Secure and Dependable Cloud Services for TPA in Cloud Computing

Nandeesh.B.B, Ganesh Kumar R, Jitendranath Mungara

Abstract: ***The cloud storage provides users to easily store their data and enjoy the good quality cloud applications need not install in local hardware and software system. So benefits are clear, such a service is also gives users' physical control of their outsourced data, which provides control over security problems towards the correctness of the storage data in the cloud. In order to do this new problem and further achieve a secure and dependable cloud storage services, we propose in this paper a flexible distributed storage integrity auditing mechanism, using the homomorphism token and distributed erasure-coded data. We are also proposing allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of hacker information. And securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. This shows the proposed scheme is highly efficient and data modification attack, and even server colluding attacks.***

Index Terms—Data integrity, dependable distributed storage, error localization, data dynamics, Cloud Computing.

I. INTRODUCTION

Many technologies are coming in the Cloud Computing, which provides Internet-based service and use of computer technology. This is cheaper and more strong processors, together with the software as a service (SaaS) computing architecture, are transforming data into data centers on huge scale. The increasing network and flexible network connections make it even possible that users can now use high quality services from data and provides remote on data centers. Storing data into the cloud offers great help to users since they don't have to care about the problems of hardware problems. The providers of Cloud Computing vendors, Rock star cloud and Amazon Elastic Compute Cloud (EC2) [2] are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is avoids the responsibility of local machines for data maintenance at the same time. As a result, users are at the interest of their cloud service providers for the availability and integrity of their data [3]. On the one hand, although the cloud services are much more powerful and reliable than personal computing devices and broad range of both internal and external threats for data integrity still exist.

Manuscript Received on August 08, 2012.

Nandeesh.B.B, 4th Sem, M-Tech Dept of CSE, CMR Institute of Technology, Bangalore, India
Mr. Ganesh Kumar R, HOD, Dept of ISE, CMRIT, Bangalore, India
Dr. Jitendranath Mungara, Prof & Dean, PG Programme, Dept of CSE &ISE, CMRIT, Bangalore, India

Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time [4]–[8]. On the other hand, since users may not keep a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion [9]. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation [10]–[12].Our work is among the first few ones in this field to consider distributed data storage security in Cloud Computing. Our contribution can be summarized as the following three aspects:

1) Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s).
2) Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
3) The experiment results demonstrate the proposed scheme is highly efficient. Extensive security analysis shows our scheme is resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

II. PROBLEM STATEMENTS

2.1 System Model

Representative network architecture for cloud storage service architecture is illustrated in Figure 1. Six different network entities can be identified as follows:

1) User: an entity, which is used to retrieve the data from cloud server.

2) Cloud Server (CS): an entity, which is managed by *cloud service provider* (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.).

3) Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

4) Owner: an entity, who has data to be stored in the cloud and relies on the cloud for data storage and computation, can be either enterprise or individual customers.

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure correcting code to further tolerate faults or server crash as user's data grows in size and importance. Thereafter, for application purposes, the user interacts with the 3 cloud servers via CSP to access or retrieve his data.

That is, users should be equipped with security means so that they can make continuous correctness assurance (to enforce cloud storage service-level agreement) of their stored data even without the existence of local copies. In case those users do not necessarily have the time, feasibility or resources to monitor their data online, they can delegate the data auditing tasks to an optional trusted TPA of their respective choices. However, to securely introduce such a TPA, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited.

2.2 Adversary Model

From user's perspective, the adversary model has to capture all kinds of threats towards his cloud data integrity. Because cloud data do not reside at user's local site but at CSP's address domain, these threats can come from two different sources: internal and external attacks. For internal attacks, a CSP can be self-interested, un-trusted and possibly malicious. Not only does it desire to move data that has not been or is rarely accessed to a lower tier of storage than agreed for monetary reasons, but it may also attempt to hide a data loss incident due to management errors, Byzantine failures and so on. For external attacks, data integrity threats may come from outsiders who are beyond the control domain of CSP, for example, the economically motivated attackers.

