ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

A Review on Privacy-Preserving Public Auditing for Storage Cloud

[I]Mukund N. Kulkarni, [II]Prof. Bharat A. Tidke
[I]P.G. Student, Flora Institute of Technology, Khopi, Pune, India
[II]Asst.Prof. & H.O.D. of Computer Engg., Flora Institute of Technology, Khopi, Pune, India

Abstract
Storing data in the cloud has become a trend. An increasing 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 corrupted. While it is easy to check data integrity 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. Many works focus on providing three advanced features for remote data integrity checking protocols: data dynamic, public verifiability and privacy against verifiers. The system in support data dynamics at the block level, including block insertion, block modification and block deletion. It supports data append operation, in addition it can be easily adapted to support data dynamics by using the techniques. On the other hand, it support public verifiability, by which anyone can perform the integrity checking operation. The system in support privacy against third party verifiers.

Keywords
Cloud Computing, Cloud Storage, Privacy-Preserving Data storage, privacy-preserving, public auditability, batch verification, zero-knowledge.

I. Introduction

In recent years, the popularity of cloud storage services has increased dramatically. Most business processes have been digitalized, i.e., information such as communication data, accounts, contracts, advertising material, construction or business plans only exists in digital form. For a company, the loss of data could ruin the basis for business. Additionally, companies are legally obliged to preserve tax records for a certain period (3-10 years), and to leave them available to the fiscal authorities. This requires secure methods of preserving important data in order to prevent unrecoverable data loss, whilst constantly keeping up with increasing demands for storage space. It is necessary to regularly make extra copies of the information, so as to be able to restore it to an earlier version if need be. These copies further escalate the demand for storage space. Additional requirements arise from the variety of devices used to access the data simultaneously [1]. Private and business users demand an easy way to synchronize and access their data independent of both device and location. The software providing these features must also be tailored to the needs of the individual with no technical background. In order to meet these demands, companies make large investments into their IT infrastructure.

Additional hardware and software is required, as well as staff for its operation and maintenance. Larger companies might have to consider building a dedicated data center. These expenses conflict with the continuing need to reduce costs in order to stay competitive. Cloud storage services offer user-friendly, easily accessible and money-saving ways of storing and automatically backing up arbitrary data. These services are available on-demand on the Internet [2]. A customer simply accesses the website of a cloud storage provider and rents storage space as necessary by selecting one of the provider's packages. If the use of cloud storage services carries such great advantages, why are individuals and companies alike still hesitant to entrust their data to the cloud? Usage of a cloud storage provider basically means entrusting data to a third party where no prior relationship based on trust has been established. Individuals who upload personal information to the cloud want to be sure that only certain people are able to access it. This should also exclude the provider, since there is no justify able reason for it to access the data. Companies may entrust containing sensitive business data and valuable intellectual property which may be of great interest for industrial reasoning [5]. The unauthorized disclosure of customer information, business secrets or research data poses a serious threat to a company's business. In addition, compliance requirements with both internal security guidelines and legal regulations have to be met. The cloud storage provider may be subject to different legal regulations than the user.

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic a user can have as much or as little of a service as they want at any given time and the service is fully managed by the cloud service provider (the consumer needs nothing but a personal computer and Internet access).The advantage of cloud is cost savings. The prime disadvantage is security. Cloud computing is used by many software industries nowadays. Since the security is not provided in cloud, many companies adopt their unique security structure. For e.g. Amazon has its own security structure. Since the data placed in the cloud is accessible to everyone, security is not guaranteed. To ensure security, cryptographic techniques cannot be directly adopted. Sometimes the cloud service provider may hide the data corruptions to maintain the reputation. To avoid this problem, this paper introduce an effective trusted third party auditor to audit the user‘s outsourced data when needed.

II. Literature Survey

Considering the large size of the outsourced data and the user‘s constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users

may not want to go through the complexity in verifying the data integrity [1,2].

The author [3] presents an algorithm to reallocate the placement of virtual machines in a cloud for better performance and resource utilization. Periodically, their algorithm identifies three sets of machines, senders (the most over utilized physical machines), receivers (the most underutilized physical machines), and guests to migrate (the most over utilized virtual machines hosted on the senders). The algorithm only considers CPU utilization to identify the over utilized and under-utilized machines. The author use a queuing network model to identify the optimal configuration of a multi-tier application hosted in a virtualized data center. However, they pre-deploy the application then identify the values of required parameters for their model. An online controller ensures that any failure of a tier is resolved without degradation of application performance [4]. Whenever a failure is detected, they use a simple select-reject technique to identify where to launch the tier. Their select-reject technique iteratively identifies a physical machine that has enough CPU capacity to host the tier and, to maximize fault tolerance, does not host another instance of the same tier. The author proposes a new system for identification and mitigation of hotspot (overloaded) virtual machines in a Xen-based virtualized data center. It gathers usage statistics for physical and virtual machines [5]. These profiles are used with prediction techniques to identify the hotspots. Then, it identifies the requirements of the overloaded virtual machines and uses a greedy algorithm to migrate virtual machines to the appropriate physical machines.

