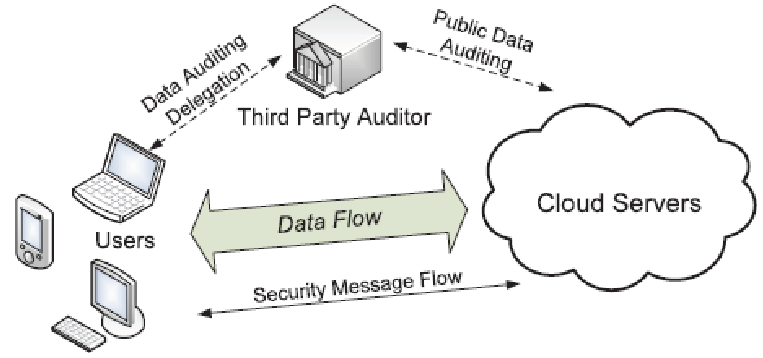

Fig. 1. The architecture of cloud storage

The threats to user's data can be categorized into internal and external attacks at CS. These may include: bugs in the software and/or in the network path, failures of hardware, malicious or accidental management errors, hackers etc. CS cannot be trusted as he may hide the data loss or data corruption incidents from user to maintain a reputation. Thus using privacy-preserving public auditing system can be a cost effective way to gain trust in cloud [1]. We assume that the TPA is in the business of auditing, capable of auditing, reliable and independent.

III. EXISTING SYSTEM

Before starting with the proposed system we highlight the demerits of using two classes of schemes. The first one is a MAC-based solution and the second one is a system based on homomorphic linear authenticators.

MAC-based solution:Users can upload the data blocks with their MACs to the cloud servers and can send the corresponding secret key to the TPA. To check the correctness of data, the TPA can randomly retrieve blocks with the MACs and verify the correctness by using the secret key. There are some drawbacks in this approach like 1) As TPA retrieves data blocks it violates the condition of privacy preserving. 2) This approach has high communication and computation complexities.

To avoid the problem of giving access of data blocks to TPA the verification may just consist of equality checking. Cloud user can choose random message authentication code keys, precompiles MACs for the whole data file, and share the verification metadata that is the keys and the MACs with the TPA. By sharing the secret key with the cloud server and asking for fresh keyed MAC for comparison each time TPA can achieve privacy-preserving auditing. But it has certain drawbacks:

1. The number of secret keys must be fixed a priori which puts limits on the number of times a particular data file can be audited. The user has to retrieve the full data to compute and publish new MACs to TPA again and again when all possible secret keys are exhausted.
2. Considering the large number of audit requests from multiple users, TPA has to maintain and update states between audits, and has to keep track of the MAC keys which are revealed to cloud server.
3. It cannot deal with the dynamic data efficiently.

HLA-based solution: HLA based solution can be used to do a public audit effectively, without having to retrieve the data blocks themselves and also some unforgeable verification metadata that authenticates the integrity of a data block. HLA allows efficient data auditing and consumes only constant bandwidth. However HLA technique may potentially reveal user data information to TPA and may violate the privacy preserving guarantee as TPA can simply solve a system of linear equations used in the HLA technique [1].

IV. PROPOSED SYSTEM

To enable privacy-preserving public auditing for cloud data storage, the design should achieve the following security and performance guarantee:

1) To allow TPA to verify the correctness of the cloud data on request of user without retrieving a copy of the data or bringing in additional on-line burden to the cloud users.
2) To ensure that without maintaining user's data integrity, no cloud server can pass the auditing process performed by the TPA by some or the other ways.
3) To ensure that the user's data content can not be derived from the information collected during the auditing process by the TPA.
4) To enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from a possibly large number of different users simultaneously.

5) To allow TPA to perform auditing with minimum communication and computation overhead.

Though HLA with random masking solves the problem of privacy-preserving, it increases the burden of maintenance and calculation of masking information on user as well as on TPA. To overcome this we propose a system in which TPA chooses some Random MACs and then generate challenge to Server for those MACs only. This improves computational time. We can use Elgamal Homomorphic encryption for Random MACs. The flow of events is shown in Fig. 2 and are explained as follows:

- User can store files on cloud server by uploading the files on the server, and when required can et files by downloading the files from the server.
- Now to check the integrity of the data file Third Party Auditor (TPA) is introduced. On user's request the TPA will perform the auditing task and get the necessary metadata from the user.
- TPA will generate a challenge for the cloud server and verify the response given by the cloud server.Here important thing to take into consideration is that TPA does not have knowledge about the content of the data file.