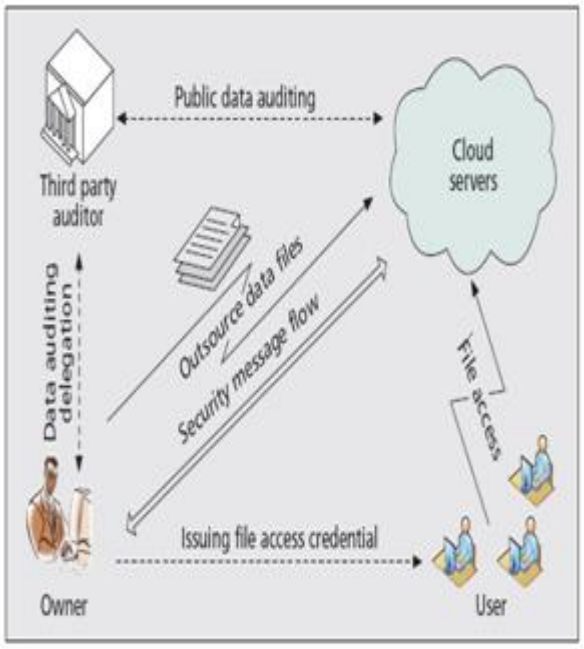
Therefore, we consider the adversary in our model has the following capabilities, which captures both external and internal threats towards the cloud data integrity. Specifically, the adversary is interested in continuously corrupting the user's data files stored on individual servers. Once a server is comprised, an adversary can pollute the original data files by modifying or introducing its own fraudulent data to prevent the original data.

2.3 Design Goals

To ensure the security and dependability for cloud data storage under the aforementioned adversary model, we aim to design efficient mechanisms for dynamic data verification and operation and achieve the following goals: (1) Storage correctness: to ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud. (2) Fast localization of data error: to effectively locate the malfunctioning server when data corruption has been detected. (3) Dynamic data support: to maintain the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud. (4) Dependability: to enhance data availability against Byzantine failures, malicious data modification and server colluding attacks, i.e. minimizing the effect brought by data errors or server failures. (5) Lightweight: to enable users to perform storage correctness checks with minimum overhead.

III. ENSURING CLOUD DATA STORAGE

In cloud data storage system, users store their data in the cloud and no longer possess the data locally.



The architecture of cloud data storage service.

Fig2.1: Architecture for cloud computing

Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures.

Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance, since it can always be the first step to fast recover the storage errors and/or identifying potential threats of external attacks.

To address these problems, our main scheme for ensuring cloud data storage is presented in this section. The first part of the section is devoted to a review of basic tools from coding theory that is needed in our scheme for file distribution across cloud servers. Then, the homomorphic token is introduced. The token computation function we are considering belongs to a family of universal hash function [24], chosen to preserve the homomorphic properties, which can be perfectly integrated with the verification of erasure-coded data [21] [25].Subsequently, it is shown how to derive a challenge response protocol for verifying the storage correctness as well as identifying misbehaving servers. The procedure for file retrieval and error recovery based on erasure correcting code is also outlined. Finally, we describe how to extend our scheme to third party auditing with only slight modification of the main design.

3.1 File Distribution Preparation

It is well known that erasure-correcting code may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, we rely on this technique to disperse the data file **F** redundantly across a set of n = m+ k distributed servers. An (m, k) Reed-Solomon erasure-correcting code is used to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the m+k data and parity vectors. By placing each of the m+k vectors on a different server, the original data file can survive the failure of any k of the m+k servers without any data loss, with a space overhead of k/m. For support of efficient sequential I/O to the original file, our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors is distributed across m + k different servers. Where **I** is a m × m identity matrix and **P** is the secret parity generation matrix with size m × k. Note that **A** is derived from a Vander monde matrix, thus it has the property that any m out of the m + k columns form an invertible matrix. By multiplying **F** by **A**, the user obtains the encoded file: **G** = **F** · **A** = (G(1),G(2), . . .,G(m),G(m+1), . . .,G(n)) = (F1, F2, . . . , Fm,G(m+1), . . . , G(n)) , where G(j) = (g(j) 1 , g(j) 2 , . . . , g(j) l)T (j ∈ {1, . . . , n}).As noticed, the multiplication reproduces the original data file vectors of **F** and the remaining part (G(m+1), . . . , G(n)) are k parity vectors generated based on **F**.

3.2 Challenge Token Pre-computation

In order to achieve assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the pre-computed verification tokens.

The main idea is as follows: before file distribution the user pre-computes a certain number of short verification tokens on individual vector G(j) (j ∈ {1, . . ., n}), each token covering a random subset of data blocks. Later, when the user wants to make sure the storage correctness for the data in the cloud, he challenges the cloud servers with a set of randomly generated block indices. Upon receiving challenge, each cloud server computes a short "signature" over the specified blocks and returns them to the user.

