International Journal of Engineering, Management, Humanities and Social Sciences Paradigms (IJEMHS)
(Volume 28, Issue 03) Publishing Month: August 2017
An Indexed and Referred Journal with Impact Factor: 2.75
ISSN: 2347-601X
www.ijemhs.com

An investigation on the depot assistance programme in the context of cloud computing

[1]Bibhuprakash Pati
Gandhi Institute of Excellent Technocrats, Bhubaneswar, India
[2]Sandeep Kumar Majhi
Samanta Chandra Sekhar Institute of Technology and Management, Koraput, Odisha, India

Abstract

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' Physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design 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 misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

Keywords**:** *Byzantine failure, homomorphic token, Security, Confidentiality*

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture,are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers.

Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) [1] 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 eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazon's S3 is such an example [2].

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Secondly, Cloud Computing is not just a third party data

warehouse. The data stored in the cloud may be frequently updated by the users, including insertion deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. However, this dynamic feature also Makes traditional integrity insurance techniques futile and entails new solutions. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world. However, such important area remains to be fully explored in the literature.

Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works [3]–[7]. These techniques, while can be useful to ensure the storage correctness without having users possessing data,can not address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols [8]–[10] for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors,i.e., the identification of the misbehaving server(s).

Our work is among the first few ones in this field to consider distributed data storage in Cloud Computing. Our contributions can be summarized as the following three aspects:

1) Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.

2) Unlike most prior works for ensuring remote data integrity,
The new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.

3) Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.
The rest of the paper is organized as follows. Section II
Introduces the system model, adversary model, our design goal and notations. Then we provide the detailed description of our scheme in Section III and IV. Section V gives the security analysis and performance evaluations, followed by Section VI which overviews the related work. Finally, Section VII gives the concluding remark of the whole paper.

II. PROBLEM STATEMENT

1. System Model

Representative network architecture for cloud data storage is illustrated in Figure 1. Three different

network entities can
Be identified as follows:

I) User: users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.

ii) Cloud Service Provider (CSP): a CSP, who has significant
Resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing systems.

iii) 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.

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 cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data. The most general forms of

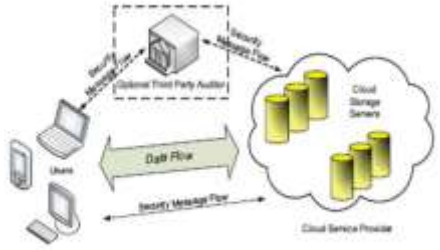

Fig. 1: Cloud data storage architecture

These operations we are considering are block update, delete, Insert and append. As users no longer possess their data locally, it is of critical importance to assure users that their data are being correctly stored and maintained. That is, users should be equipped with security means so that they can make continuous correctness assurance 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, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice with little overhead. Note that we don't address the issue of data privacy in this paper, as in Cloud computing, data privacy is orthogonal to the problem we study here.

2. Adversary Model

Security threats faced by cloud data storage can come from two different sources. On the one hand, a CSP can be self-interested, untrusted 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. On the other hand, there may also exist an economically motivated adversary, who has the capability to compromise a number of cloud data storage servers in different time intervals and subsequently is able to modify or delete users' data while remaining undetected by CSPs for a certain period. Specifically, we consider two types of adversary with different

Levels of capability in this paper:

Weak Adversary: The adversary is interested in 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 from being retrieved by the user.

Strong Adversary: This is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they are internally consistent. In fact, this is equivalent to the case where all servers are colluding together to hide a data loss or corruption incident.

III. CLOUD ARCHITECTURE

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over loose coupling mechanism such as messaging queue. The two most significant components of cloud computing architecture are known as the front end and the back end. The front end is the part seen by the client, i.e., the computer user. This includes the client's network (or computer) and the applications used to access the cloud via a user interface such as a web browser. The back end of the cloud computing architecture is the cloud itself, comprising various computers, servers and data storage devices.

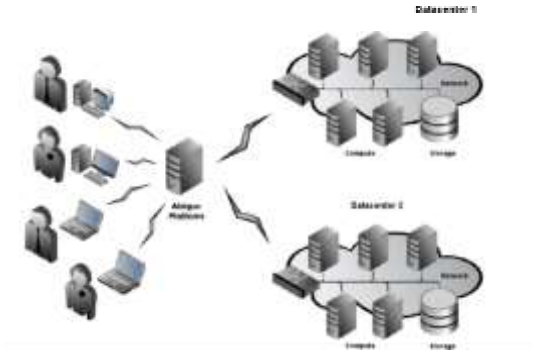

Fig. 2: Cloud data storage architecture

Representative network architecture for cloud data storage is illustrated in Figure2.Three different network entities can be identified as follows: Users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations. A CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing system, 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.