The author proposes an efficient construction, which offers not only the conventional query privacy, but also the following new features. The system allows a group of users, each possessing a distinct secret key, to insert their encrypted data records to the database while every user in the group is able to search all the records using her chosen keywords with the assistance from a semi-trusted database server [10]. The system allows the user management of the database owner organization to dynamically and efficiently revoke users. Our revocation does not require distribution of new keys, nor needs to update the encrypted database including the indexes. After a revocation, the revoked users are no longer able to search the database, while the revocation process is transparent to those non-revoked users [9]. The system also allows for dynamic user enrollment, since a user joining does not affect other user‘s settings.

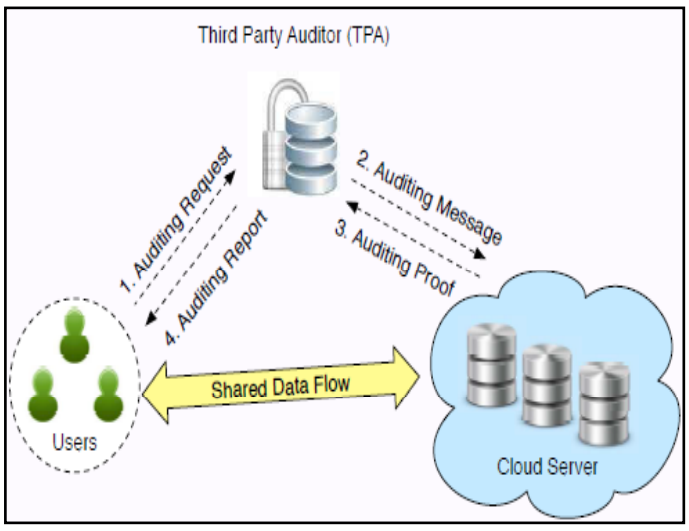

Fig. 1 : Existing TPA System

III. Problem Definition

A. System Threat Model

Consider a cloud data storage service involving three different entities, as illustrated in Fig. 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, 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.

The Cloud security responsibilities can be taken on by the customer, if he is managing the cloud, but in the case of a public cloud, such responsibilities are more on the cloud provider and the customer can just try to assess if the cloud provider is able to provide security. Cloud data storage service involving three different entities. 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 cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (do 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 security on behalf of the user upon request. Cloud users dynamically interact with the CS to access and update their stored data for various application purposes. The traditional cryptographic technologies for data integrity and availability, cannot work on the outsourced data without a local copy of data. It is not a practical solution for data validation by downloading them due to the expensive communications, especially for large size files. The ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Therefore, it is crucial to realize public auditability for CSS, so that data owners may resort to a third party auditor (TPA), who has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and credibility in clouds. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as [11] does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, 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. 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. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user’s outsourced data towards TPA through the auditing protocol should be prohibited.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate.

IV. Design Goals

The privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should follow the security and performance.

Public Audit: It allows TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data.

Storage Consistency: the data in cloud server that can pass the audit from TPA without indeed storing users' data intact.

Privacy-Preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process.

Batch Auditing: It enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.

Light Weight: It allow TPA to perform auditing with minimum communication and computation overhead.

A. Definitions and Framework

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof from the cloud server. Running a public auditing system consists of two phases, Setup and Audit:

1. 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 delete its local copy. As part of pre-processing, the user may alter; the data file F by expanding it or including additional metadata to be stored at server.

2. 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 from a function of the stored data file F and its verification metadata by executing GenProof. The TPA then verifies the response via VerifyProof. Our framework assumes the TPA is stateless, which is a desirable property achieved by our proposed solution. The TPA is stateless, i.e., TPA does not need to maintain and update state between audits, which is a desirable property in the public auditing scheme [1].

V. Privacy Preserving Public Auditing

The privacy-preserving public auditing, this paper propose to uniquely integrate the homomorphic non-linear authenticator with random masking technique. In our protocol, the non-linear 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 non-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 [13], which is based on the short signature scheme.