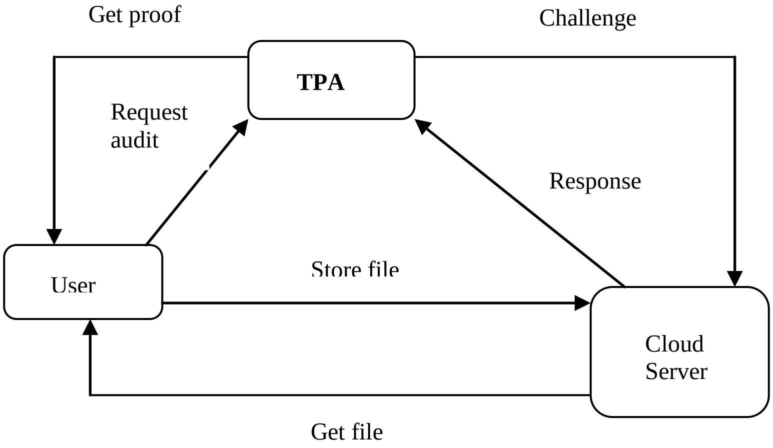

Fig. 2 Event flow

There are four algorithms which can be used in a public auditing scheme (KeyGeneration, SigGeneration, GenrateProof, and VeriProof).

- KeyGeneration: the key generation algorithm that is run by the user to setup the scheme.
- SigGeneration: used by the user to generate verification metadata, this may consist of MAC, signatures or other information used for auditing.
- GenrateProof: run by the cloud server to generate a proof of data storage correctness.
- VeriProof: run by the TPA to audit the proof from the cloud server.

V. RELATED WORK

Public auditability was first proposed by Athens et al. [4] in their "provable data possession" (PDP) model which verifies whether the data files are stored on untrusted server. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces the computation on the server, the file block accesses, and the client-server communication. However, their public auditability scheme reveals the linear combination of sampled blocks to the external auditor, their protocol is not provably privacy preserving, and may leak user data information to the external auditor. A "proof of retrievability"(PoR) model described by Juels et al. [6],enables user to retrieve a target file F which proves that an archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. A POR is specially designed to handle a large file (or bitstring) F. However, public auditability is not supported in their main scheme and the number of audit challenges a user can perform is fixed a priori. Shacham and Waters [8] design an improved PoR scheme, similar to the construction in [4], they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures [10]. Again, their approach is not privacy preserving due to the same reason as [4]. Shah et al. [9], [5] propose a scheme where the data is first encrypted then a number of precomputed symmetric-keyed hashes over the encrypted data are sent to the third party auditor. The auditor verifies the integrity of the data file and the server's possession of a previously committed decryption key. The scheme requires the auditor to maintain state, it works only for encrypted files, and suffers from bounded usage, which eventually brings in online burden to users when the keyed hashes are used up.

VI. CONCLUSIONS

In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We propose the use of homomorphic random authentication which will not allow TPA to learn any knowledge about the data content stored on the cloud server during the auditing process. We can extend the proposed system into batch auditing, where TPA can perform multiple auditing tasks in a batch manner. The full-fledged implementation of the mechanism on the commercial public cloud is 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.

REFERENCES

[1] Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE, “Privacy-Preserving Public Auditing for Secure Cloud Storage” *IEEE Transactions on Computers, vol. 62, no. 2, February 2013.*

[2] Cloud Security Alliance, “Top Threats to Cloud Computing,” http://www.cloudsecurityalliance.org, 2010.

[3] 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.*

[4] 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.*

[5] M.A. Shah, R. Swaminathan, and M. Baker, “Privacy-Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008/186, 2

[6] 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. 2007.008.*

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

[8] 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.

[9] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, “Auditing to Keep Online Storage Services Honest,” Proc. 11th USENIXWorkshop Hot Topics in Operating Systems (HotOS ’07), pp. 1-6, 2007.

[10] D. Boneh, B. Lynn, and H. Shacham, “Short Signatures from the Weil Pairing,” J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.