Isolation support of Public Auditing for Secure Cloud Storage

[1] N. Satyavani , [2]G.Rama Swamy

[1]M.Tech Research Scholar, Department of CSE,
[2] Professor, Department of CSE
Priyadarshini Institute of Technology & Science, Chintalapudi

Abstract—*Cloud computing is new drift now to store large amount of data and distribution of data. There are so many schemes to distribute the information securely. The cloud data storage has many benefits over local data storage. Users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. The problem is that ensuring data security and integrity of data of user. So here, we are having public audit ability for data storage users that can restore to a third-party auditor (TPA) to check the integrity of data. Here, this paper gives the various issues related to security during the TPA auditing. Without appropriate security and privacy solutions designed for cloud computing paradigm could become a big failure. Through a formal analysis, the correctness and security of the protocol is being verified.*

Index Terms– Cloud computing, Data storage, Security, Integrity.

Introduction

Storing data in the cloud has become a drift. Increasing the number of clients store their important data in remote servers in the cloud, without leaving a copy in their local computers. Sometimes the data stored in the cloud is so important that the clients must ensure it is not lost or despoiled. While it is easy to check data reliability after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Hence, a lot of works have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data. Remote data integrity checking is first introduced in the cloud storage which independently suggests RSA based methods for solving this problem. Propose a remote storage auditing method based on pre-computed dispute

response pairs.

While cloud computing creates these profit more consistent than personal computing mechanisms, they are Second, there do exist different brainwaves for CSP to perform unfaithfully near the cloud users regarding their outsourced data position their data. As a result, the truth of the data in the cloud is being still looking the broad range of both internal and external or even hide data loss events to keep a reputation. In short, although outsourcing data to the cloud is efficiently attractive for long-term large-scale storage, it does not immediately offer any assurance on data reliability and availability. This problem, if not appropriately tackled, may delay the success of cloud architecture. Threats are much more Examples of outages and security rifts of for examples, CSP might get back to the storage for financial reasons by disposal data that have not been or are rarely accessed, Interesting than ever, it also takes new and demanding security threats toward users' outsourced data. While cloud service suppliers (CSP) are split managerial entities, data outsourcing is really surrendering users eventual manage over the chance of important cloud services appear from time to time. Located at danger due to the following reasons. For data reliability. Although the communications below the cloud First of all, potent and

In this we deal with the problem of applying a protocol for obtaining a proof of data possession in the cloud sometimes referred to as Proof of irretrievability (POR).This problem tries to find and validate a proof that the data that is stored by a user at a remote data storage in the cloud (called cloud storage archives or simply archives) is not modified by the records and thereby the integrity of the data is assured. Consider the large size of the outsourced electronic data and the users controlled resource capability, the core of the difficulty can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files. As data generation is far Outsourcing data storage it proves costly for small firms to regularly update their hardware whenever extra data is created. Also maintaining the storages can be a tricky task. It can also assure a reliable storage of significant data by remaining multiple copies of the data thereby dropping the chance of losing data by hardware failures. To fully make sure the data integrity and save the cloud clients' computation resources as well as online burden, it is of critical significance to allow 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 required. The TPA, who has proficiency and capabilities that clients do not, can periodically verify the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and reasonable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to estimate 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 negotiation purposes. To deal with these problems, our work uses the technique of public key-based Homomorphic linear authenticator (or HLA for short) , which allows TPA to do the auditing without commanding the local copy of data and thus radically reduces the communication and computation overhead as related to the straightforward data auditing methods. By integrating the HLA with random covering, 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 additional advantage our design for the group auditing. Specifically, our contribution can be reviewed as the following three features:

We stimulate the public auditing system of data storage refuge in cloud computing and provide a security in data storage auditing protocol. Our plan enables an external auditor to audit client's cloud data without knowledge the data content.

