International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-4, July 2015
ISSN: 2395-3470
www.ijseas.com

Effective Privacy-Preserving Public Auditing for Data Sharing in Cloud

Mr. Kedar Jayesh Rasal [1], Dr. Shyamrao V. Gumaste [2], Prof.Sandip A. Kahate[3]

Computer Engineering, Pune University,
SPCOE, Otur, Pune, Maharashtra, India

Professor And Head, Department of Computer Engineering,
SPCOE, Otur, Pune, Maharashtra, India.

Abstract— Cloud computing represents today's most surprising computing paradigm shift in information technology. Without taking burden of local data storage and maintenance by 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. Most primary obstacles to its wide adoption are security and privacy. Shorter physical possession of the outsourced data of users makes the data integrity protection in Cloud. If the cloud storage is local then users should be able to just use it, without worrying about the need to verify its integrity. Enabling 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 introduce an effective TPA with security the auditing process should brings it with no new vulnerabilities towards user data privacy and also introduces no additional online burden to user. The secure cloud storage system supporting privacy-preserving public auditing is discussed in this paper.

Index Terms— Cloud computing, data storage, privacy-preserving, public auditability, cryptographic protocols.

I. INTRODUCTION

Cloud computing and storage allow users are to access and share resources offered by cloud service providers at a lower marginal cost. It is routine for users to have cloud storage services to use data with others in a group as data sharing is unique feature in many cloud storage offerings. The integrity of data in cloud storage, subject to doubt and challenge, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud and then check data integrity by checking the correctness of the entire data.

Cloud computing makes many advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced data. Cloud service providers (CSP) are separate administrative entities where the data outsourcing is actually relinquishing user's ultimate control over the fate of their data. So in result, the correctness of the data in the cloud is being put at risk due to the following reasons. Even though the infrastructures under the cloud are more powerful and reliable than personal computing devices, still they are facing the broad range of both internal and external threats for data integrity. The outages and security breaches of noteworthy cloud services are the best examples which appear from time to time .Second, there exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. 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 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. If this problem is not properly addressed it may impede the success of cloud architecture [2], [4],[7],[12]. For managing easily, it is desirable that cloud only entertains verification request from a single designated party.

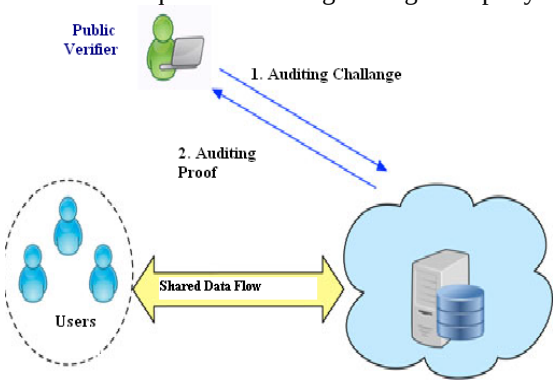

Fig. 1. System model which includes the cloud server, a group of users and a public verifier.

If the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and response protocol between a public verifier and the
cloud server [9].

The TPA, who has expertise and capabilities that users do not, it 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 [11]. Most of the existing schemes do not consider the privacy protection of user's data against external auditors [15], [16].

To solve the privacy issue on shared data, this paper discusses novel privacy retaining public auditing mechanism so that to verify the integrity of shared data by a public verifier without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier [1].

II. LITERATURE SURVEY

In existing systems Ring signatures were used to compute verification metadata needed to audit the correctness of shared data. The identity of owner of each block in shared data was kept private from public verifiers so in some special cases it couldn't identify the author. Only data blocks were authenticated not their version. There were Static data means they were unable to add data dynamically.

Currently, many available mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing [5]. In the available mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking [13]. A public verifier data users such as researcher would like to utilize the owner's data via the cloud or a third-party auditor (TPA) which can provide expert integrity checking services [3],[12]. Wang et al. designed an advanced auditing mechanism [5] (named as WWRL in this paper), so that during public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers[1].

Even though the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. An open challenging task in cloud computing is how to achieve a secure and efficient design to integrate seamlessly for data storage service. Sharing data among multiple users is one of the most engaging features that motivate cloud storage. So it becomes necessary to ensure the integrity of shared data in the cloud is correct.

III.PROBLEM DEFINITION AND DESIGNING GOALS

A. Problem Definition

Now a day there has been increase in use of Cloud data services. These are used everywhere for many purposes. The integrity of cloud data is main concerned so this needs to check integrity of data. This checks the integrity of data to check correctness of data. Having the large size of the cloud data and the user's capability of resources, the tasks of analyzing the data correctness in a cloud environment can become difficult and expensive for the cloud users. The overhead of using cloud data storage should be kept minimum as much as possible, so that a user does not need to perform too many operations to use the data (in addition to retrieving the cloud data). In specific, users may not want to go through the hazards; it needs the verification of the data integrity. Apart from this, there may be more than one user accesses the same cloud data, say in an enterprise setting. There are many systems to check the correctness of data but no identity privacy is provided in any mechanism. In Public auditing there used to be reveal of confidential information. So privacy-retaining is important mechanism.

B. Design Goals

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 storing users' data intact in reality.
3) Privacy preserving: to ensure that users' data content from the information collected during the auditing process cannot derived by TPA .
4) Batch auditing: to enable TPA with secure and efficient auditing capability to work with multiple auditing delegations from possibly different, large number of users simultaneously.
5) Lightweight: to permit TPA to perform auditing with minimum communication and computation overhead.

