Verifying Data Integrity Using PDP Technique in Multi-Cloud Storage

Sachin S Lakde
PG Department of Computer Science and Engineering,
JD College of Engineering and Management
RTM Nagpur University, Maharashtra, India
sachinlakde@gmail.com

Mirza M Baig
Assistant Professor
PG Department of Computer Science and Engineering,
JD College of Engineering and Management
RTM Nagpur University, Maharashtra, India
mirzammb@gmail.com

***Abstract*—*In this paper we are going to say an efficient technique that describes how integrity is maintained in storage of the data. we propose a cooperative provable data possession scheme in hybrid clouds to support scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data.*

Keywords*-*PDP,multi-cloud,integrity verification,Index hierachy.

*****

I. INTRODUCTION

In cloud computing, one of the core design principles is dynamic scalability, which guarantees cloud storage service to handle growing amounts of application data in a flexible manner or to be readily enlarged. Parallel computing can be implemented in several ways of computing like instruction level, bit level, task and data parallelism. Based on the level at which hardware supports parallelism, it can be classified as multi-core and multi-processor. Simply, Provable Data possession (PDP) is a technique for validating data integrity over remote servers formalized a PDP model. In that model, the data owner pre-processes the data file to generate some metadata that will be used later for verification purposes through a challenge response protocol with the remote/cloud server. The file is then sent to be stored on an untrusted server, and the owner may delete the local copy of the file. Later, the server demonstrates that the data file has not been deleted or tampered with by responding to challenges sent from the verifier who can be the original data owner or other trusted entity that shares some information with the owner. Researchers have proposed different variations of PDP schemes under different cryptographic assumptions.We will present various PDP schemes in the literature survey section, and will elaborate how they vary from different perspectives.

- Integrity Entity: It concerns with concept of Primary keys.
- Integrity Referential: It concerns the concept foreign keys.
- Integrity Domain: It specifies that all the columns in database must be declared upon a defined domain.

II. RELATED WORK

Privacy preservation and data integrity are two of the most critical security issues related to user data [4]. In conventional paradigm, the organizations had the physical possession of their data, and thus have an ease of implementing better data security policies. But in case of cloud computing, the data is stored on an autonomous business party, that provides data storage as a subscription service. The users have to trust the *service provider* (*SP*) with security of their data. In [7], the author discussed the criticality of the privacy issues in cloud computing, and pointed out that an information is obtaining.

A) Multi-Cloud Framework Verification

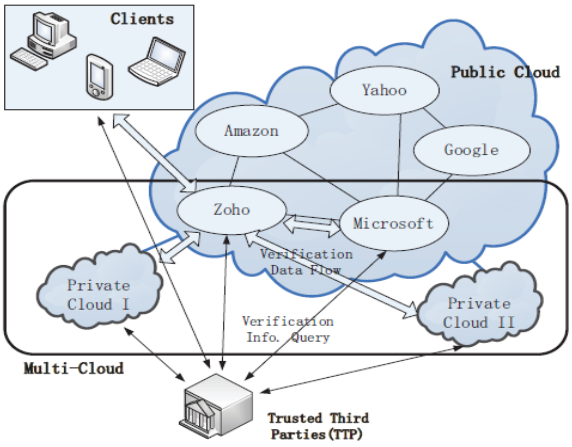
Multi-cloud technique is the use of two or more cloud services to minimize the risk of large amount of data loss or temporary fault in the computers due to a localized component failure in a cloud computing environment. Such a failure may occur in hardware, software, or infrastructure. A multi-cloud approach is also used to control the traffic from different customer bases or partners through the fastest possible parts of the network. Some clouds are better suited than others for a particular task.

III. STRUCTURE AND TECHNIQUE

In a world that sees new technological trends bloom and fade on almost a daily basis, one new trend promises for a long time. This trend is called cloud computing. Cloud is a collection of computers and servers that are publically accessible via the internet. This hardware is typically owned and operated by a third party on a consolidated basis in one or more data center locations. There are many advantages of using cloud technology. In order to prove the integrity of data stored in a multi-cloud environment, we define a framework for CPDP based on interactive proof system (IPS) and multi-prover zero-knowledge proof system (MPZKPS) We make use of this simple hierarchy to organize data blocks from multiple CSP services into a large size file by shading their differences among these cloud storage systems. The resources in Express Layer are split and stored into three CSPs, that are indicated by different colors, in Service Layer. In turn, each CSP fragments and

 ISSN: 2321-8169
1172– 1174

stores the assigned data into the storage servers in Storage Layer. We also make use of colors to distinguish different CSPs. Moreover, we follow the logical order of the data blocks to organize the Storage Layer. This architecture also provides special functions for data storage and management.