1. To the most excellent of our knowledge, our plan is the first to maintain scalable and efficient security in data storage public storage auditing in cloud. Specifically, our method achieves batch auditing where multiple assigned auditing jobs from different clients can be done simultaneously by the TPA in a privacy-preserving manner.
2. We prove the security and justify the presentation of our future methods through real experiments and comparisons with the state of the art.

The rest of the paper is prepared as follows: Section 2 presents the system and risk model, and our intend goals. Then, we provide the detailed explanation of our method in Section 3. Section 4 provides the security analysis and performance evaluation. Section 5 presents additional discussions on a zero-knowledge auditing protocol, followed by Section 6 that indicates the related work. Finally, Section 7 provides the concluding statement of the whole paper.

2 PROBLEM STATEMENTS

2.1 The System and Threat Model

Let us take a cloud data storage service engaging three different things, as demonstrated in Fig. 1: The cloud client is a person, who supplies large amount of data or files on cloud server. Cloud server is a place where we are saving cloud data and that data will be managed by the cloud service provider.

The third-party auditor will perform the auditing on clients demand for storage correctness and reliability of data. In the cloud pattern, by placing the large data files on the isolated servers, the clients can be relieved of the load of storage and computation. As clients no longer possess their data locally, it is of serious significance for the clients to ensure that their data are being properly stored and

maintained. That is, clients should be prepared with certain security means so that they can periodically verify the correctness of the isolated data even without the existence of local copies. In case those clients do not necessarily have the time, feasibility or resources to monitor their data, they can assign the monitoring task to a trusted TPA.

The verifier before saving the file at the archive preprocesses the file and adds some Meta data to the file and saves at the archive. At the time of confirmation the verifier uses this Meta data to confirm the integrity of the data. It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally customized or deleted. It does not stop the archive from adjusting the data.

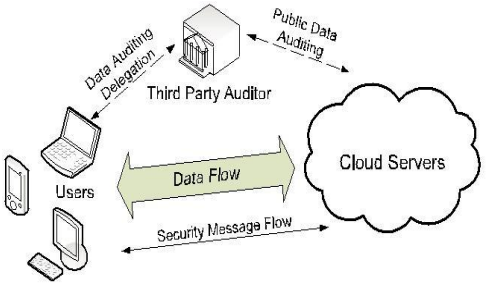


Fig. 1. The architecture of cloud data storage service.

2.2 Design Goals

To allow security in data storage using public auditing for cloud computing under the aforesaid model, our protocol design should get the following security and performance guarantees:

1. Lightweight: to allow TPA to do auditing with the less communication and computation overhead.
2. Batch auditing: to enable TPA with secure and capable auditing ability to manage with multiple auditing allocations from possibly large number of different clients simultaneously.
3. Public auditability: to allow TPA to check the correctness of the cloud data on demand without accessing a copy of the whole data or introducing additional online burden to the cloud clients.
4. Privacy preserving: to ensure that the TPA cannot read clients' data content during the auditing process.
5. Storage correctness: The data stored on a cloud is as it. No data modification is done.

3 THE PROPOSED SCHEMES

This section presents our public auditing scheme which provides a complete outsourcing solution of data not only the data itself, but also its integrity checking. After introducing notations and brief preliminaries, we start from an overview of our public auditing system and discuss two straightforward schemes and their demerits. Then, we present our main scheme and show how to extent our main scheme to support batch auditing for the TPA upon delegations from multiple users. Finally, we discuss how to generalize our privacy-preserving public auditing scheme and its support of data dynamics.

3.1 Notation and Preliminaries

- F- the data file to be outsourced, denoted as a sequence of n blocks $m_1, \ldots, m_i, \ldots, m_n \in Z_p$ for some large prime p.
- $MAC_{(.)}(.)$—message authentication code (MAC) function, defined as: $K * \{0, 1\}^x \rightarrow \{0, 1\}^l$ where K denotes the key space.
- H(.), h(.)—cryptographic hash functions.

