Research Inventy: International Journal of Engineering And Science
Vol.4, Issue 5 (May 2014), PP 26-31
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Co-Operative Multiple Replica Provable Data Possession for Integrity Verification in Multi-Cloud Storage

Mr.Susheel George Joseph M.C.A, M,Tech, M.Phil(CS)
(Assistant Professor, Department of M.C.A, Kristu Jyoti College of Management and Technology, Changanassery,susheelgj@gmail.com)

ABSTRACT : *Many storage systems rely on replication to increase the availability and durability of data on untrusted storage systems. At present, such storage systems provide no strong evidence that multiple copies of the data are actually stored. Storage servers can collude to make it look like they are storing many copies of the data, whereas in reality they only store a single copy. We address this shortcoming through multiple-replica provable data possession (MR-PDP): A provably-secure scheme that allows a client that stores t replicas of a file in a storage system to verify through a challenge-response protocol that each unique replica can be produced at the time of the challenge and that the storage system uses t times the storage required to store a single replica. MR-PDP extends previous work on data possession proofs for a single copy of a file in a client/server storage system.*

KEYWORDS: *Availability, Data Possession, Homomorphic, Integrity, Multi-Cloud, Multi-Replica,Zero-Knowledge*

I. INTRODUCTION

This is a study that combines the advantages of both CPDP (Co operative PDP) and MRPDP (Multiple Replica PDP). The proposed paper manages user's data in multiple cloud storage by ensuring the integrity and availability of user's data. To describe this we should have to describe both MRPDP and CPDP.Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this scheme, we address the construction of an efficient PDP scheme for distributed cloud storage to support the 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. We present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

Using multiple-replica provable data possession(MR-PDP) to store t replicas is computationally much more efficient than using a single-replica PDP scheme to store t separate, unrelated files (e.g., by encrypting each file separately prior to storing it). Another advantage of MR-PDP is that it can generate further replicas on demand, at little expense, when some of the existing replicas fail. The generation of replicas is on demand by the user's request that is based on the security choice selected by the user at the time of file upload. The user can choose three options *Low*, *Medium*, *High* at the time of file upload. The uploaded file is divided in to N blocks of different sizes to achieve the efficiency in storage and is also used to improve security, here N represent the number of clouds we are using. Low means the file is divided into N blocks (here 3), and each block is stored in N different location of the single cloud. Medium means the file is divided into N blocks and each block is stored in N different clouds which improves the security of data but reduce the availability. High means the file is divided into N blocks and each N block is stored in N different clouds that are we are keeping the replicas of file in N different clouds. The system maintains a download count to dynamically create the replicas in accordance with the users demand.The system which consists of three users namely *User* who have the access rights to upload, download and delete file, *TPA (Third Party Auditor)* who verifies the files that are uploaded by the registered user and the user can download the file only after this verification, *Admin* who own the system and who have the full access right, can create or delete TPAs and can view the uploaded files and details about the uploads. A single cloud can have different TPA's and the work load is divided among by using the random function to select the corresponding files from the cloud.

The creation and deletion of TPA is based on the work load and efficiency of TPA which is monitored by the administrator. The data uploaded by the user is temporarily stored in an encrypted form by using the

homomorphic encryption algorithm. We can use any kind of encryption algorithms along with this applications but it is better to choose a zero knowledge proof algorithm. This uses an encryption key which is automatically supplied to the user at the time of file upload. The data is stored in cloud only after it is verified by TPA. The actual storage of data is in an encrypted form called *Meta Data*, which ensures additional security measure for the cloud data. The user gets the original file when he/she downloads the needed file from the cloud storage, which ensures the integrity of data. The user is unaware of the background processes. This system reduces the overload of admin by creating TPAs. The TPA can be of any number for each cloud depending on the number of clouds we are using.

II. NEED FOR THE SYSTEM

The main objective of this paper is to provide an insight to build a system at low-cost, scalable, location independent platform for managing clients' data, current cloud storage systems adopt several new distributed file systems, for example, Apache Hadoop Distribution File System (HDFS), Google File System (GFS), Amazon S3 File System, CloudStore etc. These file systems share some similar features: a single metadata server provides centralized management by a global namespace; files are split into blocks or chunks and stored on block servers; and the systems are comprised of interconnected clusters of block servers. Those features enable cloud service providers to store and process large amounts of data. However, it is crucial to offer an efficient verification on the integrity and availability of stored data for detecting faults and automatic recovery. Moreover, this verification is necessary to provide reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures.

Some of the Objectives can be summarized as

Usability aspect: A client should utilize the integrity check in the way of collaboration services. The scheme should conceal the details of the storage to reduce the burden on clients.