A. *Verifiable Homomorphic Technique*

A homomorphism is a map $f : \mathbb{P} \rightarrow \mathbb{Q}$ between two groups such that $f(g1 \oplus g2) = f(g1) \otimes f(g2)$ for all $g1$, $g2 \in \mathbb{P}$, where $\oplus$ denotes the operation in $\mathbb{P}$ and $\otimes$ denotes the operation in $\mathbb{Q}$. This notation has been used to define Homomorphic Verifiable Tags (HVTs) in [2]: Given two values σi and σj for two messages mi and mj , anyone can combine them into a value σ' corresponding to the sum of the messages $mi + mj$

B. *Hash Index Hierachy Technique*

An architecture for data storage in hybrid clouds. This architecture is based on a hierarchical structure with three layers to represent the relationship among all blocks for stored resources. Three layers can be described as follows:

- Layer1 (Express Layer): offers an abstract representation of the stored resources;
- Layer2 (Service Layer): promptly offers and manages cloud storage services; and
- Layer3 (Storage Layer): directly realizes data storage

C. *CPDP Model*

KeyGen(1_): takes a security parameter _ as input, and returns a secret key sk or a public-secret keypair (pk, sk);

TagGen(sk,F,P): takes a secret key sk, a file F, and a set of CSPs P = {Pk}, and returns the triples (_, , _), where _ is the secret of tags, = (u,$\mathcal{H}$) is a set of verification parameters u and an index hierarchy $\mathcal{H}$ for F, _ = {_(k)}Pk∈P denotes a set of all tags, where _(k) is the tags of the fraction F(k) of F in Pk;

Proof(P, V): is a protocol for a proof of data possession between CSPs P = {Pk} and a verifier V. At the end of the protocol, V returns a bit {0|1} denoting a binary decision for either false or true.

In this work, to mitigate the threats facing cloud storage, we extended the cloud data storage to include multiple service providers, where each cloud storage represents a different service provider. Our motivation behind such an extension is that, the adversary, similar to any other cloud user, is abstracted from the actual clouds of servers implemented by different cloud service provider..

D. *Error Recovery*

Once the data corruption is detected, next important step is to recover the corrupted data and bring data storage back to consistent state. The comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server. Therefore user can recover the corrupted data. Our system recovers data from backup server & distributes all data vectors to corresponding servers. This will results in successful recovery of corrupted data. But due to file splitting we made at the time of file distribution, user‟s need to recover file from all the servers. Error localization is limited to misbehaving servers only, i.e. servers giving false assurance of posing user‟s data.

In this section we describe the setup for the linear programming assignment problem (LP-Assignment) that describes our proposed model. Each cloud customer is provided with *p* cloud service providers, where each of them offers a *QoS* level for storage services and required a cost *C* to be paid by the customer per storage unit of data. Previous studies in [16] [17] proposed a dividing scheme for user's data in such a way that, the user will divide his data into *N* data pieces where at-least *k* data pieces out of *N* data pieces are required to recover any meaningful information of the data. In addition to this (*k;N*) *threshold*, we propose another *threshold* of (*q; p*); which states that, at least *q* number of cloud service providers out of *p* number of cloud service providers must take part in retrieving

IV. CONCLUSION

We have analyzed the data security concerns in cloud data storage, which is a distributed storage system. We proposed a distributed scheme to ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud. To provide redundancy we used erasure correcting code in the file distribution preparation. As we all know cloud is not just a third party data warehouse. So providing support for dynamic operations is very important. Our scheme maintains the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud. Challenge response protocol along with pre-computed token is used to verify the storage correctness of user‟s data & to effectively locate the malfunctioning server when data corruption has been detected. Through detailed performance analysis, we show that our scheme is having very low communication overhead & guarantees to detect every single unauthorized data modification. Our scheme has no limitation on number of pre-computed tokens

 ISSN: 2321-8169
1172– 1174

used for challenging the cloud servers. Unlimited number of challenges can be made. We removed burden of calculating pre-computed tokens & storing the locally from the users. By splitting the file according to the number of server"s we are added extra security to system. But we still believe that data storage security in Cloud computing is an area full of challenges and of paramount importance.

V. REFERENCES

[1] P. Mell, T. Grance, "Draft NIST working definition of cloud computing", Referenced on June. 3rd, 2009, Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2009.

[2] Arvind D Meniya, Harkishan B Jethva " Single-Sign-On (SSO) across open cloud computing federation", Vol. 2, Issue 1,Jan-Feb 2012

[3] M. Arrington, "Gmail Disaster: Reports of mass email deletions",Onlineattp://www.techcrunch.com/2006/12/28/gmail-disasterreportsofmass-email-deletions/, December 2006.

[4] Amazon.com, "Amazon s3 availability event: July 20, 2008"Onlineathttp://status.aws.amazon.com/s320080720.html, 2008.

[5] B.Krebs,"PaymentProcessorBreachMayBeLargestEver",Onlineathttp: //voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html.

[6] S. P. Jaikar & M. V. Nimbalkar IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 1, Issue 6 (July-Aug. 2012), PP 43-49 "Securing Cloud Data Storage"

[7] Oualha, N., Onen, M., Roudier, Y.: A Security Protocol for Self-Organizing Data Storage.Tech. Rep. EURECOM +2399, Institut Eurecom, France (2008).

[8] Oualha, N., Roudier, Y.: A game theoretic model of a protocol for data possession verification. In: IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing (TSPUC'07). Helsinki, Finland (2007)

[9] A. Juels and B.S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th CM conference on Computer and communications security. [10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07.