Now we introduce some needed cryptographic back-ground for our proposed scheme.

Bilinear Map: Let G_1, G_2, and G_T be multiplicative cyclic groups of prime order p. Let g_1 and g_2 be generators of G_1 and G_2, respectively. A bilinear map is a map $e : G_1 * G_2 \rightarrow G_T$ such that for all $u \in G_1$, $v \in G_2$ and a, b $\in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$. This bilinearity implies that for any $u_1, u_2 \in G_1$, $v \in G_2$, $e(u_1 . u_2, v) = e(u_1, v). e(u_2, v)$. Of course, there exists and efficiently computable algorithm for computing e and the map should be nontrivial, i.e., e is no degenerate: $e(g_1, g_2) \neq 1$.

3.2 Definitions and Framework

In this section we follow regular description of earlier proposed schemes in the background of isolated data integrity verifying and adapt the structure of maintain a security using public auditing system.

A public auditing system consists of two stages, Setup and Audit:

- **Setup**: The client may assign the public and secret constraints of the system by accomplishing KeyGen, and preprocesses the data file F by using SigGen to make the verification metadata. As part of preprocessing, the user may alter the data file F by increasing it or counting extra metadata to be saved at server.

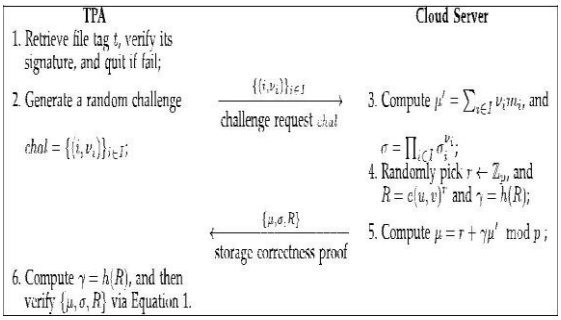
- **Audit:** The TPA gives an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. Suppose the structure of TPA is position less, i.e., TPA does not require to keep and update position between audits, which is a attractive possessions particularly in the public auditing system. We can't imagine the design of TPA and additional property in data file.

HLA-based solution: To efficiently support public auditability without having to regain the data blocks themselves, the HLA method can be used. HLAs, like MACs, are also some unforgeable checking the metadata that validate the integrity of a data block. The difference is that HLAs can be collective.

Though allocating efficient data auditing and using only stable bandwidth, the direct implementation of these HLA-based methods is still not suitable for our principles. This is because the linear combination of blocks, $\mu = \sum_i v_i . m_i$, may potentially expose user data information to TPA, and breaks the data security maintenance guarantee. Specifically, by challenging the same set of c block $m_1, m_2, \ldots, m_c$ using c different sets of random coefficients $\{v_i\}$, TPA can gather c different

linear combinations $\mu_1, \ldots, \mu_e$. With $\{\mu_i\}$ and $\{v_i\}$, TPA can derive the client's data $m_1; m_2; \ldots; m_c$ by simply solving a system of linear equations.

Table 1
The Privacy-Preserving Public Auditing Protocol



3.4 Privacy-Preserving Public Auditing Scheme

Overview: To reach security maintenance using public auditing, we suggest to uniquely adding the Homomorphic linear authenticator with random covering method. With random covering, the TPA no longer has all the required information to construct a correct group of linear equations and therefore cannot derive the client's data content, no matter how many linear combinations of the same set of file blocks can be collected.

Scheme details: Let G_1, G_2, and G_T be multiplicative cyclic groups of prime order p, and e : $G_1 * G_2 \rightarrow G_T$ be a bilinear map as began in preliminaries. Let g be a generator of G_2. H(.) is a secure map-to-point hash function: $\{0, 1\}^* \rightarrow G_1$, which maps strings uniformly to G_1. Another hash function h(.) : $G_T \rightarrow Z_p$ maps group element of G_T uniformly to Z_p. Our scheme is as follows:

Setup Phase: The cloud client runs KeyGen to make the public and secret parameters. Specifically, the user chooses a random signing key pair (spk, ssk), a random $x \leftarrow Z_p$, a random element $u \leftarrow G_1$, and computes $v \leftarrow g^x$. The secrete parameter is sk = (x, ssk) and the public parameters are pk = (spk, v, g, u, e(u, v)).