IV. DESIGN METHODOLOGIES

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity. Part of data design may occur in conjunction with the design of software architecture. More detailed occurs as each software component is designed. The architectural design data design defines the relationship between major structural elements of the software, the design patterns that can be used to achieve the requirements the system architecture, the interface

representation, and the component level detail. During each design activity, we apply basic principles that lead to high quality.

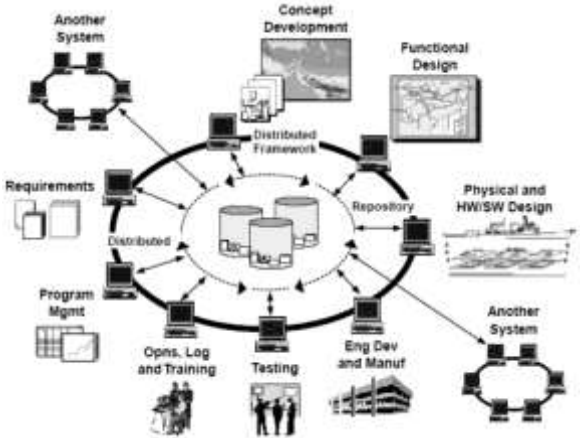

Figure.3 Translating Analysis model into Design model

To ensure the security and fidelity for cloud data storage under the aforesaid antagonist 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 unharmed all the time in the cloud.(2) Fast localization of data error: to effectively locate the faulty 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 intricate failures, malevolent 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.

V. ENSURING CLOUD DATA STORAGE

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. 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 be the first step to fast recover the storage errors. 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 [11], chosen to reserve the homomorphic properties, which can be perfectly integrated with the verification of erasure-coded data [8] [12].Subsequently, it is also shown how to derive a challenge response protocol for verifying the storage correctness as well as identifying misbehaving servers.

VI. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

Our security analysis focuses on the adversary model as defined. We also evaluate the efficiency of our scheme via implementation of both file distribution preparation and verification token

precipitation. In our scheme, servers are required to operate on specified rows in each correctness, verification for the calculation of requested token. We will show that this "sampling" strategy on selected rows instead of all can greatly reduce the computational overhead on the server, while maintaining the detection of the data corruption with high probability. Suppose nc servers are misbehaving due to the possible compromise or Byzantine failure. In the following analysis, we do not limit the value of nc, i.e., nc ≤ n. Assume the adversary modifies the data blocks in z rows out of the l rows in the encoded file matrix. Let r be the number of different rows for which the user asks for check in a challenge. Let X be a discrete random variable that is defined to be the number of rows chosen by the user that matches the rows modified by the adversary.

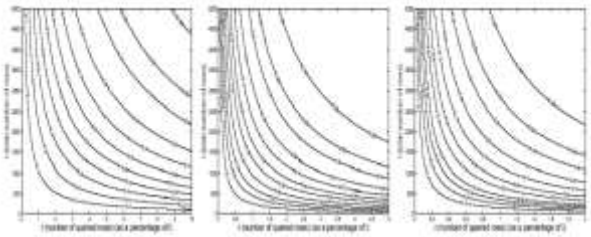

Fig 4: performance analysis results

VII. Experimental Results

Fig 5: Login

Fig 6: Authorized User

Fig7: Stroage Verification

Fig 8: Data Verification

VII. CONCLUSION

In this paper, we investigated the problem of data security in cloud data depot, which is essentially a distributed storage system. To assure the correctness of users' data in cloud data depot, we proposed 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. By utilizing the homomorphic token with distributed verification of erasurecoded data, our scheme achieves the integration of storage correctness insurance and data error localization, Through detailed security and performance analysis, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

VIII. REFERENCES

[1] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws.amazon.com, 2008.

[2] N. Gohring, "Amazon's S3 down for several hours," Online at http://www.pcworld.com/businesscenter/article/ /142549/amazons s3 down for several hours.html, 2008
[3] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.
[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc.of Asiacrypt '08, Dec. 2008.
[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175,2008, http://eprint.iacr.org/.
[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007.
[7] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm '08, pp. 1–10, 2008.
[8] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc.of ICDCS '06, pp. 12–12, 2006.
[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard,"A Cooperative Internet Backup Scheme," Proc. of the 2003 USENIX Annual Technical Conference (General Track), pp. 29–41, 2003.
[10] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.
[11] L. Carter and M. Wegman, "Universal Hash Functions," Journal of Computer and System Sciences, vol. 18, no. 2, pp. 143–154, 1979.
[12] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasurecoded Data," Proc. 26th ACM Symposium on Principles of Distributed Computing, pp. 139–146, 2007.
[13] J. S. Plank and Y. Ding, "Note: Correction to the 1997 Tutorial on Reed-Solomon Coding," University of Tennessee, Tech. Rep. CS-03-504, 2003.
[14] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. of IEEE INFOCOM, 2009.
[15] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. of ICDCS '08, pp. 411–420,2008.
[16] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.
[17] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1–6, 2007.