3.3 Towards Third Party Auditing

As discussed in our architecture, in case the user does not have the time, feasibility or resources to perform the storage correctness verification, he can optionally delegate this task to an independent third party auditor, making the cloud storage publicly verifiable. However, as pointed out by the recent work [27], [28], to securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy. Namely, TPA should not learn user's data content through the delegated data auditing. Now we show that with only slight modification, our protocol can support privacy-preserving third party auditing.

The new design is based on the observation of linear property of the parity vector blinding process. Recall that the reason of blinding process is for protection of the secret matrix **P** against cloud servers. However, this can be achieved either by blinding the parity vector or by blinding the data vector (we assume k < m).Thus, the overall computation overhead and communication overhead remains roughly the same.

Algorithm: Token pre Computation

1: Procedure
2: Choose a parameters l,n and function f,q;
3: Choose the number of t tokens;
4: Choose the number r of indices per verification;

5: Generate master key Kprp and challenge key Kchal;
6: for vector G(j), j←1,n do;
7: for round i← 1,t do;
8: Derive αi = fkchal (i) and k(i) prp from KPRP.
9: Compute v(j) i = _r q=1 αq i ∗ G(j)[φk(I prp (q)].
10: **end for**
11: **end for**
12: Store all the v i's locally.
13: **end procedure.**

IV. PERFORMANCE EVALUATIONS

We now assess the performance of the proposed storage auditing scheme. We focus on the cost of file distribution preparation as well as the token generation. Our experiment is conducted on a system with an Intel Core 2 processor running at 1.86 GHz, 2048 MB of RAM, and a 7200 RPM Western Digital 250 GB Serial ATA drive. Algorithms are implemented using open-source erasure coding library Jerasure [31] written in C. All results represent the mean of 20 trials.

4.1 File Distribution Preparation

As discussed, file distribution preparation includes the generation of parity vectors (the encoding part) as well as the corresponding parity blinding part. We consider two sets of different parameters for the (m, k) Reed- Solomon encoding, both of which work over GF(216).shows the total cost for preparing a 1 GB file before outsourcing. In the figure on the left, we set the number of data vectors m constant at 10, while decreasing the number of parity vectors k from 10 to 2. In the one on the right, we keep the total number of data and parity vectors m + k fixed at 22, and change the number of data vectors m from 18 to 10. From the figure, we can see the number k is the dominant factor for the cost of both parity generation and parity blinding.

4.2 Challenge Token Computation

This can be explained as follows: on the one hand, k determines how many parity vectors are required before data outsourcing, and the parity generation cost increases almost linearly with the growth of k; on the other hand, the growth of k means the larger number of parity blocks required to be blinded, which directly leads to more calls to our non-optimized PRF generation in C. By using more practical PRF constructions, such as HMAC [32], the parity blinding cost is expected to be further improved. Compared to the existing work [20], it can be shown from This is because in [20] an additional layer of error-correcting code has to be conducted on all the data and parity vectors right after the file distribution encoding. For the same reason, the two layer coding structure makes the solution in [20] more suitable for static data only, as any change to the contents of file **F** must propagate through the two-layer error correcting code, which entails both high communication and computation complexity. But in our scheme, the file update only affects the specific "rows" of the encoded file matrix, striking a good balance between both error resilience and data dynamics.

Although in our scheme the number of verification token t is a fixed priori determined before file distribution, we can overcome this issue by choosing sufficient large t in practice. For example, when t is selected to be 7300 and 14600, the data file can be verified every day for the next 20 years and 40 years, respectively, which should be of enough use in practice.

V. RELATED WORKS

Ji Hu; Klein,A. "A Benchmark of Transparent Data Encryption for Migration of Web Applications in the Cloud" Cloud computing is a new computing style which provides IT infrastructure and software as dynamic, scalable, and pay-per-use services. One of its characteristics is that enterprise data are hosted by storage service providers in the cloud. Singh, Y.; Kandah, F.; Weiyi Zhang "A secured cost-effective multi-cloud storage in cloud computing "cloud data storage redefines the security issues targeted on customer's outsourced data (data that is not stored/retrieved from the costumers own servers). In this work we observed that, from a customer's point of view, relying upon a solo SP for his outsourced data is not very promising. In addition, providing better privacy as well as ensuring data availability can be achieved by dividing the user's data block into data pieces and distributing them among the available SPs in such a way that no less than a threshold number of SPs can take part in successful retrieval of the whole data block. Erotokritou, S.; Nair, S.K.; Dimitrakos, T. "An Efficient Secure Shared Storage Service with Fault and Investigative Disruption Tolerance" We focus on the problem of securing distributed data storage in a cloud computing environment by

designing a specialized multi-tenant data-storage architecture.