3.5 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 achieve 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. The details are described as follows:

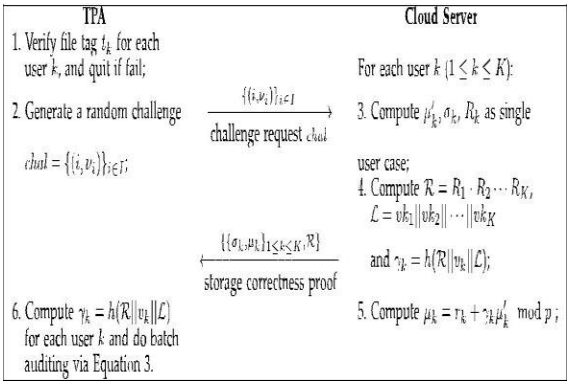
Efficiency improvement: As shown, batch auditing not only permits TPA to do the multiple auditing jobs simultaneously, but also greatly decreases the computation cost on the TPA side.

Identification of invalid responses: The

checking equation catches when all the responses are suitable, and fails with high chance when there is even one single invalid response in the batch auditing, as we will show in Section.

Table 2

The Batch Auditing Protocol



3.6 Support for Data Dynamics

In cloud computing, outsourced data might not only be admission but also updated regularly by clients for individual application principles. Hence, supporting data dynamics for security maintenance using public auditing is also of paramount importance. Application to version control system: The above scheme permits TPA to always maintain the new tree root for auditing the updated data file. But it is worth noting that our mechanism can be simply comprehensive to work with version control system, where both current and previous versions of the data file F and the corresponding authenticators are stored and need to be audited on demand.

3.7 Generalization

In the above mechanism, our protocol is depending on the HLA. It has been shown that HLA can be built by Homomorphic identification protocols. One may concern the random using method we used to build the corresponding zero knowledge proof for different Homomorphic recognition protocols. Therefore, our security maintenance using public auditing system for secure data storage can be generalized depending on other complexity statements, such as factoring.

4 EVALUATIONS

4.1 Security Analysis

We evaluate the security of the existing system by examining its completion of the security guarantee explained in Section 2.2, namely, the storage correctness and security maintenance property. We begin from the single client case, where our main result is originated. Then, we show the security guarantee of batch auditing for the TPA in multiuser setting.

4.1.1 Storage Correctness Guarantee

We need to prove that the cloud server cannot generate valid response for the TPA without faithfully storing the data, as captured by Theorem 1.

Theorem 1: If the cloud server passes the Audit phase, it must indeed possess the specified data intact as it is.

4.1.2 Privacy-Preserving Guarantee

The below theorem shows that TPA cannot derive users' data from the information collected during auditing.

Theorem 2: From the server's response {σ,μ,R}, TPA cannot recover μ'.

4.1.3 Security Guarantee for Batch Auditing

Now, we show that our way of extending our result to a multiuser setting will not affect the aforementioned security insurance, as shown in Theorem 3.Our batch auditing protocol achieves the same storage correctness and privacy-preserving guarantee as in the single-user case.

4.2 Performance Analysis

Now we take details some presentation results of our trials. We consider our auditing mechanism occurs between a committed TPA and some cloud storage node, where client's data are outsourced to. In our experiment, the TPA/user side process is implemented on a workstation with an Intel Core 2 processor running at 1.86 GHz, 2,048 MB of RAM, and a 7,200 RPM Western Digital 250 GB Serial ATA drive.