C. Public Auditing Scheme Algorithms

Public auditing scheme provides a complete outsourcing solution of data with data integrity check. A public auditing scheme consists of four algorithms(KeyGen, SigGen, GenProof, VerifyProof).
1)KeyGen – It is a key generation algorithm that is run by the user to setup the scheme.
2)SigGen- It is used by the user to generate verification metadata, which may consist of digital signatures.
3)GenProof- It is run by the cloud server to generate a proof of data storage correctness.
4)VerifyProof – It is run by the TPA to audit the roof. Running a public auditing system consists of two phases, Setup and Audit:
i)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. Then user stores the data file F and the verification metadata at the cloud server, and deletes its local copy. In the preprocessing part the user may alter the data file F by expanding it or including additional metadata to be stored at server.
ii)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. By executing GenProof using F and its verification metadata as inputs the cloud server will derive a response message. Then TPA verifies the response via VerifyProof.

This framework assumes that the TPA is stateless, i.e. there is need to maintain and update state between audits by TPA, which is a desirable property especially in the public auditing system [13]. Essentially by splitting the verification metadata into two parts which are stored by the TPA and the cloud server, respectively it is easy to extend the framework above to capture a stateful auditing system,. This design does not assume any additional prop error resilience, he can first redundantly encode the data file and then it uses system with the data that has error correcting codes integrated.

IV. BASIC SCHEMES

To warm-up the result the scheme is divided in to two classes. The first is a MAC-based solution which suffers from undesirable systematic demerits—bounded usage and stateful verification, which may pose additional online burden to users, in a public auditing setting. This also shows that the auditing problem is still not easy to solve even if TPA introduced. The second one is a system based on homomorphic linear authenticators, which covers much recent proofs of storage systems. Why all existing HLA-based systems are not privacy preserving. The analysis of these basic schemes leads to main result, which overcomes all these drawbacks. This paper discusses the main scheme which is based on a specific HLA scheme.

A. MAC-based solution

To authenticate the data there are two possible ways to make use of MAC. A common way is just uploading the data blocks with their MACs to the server, and sends the corresponding secret key sk to the TPA. After , the TPA can randomly retrieve blocks with their MACs and check the correctness by sk. The TPA requires the knowledge of the data blocks for verification apart from the high communication and computation complexities,. To circumvent the requirement of the data in TPA verification, any one may restrict the verification to just consist of equality checking. The idea is as follows:

However, it suffers from the following severe drawbacks:
1) 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. All possible secret keys are exhausted by the user then has to retrieve data in full to recompute and republish new MACs to TPA;
2) The TPA also has to keep track on the revealed MAC keys to maintain and update state between audits. By considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone;
3) It cannot efficiently deal with dynamic data at all and can only support static data . The supporting data dynamics is also of more importance for cloud storage systems. The main protocol will be presented

based on static data for the reason of brevity and clarity.

B. HLA-based solution

HLA technique can be used to effectively support public auditability without having to retrieve the data blocks themselves. HLAs also having some unforgivable verification metadata that authenticate the integrity of a data block. The main difference is that HLAs can be aggregated. It is also possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks.

V. PRIVACY-PRESERVING PUBLIC AUDITING SCHEME

To achieve privacy-preserving public auditing, it uniquely integrate the homomorphic linear authenticator with random masking technique. In this 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. 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. Scheme design makes use of a public key-based HLA, to equip the auditing protocol with public auditability. Specifically it uses the HLA proposed in which is based on the short signature scheme proposed by Boneh, Lynn, and Shacham (hereinafter referred as BLS signature) [19].

VI. BATCH AUDITING FOR MULTICLIENT DATA

As cloud servers may concurrently handle multiple verification sessions from different clients, given K signatures on K distinct data files from K clients, it is more advantageous to aggregate all these signatures into a single short one and verify it at one time. To achieve this goal, it allow for provable data updates and verification in a multi client system. The key idea is to use the Bilinear aggregate signature scheme, as in the BLS based construction, the aggregate signature scheme allows the creation of signatures on arbitrary distinct messages. Moreover, it supports the aggregation of multiple signatures by distinct signers on distinct messages into a single short signature, and thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages

CONCLUSION

This paper discusses a privacy-preserving public auditing system for data storage security in cloud computing. This utilizes 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.

This paper also discusses the different security and performance challenges such as the public auditing, privacy preserving, batch auditing.

ACKNOWLEDGMENT

I am thankful to Dr. Shyamrao V. Gumaste Sir, Prof. Sandip A. Kahate Sir and Prof. Gajanan S. Deokate for their guidance. I also thank the college authorities for providing the required infrastructure and support. Finally, I would like to extend a heartfelt gratitude to my friends and family members.

References

[1]Cong Wang, Sherman S.M. Chow, Qian Wang,Kui Ren, Wenjing Lou," Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE TRANSACTIONS ON COMPUTERS VOL. 62, NO. 2, FEBRUARY 2013

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[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] 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.

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

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

[8] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. 15th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT), pp. 319-333, 2009.

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

[10] 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.

[11] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198, 2009.
[12] Amazon.com, "Amazon s3 Availability Event: July 20, 2008,"
[13] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.
[14] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik,"Scalable and Efficient Provable Data Possession," Proc. Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.
[15] 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.
[16] 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.