Security aspect: The scheme should provide adequate security features to resist some existing attacks, such as data leakage attack and tag forgery attack.

Performance aspect: The scheme should have the lower communication and computation overheads than non-cooperative solution.

III. EXISTING ARCHITECTURE

There exist various tools and technologies for multicloud, such as Platform VM Orchestrator, VMwarevSphere, and Ovirt. These tools help cloud providers construct a distributed cloud storage platform for managing clients' data. However, if such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. For example, the confidential data in an enterprise may be illegally accessed through a remote interface provided by a multi-cloud, or relevant data and archives may be lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. Therefore, it is indispensable for cloud service providers to provide security techniques for managing their storage services.More Moreover another limitations of the existing system is that it is not suitable for multicloud storage services.To check the availability and integrity of outsourced data in cloud storages, researchers have proposed two basic approaches called Provable Data Possession and Proofs of Retrievability. Ateniese et al. first proposed the PDP model for ensuring possession of files on untrusted storages and provided an RSA-based scheme for a static case that achieves the communication cost.

They also proposed a publicly verifiable version, which allows anyone, not just the owner, to challenge the server for data possession..They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption, but the servers can deceive the owners by using previous metadata or responses due to the lack of randomness in the challenges. The numbers of updates and challenges are limited and fixed in advance and users cannot perform block insertions anywhere.

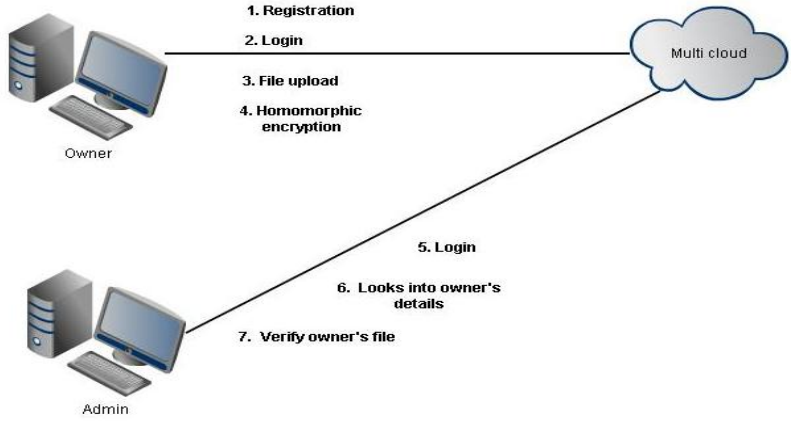

Fig 1. Existing Architecture

IV. PROPOSED ARCHITECTURE

Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the 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. We present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

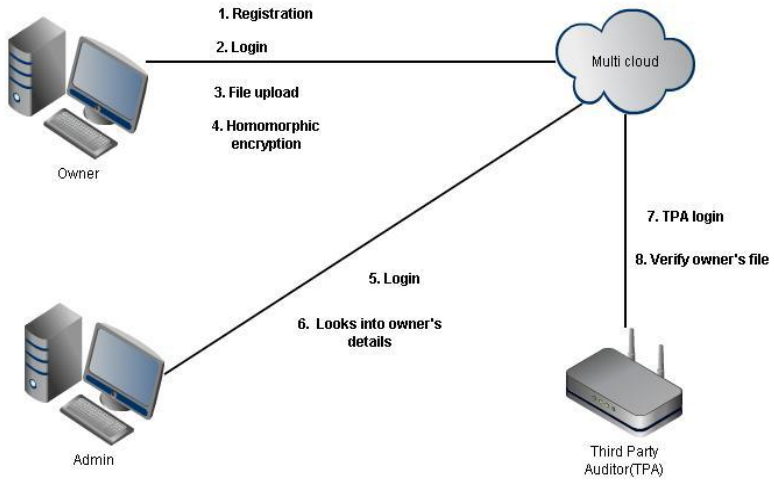

Fig 2. Proposed Architecture

V. ARCHITECTURE DETAILS – MODULARIZATION

The modules are formed on the basis of the functionalities that are found in the proposed system. The functionalities are performed by three users namely *Admin*, *TPA*, and registered *User*.

Cooperative accessing

We can access data from multiple clouds by using the CPDP scheme, it is based on homomorphic verifiable response (HVR) and hash index hierarchy (HIH). We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties.

Replica generation

The replica generation can be performed by the MR-PDP scheme. This scheme allows a client to stores t replicas of a file in a storage system to verify through a challenge-response protocol that each unique replica can be produced at the time of the challenge and that the storage system uses t times the storage required to store a single replica.