TABLE 3: Notation of Cryptographic Operations

$Hash^t_{\mathbb{G}_1}$	hash t values into the group $\mathbb{G}_1$.
$Mult^t_{\mathbb{G}}$	t multiplications in group $\mathbb{G}$.
$Exp^t_{\mathbb{G}}(\ell)$	t exponentiations g^{a_i}, for $g \in \mathbb{G}$, $\lvert a_i \rvert = \ell$.
$m\text{-}MultExp^t_{\mathbb{G}}(\ell)$	t m-term exponentiations $\prod_{i=1}^{m} g^{a_i}$.
$Pair^t_{\mathbb{G}_1,\mathbb{G}_2}$	t pairings $e(u_i, g_i)$, where $u_i \in \mathbb{G}_1$, $g_i \in \mathbb{G}_2$.
$m\text{-}MultPair^t_{\mathbb{G}_1,\mathbb{G}_2}$	t m-term pairings $\prod_{i=1}^{m} e(u_i, g_i)$.

4.2.1 Cost of Privacy-Preserving Protocol

We begin by estimating the cost in terms of basic cryptographic tasks (refer to Table 3 for notations). Suppose there are c random blocks specified in the challenge message chal during the Audit phase. Under this setting, we quantify the cost introduced by the privacy-preserving auditing in terms of server computation, auditor computation as well as communication overhead. Since the difference for choices on s has been discussed previously, in the following privacy-preserving cost analysis we only give the atomic operation analysis for the case s ¼ 1 for simplicity. The analysis for the case of s = 10 follows similarly and is thus omitted.

4.2.2 Batch Auditing Efficiency

Discussion in Section 3.5 presents an asymptotic efficiency analysis on the batch auditing, by permitting for only the total number of pairing tasks. However, on the practical side, there are additional less expensive tasks required for batching, such as modular exponentiations and multiplications.

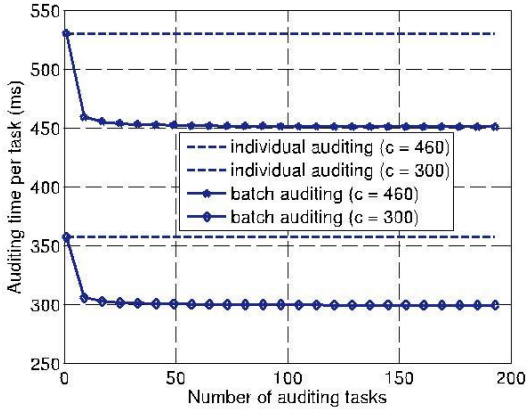


Fig.2.Comparison on auditing time between batch and individual auditing: Per task auditing time denotes the total auditing time divided by the number of tasks.

It can be shown that compared to individual auditing, batch auditing indeed helps reducing

the TPA's computation cost, as more than 15 percent of per-task auditing time is saved.

4.2.3 Sorting Out Invalid Responses

Now, we use experiment to clarify the efficiency of our recursive binary search approach for the TPA to arrange out the invalid responses for negative batch auditing result, as discussed in Section 3.5. This experiment is tightly pertained to the work in [20], which evaluates the batch verification of various short signatures.

5. CONCLUSIONS

In this paper, we 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 our 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. Our 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.

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/cloud-computing/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 Comput-ing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009

[4] Cloud Security Alliance, "Top Threats to Cloud Computing,"

[5] http://www.cloudsecurityalliance.org , 2010.

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

deletions/, 2006.

[7] J. Kincaid, "Medi aMax/TheLinkup Closes Its Doors," http:// www.techcrunch.com/2008/07/10/media maxthelinkup-closes-its-doors/, July 2008.

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

[9] 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.

[10] 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.

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

[12] 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.

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

[14] Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107, Dec. 2008.

[15] 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.

[16] 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.

[17] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," http://aspe.hhs.gov/ admnsimp/pl104191.htm, 1996.