The architecture we present not only provides high degrees of availability and confidentiality of customer data but is also able to offer these properties even after seizures of various parts of the infrastructure have been carried out through a judicial process. Shuai Zhang; Shufen Zhang; Xuebin Chen; Xiuzhen Huo "Cloud Computing Research and Development Trend" The concept of computing comes from grid, public computing and SaaS. It is a new method that shares basic framework. The basic principles of cloud computing is to make the computing be assigned in a great number of distributed computers, rather than local computer or remoter server. The running of the enterprise's data center is just like Internet. This makes the enterprise use the resource in the application that is needed, and access computer and storage system according to the requirement. Wenjun Luo; Guojing Bai "Efficient Sharing of Secure Cloud Storage Services" Suppose Bob, the boss in Company A, pays a secure cloud storage service and authorizes all the employees in that company to share such a service. There exists a user hierarchy: Bob is the user at the upper level and all the employees in the company are the users at the lower level. In this paper, we design and construct a scheme, which enables the user at the upper level to efficiently share the secure cloud storage services with all the users at the lower level.

VI. CONCLUSIONS & FUTURE SCOPE

In this paper, we investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. In location-based services, users with location-aware mobile devices are able to make queries about their surroundings anywhere and at any time. While this ubiquitous computing paradigm brings great convenience for information access, it also raises concerns over potential intrusion into user location privacy. To protect location privacy, one typical approach is to cloak user locations into spatial regions based on user-specified privacy requirements, and to transform location-based queries into region-based queries. In this paper, we identify and address three new issues concerning this location cloaking approach. First, we study the representation of cloaking regions and show that a circular region generally leads to a small result size for region based queries. Second, we develop a mobility-aware location cloaking technique to resist trace analysis attacks. Two cloaking algorithms, namely MaxAccu Cloak and MinComm Cloak, are designed based on different performance objectives. Finally, we develop an efficient polynomial algorithm for evaluating circular region- based kNN queries. Two query processing modes, namely bulk and progressive, are presented to return query results either all at once or in an incremental manner. Experimental results show that our proposed mobility-aware cloaking algorithms significantly improve the quality of location cloaking in terms of an entropy measure without compromising much on query latency or communication cost. Moreover, the progressive query processing mode achieves a shorter response time than the bulk mode by parallelizing the query evaluation and result transmission.

REFERENCES

1. Wang, C. "Privacy-Preserving Public Auditing for Secure Cloud Storage" computers IEEE transaction.
2. Cong Wang ; Kui Ren ; Wenjing Lou ; Jin Li **"**Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" International conference on may 2011.
3. Shuai Zhang; Shufen Zhang; Xuebin Chen; Xiuzhen Huo "cloud computing research and development" international conference on digital networks in 2010.
4. Ji Hu; Klein, A. " A Benchmark of transparent data encryption migration of web applications International conference on digital identifier.
5. Tang, Y.; Lee, P.; Lui, J.; Perlman, R. **"**Secure Overlay Cloud Storage with Access Control and Assured Deletion" transaction paper.
6. Min Chao Wang; Xing Wu; Wu Zhang; FuQiang Ding; Jun Zhou; GuoCai Pei "A conceptual platform of SLA in cloud computing IEEE 9th international conference.

7. Singh, Y.; Kandah, F.; Weiyi Zhang "A ecure cost effective multi cloud storage in cloud computing" on computer communication conference.
8. Syam Kumar, P.; Subramanian, R.; Thamizh Selvam, D. "Ensuring data storage security in cloud computing" International conference.
9. Ramachnadran, N. "The immenant convergence of the tech and cloud computing" in IEEE conference 2010.
10. Wenjun Luo; Guojing Bai "Efficient Sharing of Secure Cloud Storage Services" in international conference 2011.
11. Cong Wang; Qian Wang; Kui Ren; Wenjing Lou "Privacy-Preserving Public Auditing for Secure Cloud Storage" International conference in 2012.
12. Singh, Y.; Kandah, F.; Weiyi Zhang "A secured cost-effective multi-cloud storage in cloud computing".
13. Erotokritou, S.; Nair, S.K.; Dimitrakos, T. "An Efficient Secure Shared Storage Service with Fault and Investigative Disruption Tolerance" published in 2010.
14. Qian Wang; Cong Wang; Kui Ren; Wenjing Lou; Jin Li "Enabling Public Audit ability and Data Dynamics for Storage Security in Cloud Computing" in conference.
15. Takayama, K.; Yokota, H. "Performance and Reliability of a Revocation Method Utilizing Encrypted Backup Data" in 2011 conference.