IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Solitude Preservative Public Auditing with Access and Permission

S.Sharmiladevi[1], Prof.G.Ilanchezhiapandian[2]

[1]PG Student, Department of Computer science and Engineering, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore - 632102, Tamil Nadu, India

[2]Dean(academic)HOD, Department of Computer science and Engineering,Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore-632 102, Tamil Nadu, India

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. In this paper, i propose a privacy-preserving public auditing system for data storage security in cloud computing. I 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.

Keywords-*Data storage, privacy preserving, public auditability, cloud computing, delegation, batch verification.*

1. Introduction

CLOUD computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of exceptional advantages in the IT history: on-demand self-service, ever-present network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and conveyance of risk. One fundamental aspect of this example shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both persons and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings pleasing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenses on hardware, software, and personnel maintenances, etc., 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 relinquish 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.

Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, cloud service provider might reclaim storage for economic reasons by discarding data that have not been or are not often accessed, or even hide data loss incidents to maintain a reputation. The notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different system and security models. Public auditability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data.

To address these problems, my work utilizes the technique of public key-based homomorphic linear authenticator which 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, my 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 my design for the batch auditing. Specifically, my contribution can be summarized as the following three aspects:

1. I motivate the public auditing system of data storage security in cloud computing and provide a privacy preserving auditing protocol. I scheme enables an external auditor to audit user's cloud data without learning the data content.
2. To the best of i knowledge, our scheme is the first to support scalable and efficient privacy-preserving public storage auditing in cloud. Specifically, my scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.

3. I prove the security and justify the performance of my proposed schemes through concrete experiments and comparisons with the state of the art.

2. System Description and Methodology

We consider a cloud data storage service involving three different entities, the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider 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, 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. 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.

We assume 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 their own 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. We assume the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit.

To enable the public auditability for cloud data storage security I propose a solution for privacy of users, so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met:

1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user;

2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. In this paper, i utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements.

2.1 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:

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.
Storage correctness	to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact
Privacy preserving	to ensure that the TPA cannot derive users' data content from the information collected during the auditing process.
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.
Lightweight	to allow TPA to perform auditing with minimum communication and computation overhead

3. Methodology

3.1 Basic Scheme

3.1.1 MAC-based solution.

There are two possible ways to make use of MAC to authenticate the data. A trivial way is just uploading the data blocks with their MACs to the TPA. Later, the TPA can randomly retrieve blocks with their MACs and check the correctness via sk. Apart from the high (linear in the sampled data size) Communication and computation complexities, the TPA requires the knowledge of the data blocks for verification. The idea is as follows: Before data outsourcing, the cloud user chooses s arbitrary message authentication code keys precomputes (deterministic) MACs for the whole data file F, and publishes these verification metadata (the keys and the MACs) to TPA. The TPA can reveal a secret key sk to the cloud server and ask for a fresh keyed MAC for evaluation in each audit.There are few drawbacks in this MAC based solution there are.,

The number of times a particular data file can be audited is limited by the number of secret keys that must be fixed a priori. Once all possible secret keys are worn out the user then has to retrieve data in full to recompute and republish new MACs to TPA; The TPA also has to maintain and update state between audits, i.e., keep track on the revealed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone

3.1.2 HLA based solution

To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique can be used. HLAs, like MACs, are also some unforgeable verification metadata that authenticate the integrity of a data block. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks. The user still authenticates each element of by a set of HLAs . The TPA verifies the cloud storage by sending a random set of challenge This is because the linear combination of blocksP may potentially reveal user data information to TPA, and violates the privacy-preserving guarantee. Specifically, by challenging the same set of c block m1;m2; . . .;mc using c different sets of random coefficients TPA can accumulate c different linear combinations With TPA can derive the user's data m1;m2; . . .;mc by simply solving a system of linear equations.

3.2 Privacy Preserving Public Auditing Scheme

To achieve privacy-preserving public auditing, i 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 by 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. Running a public auditing system consists of two phases, Setup and Audit.

Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F and the verification metadata at the cloud server, and deletes its local copy. As part of preprocessing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof.

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof).

KeyGen: key generation algorithm that is run by the user to setup the scheme

SigGen: used by the user to generate verification metadata, which may consist of MAC, signatures or other information used for auditing

GenProof: run by the cloud server to generate a proof of data storage correctness

VerifyProof: run by the TPA to audit the proof from the cloud server.

3.3 Support for batch auditing

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 (3). As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

Setup phase: Basically, the users just perform Setup independently. Suppose there are K users in the system, and each user k has a data file Fk ¼ ðmk;1; . . .;mk;nÞ to be outsourced to the cloud server, where k 2 f1; . . .;Kg. For simplicity, we assume each file Fk has the same number of n blocks. For a particular user k, denote his/her secret key as ðxk; sskkÞ, and the corresponding public parameter as ðspkk; vk; g; uk; eðuk; vkÞÞ where vk ¼ gxk . Similar to the single user case, each user k has already randomly chosen a different (with overwhelming probability) name namek 2 ZZp for his/her file Fk, and has correctly generated the corresponding file tag tk ¼ namekkSSigsskk ðnamekÞ.

Audit phase: TPA first retrieves and verifies file tag tk for each user k for later auditing. If the verification fails, TPA quits by emitting FALSE. Otherwise, TPA recovers name k and sends the audit challenge chal ¼ fði; _iÞgi2I to the server for auditing data files of all K users.