A. Periodic Sample Audit

In the Cloud Server environment random "sampling" checking greatly reduces the workload of audit services, while still achieve an effective detection of misbehavior. Thus, the probabilistic audit on sampling checking is preferable to realize the abnormality detection in a timely manner, as well as rationally allocate resources. The fragment structure can provide the support of probabilistic audit as well: given a random chosen challenge (or query) $Q = \{(i, v_i)\}\ i \varepsilon I$, where I is a subset of the block indices and v_i is a random coefficient, an efficient algorithm is used to produce a constant-size response $(\mu_1, \mu_2, \ldots\ldots\ldots \mu_i)$, where μ_i comes from all $\{m_k, i, v_k\} k \varepsilon I$ and all $\{k, v_k\}\ k \varepsilon I$. Generally, this algorithm relies on homomorphic properties to aggregate data and tags into a constant size response, which minimizes network communication. Since the single sampling checking may overlook a very small number of data abnormality, this paper propose a periodic sampling approach to audit outsourcing data, which is called as Periodic Sampling Audit. In this way, the audit activities are efficiently scheduled in an audit period, and a TPA needs merely access small portions of file to perform audit in each activity. Therefore, this method can detect the exceptions in time, and reduce the sampling numbers in each audit.

B. Security Consistency for Batch Auditing

The way to describe the result to a multi-user setting will not affect the aforementioned security. The batch auditing protocol achieves the same storage correctness and privacy preserving guarantee as in the single-user case.

The privacy-preserving guarantee in the multiuser setting. The storage correctness guarantee, this is going to reduce it to the single-user case. Here proposed to use the forking technique for the verification equation for the batch audits involves K challenges from the random block. This time it needs to ensure that all the other $K - 1$ challenges are determined before the forking of the concerned random oracle response. This can be done using the idea in [4]. As soon as the adversary issues the very first random oracle query for $i = h(R\|v_i\|L)$ for any $i \varepsilon [1,K]$, the simulator immediately determines the values $j = h(R\|v_j\|L)$ for all $j \varepsilon [1, K]$. This is possible since they are all using the same R and L. Now, all but one of the k's are equal, so a valid response can be extracted similar to the single-user case.

VI. Conclusion

This paper propose a privacy-preserving public auditing system for data storage security in Cloud Computing. Which utilize the homomorphic non-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 security. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, which further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a

batch manner for better efficiency. Extensive analysis shows that our schemes are probably secure and highly efficient.

References

[1] *C. Wang et al., 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, 2009, http://csrc.nist.gov/groups/SNS/cloud-computing/index.html.*
[3] *M. Armbrust et al., Above the Clouds: A Berkeley View of Cloud Computing, Univ. California, Berkeley, Tech. Rep. UCBEECS-2009-28, Feb. 2009.*
[4] *Amazon.com, Amazon s3 Availability Event: July 20, 2008, July 2008, http://status.aws.amazon.com/s3-20080720.html.*
[5] *M. Arrington, Gmail Disaster: Reports of Mass Email Deletions, Dec. 2006.*
[6] *T. Ristenpart et al., Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds, Proc. 16th ACM Conf. Computer and Communications Security, ACM Press, 2009, pp. 199–212.*
[7] *G. Klein et al., Formal Verification of an OS Kernel, Proc. ACM SIGOPS 22nd Symp. Operating Systems Principles (SOSP 09), ACM Press, 2009, pp. 207–220.*
[8] *Joseph, Randy Katz, Above the Clouds: A Berkeley View of Cloud Computing, University of California Electrical Engineering & Computer Science, February 10th, 2009.*
[9] *Patel, Chandrakant D., Shah, Amip J., Cost Model for Planning, Development, and Operation of a Data Center, Internet Systems and Storage Laboratory, HP Laboratories, Palo Alto, June 9, 2005.*
[10] *N. Leavitt, "Is Cloud Computing Really Ready for Prime Time?" Computing Now, pp. 15-20, 2009.*
[11] *Anthes, Gary, Data Centers Get a Makeover‖, Computerworld news article, published November 1, 2005.*
[12] *Hughes, Ron, The data center of the future-Part 1-Current trends, The Data Center Journal news article, published May17, 2005.*
[13] *Fichera, Richard, Power And Cooling Heat Up The Data Center, Forrester Research, Inc. March 8, 2006.*
[14] *Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.*
[15] *G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores,"in Proc. of CCS'07, 2007, pp. 598–609.*
[16] *H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt, vol. 5350, Dec 2008, pp. 90–107.*