3.3.1 Efficiency improvement

Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side. This is because aggregating K verification equations into one helps reduce the number of relatively expensive pairing operations from 2K, as required in the individual auditing, to K þ 1, which saves a considerable amount of auditing time.

Sequence Diagram for Privacy Preserving Public Auditing

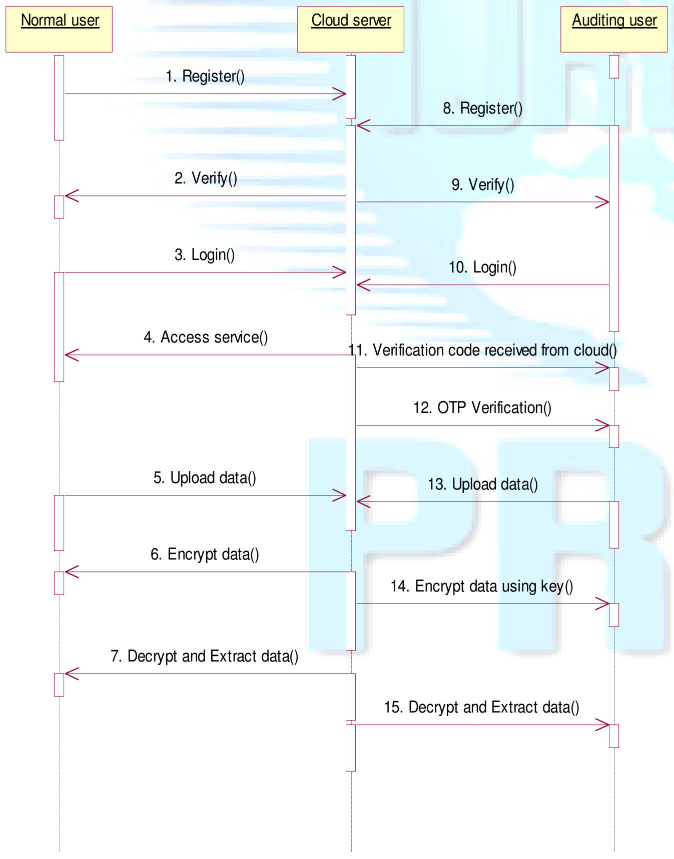

Figure 1: Sequence diagram for public auditing

4. System Design

4.1 Architectural Framework

4.1.1 User

The Cloud User who have a large amount of data to be stored in multiple clouds and have the permissions to access and manipulate stored data. the User's Data is converted into data blocks . the data blocks is uploaded to the cloud. The TPA view the data blocks and Uploaded in multi cloud. The user can update the uploaded data. If the user wants to download their files, the data's in multi cloud is integrated and downloaded

4.1.2 Third Party Auditor

Trusted Third Party (TTP) who is trusted to store verification parameters and offer public query services for these parameters. In our system the Trusted Third Party, view the user data blocks and uploaded to the distributed cloud. In distributed cloud environment each cloud has user data blocks. If any modification tried by cloud owner an alert is send to the Trusted Third Party.

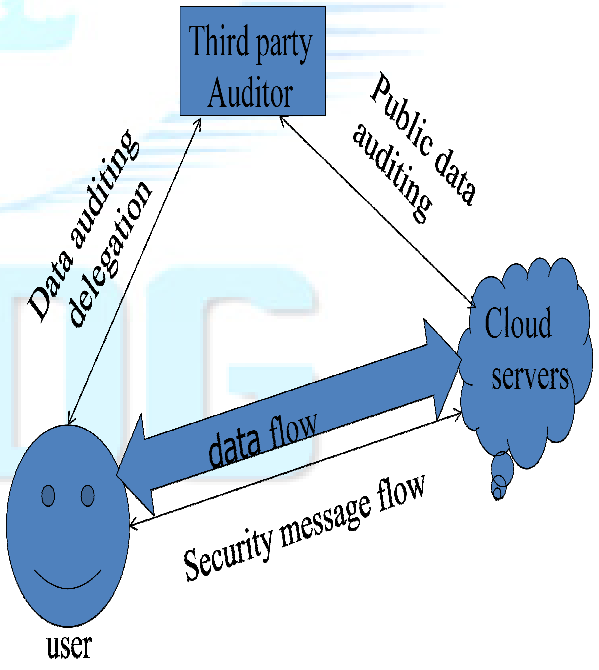

Figure 2: Architectural diagram of cloud data storage.

4.1.3 Batch Auditing

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 (3). As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

5. Conclusion

In this paper, i propose a privacy-preserving public auditing system for data storage security in Cloud Computing, where TPA can perform the storage auditing without demanding the local copy of data. I utilize the homomorphic authenticator and random mask technique to guarantee that 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, i further extend our privacy preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Extensive security and performance analysis shows that the proposed schemes are provably secure and highly efficient. I believe all these advantages of the proposed schemes will shed light on economies of scale for Cloud Computing.

References

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security 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] J. Kincaid, "MediaMax/TheLinkup Closes Its doors,"http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits- doors/, July 2008.

[6] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine,
vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[7] A.L. Ferrara, M. Green, S. Hohenberger, and M. Pedersen, "Practical Short Signature Batch Verification," Proc. Cryptographers' Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA), pp. 309-324, 2009.

[8] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.

[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, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems HotOS '07), pp. 1-6, 2007.

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. computer and Comm. Security (CCS '09), pp. 213-222, 2009.