Integrity verification

Integrity requires that authorized changes must be detected and tracked, and changes must be limited to a specific scope. Due to the increased number of entities and access points in a cloud environment, authorization is crucial in assuring that only authorized entities can interact with data. A cloud computing provider is trusted to maintain data integrity and accuracy. To verify integrity, we must examine the net effects on the cloud data related to data integrity. There will be a set of standards for monitoring the integrity of data. Integrity monitoring is essential in cloud storage as data integrity is critical for any data centre. Here, we consider the existence of multiple CSPs to collaboratively store and maintain the clients' data. The traditional cryptographic primitives for data integrity and availability based on hashing and signature schemes are not applicable on the outsourced data. Hence lots of new techniques have been found out for integrity verification. This technique too doesn't guarantee the expected security in clouds. Here we use the CPDP scheme for integrity verification.

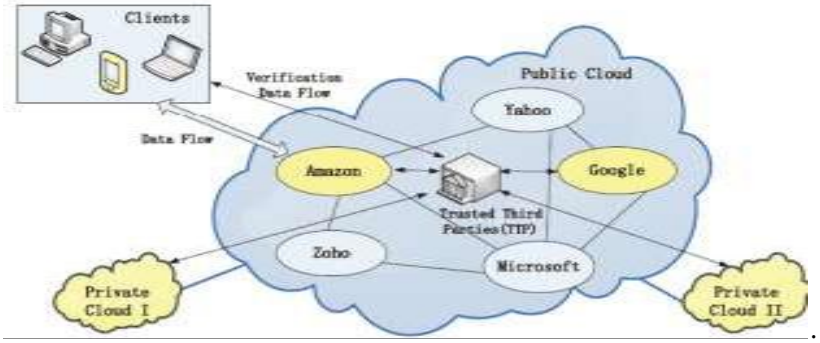

Fig 3 .Integrity Verification in Hybrid Cloud

File Division

The Cloud User who has a large amount of data to be stored in multiple clouds and has the permissions to access and manipulate stored data. The User's Data is converted into data blocks of different sizes for improving the efficiency of storage and as well as to improve the security of file.

Registration

The user can store the file into the cloud storage only if he/she is a registered owner of this web application. The registration can be made as either free or a paid registration depending on the organization's requirement.

File Upload

Not all files are directly stored in multiple clouds, but only the files that are verified by the trusted TPA are uploaded. If any corrupted file is loaded, then that file cannot be saved instead they may be deleted by the TPA. The File may be encrypted using the cryptographic key which is randomly generated.

File verification

Using the cryptographic key the file is encrypted and by using this key the file contents may be decrypted by the TPA for the verification process. He can view the file contents by the decryption process and can verify the files.

File download

Only the verified Files can be downloaded by the File Owner. If the user wants to download their files, the data stored in multi-cloud is integrated and downloaded. Here the downloading count is saved for generating the replicas of the file by ensuring the demand of file. The user could get acknowledgement about the server which the file is saved, it is also an additional provision which is depends on the client's requirement. We can increase the security of downloading by verifying the key along with downloading process. One problem can arise is in the case of key remembrance. But this can be solved if we are using *Grid Security* along with it. It can be considered as an optional module along with file downloading and deletion modules.

File Deletion

The Uploaded file can be deleted by the File Owner. The security can be increased if we are making key verification along with the deletion process. One problem can arise is in the case of key remembrance. But this can be solved if we are using *Grid Security* along with it. It can be considered as an optional module along with file downloading and deletion.

File Storage

Verified File is stored in the cloud in an Encrypted format using the Cryptographic key. So file security is ensured and no one can decrypt or hack the file. The Verified File can be stored in three Clouds. The uploaded file is split in to different blocks based on the number of clouds we are using. When user uploads the file there is a choice of file security-Low, Medium, High. If the security is 'Low' then it is stored in Cloud1 and it is of 'Medium' then the encrypted file blocks are saved in each cloud individually. If the security option is 'High' then the same file blocks are saved in each Clouds independently.

TPA Creation and deletion

TPA is one of the users in this application. TPA is used to verify the files that are uploaded by the *User*. The *User* file is uploaded to the cloud storage by the TPA only after the verification process. TPA can view the file content without downloading; he can decrypt the information by using the corresponding encrypted key. TPA creation is done by the Administrator for reducing the overhead in managing each cloud user. The deletion task can be used if the TPA is no longer needed for the particular cloud.

View All Files

All the Files in the web including verified and not-verified are viewed by the Administrator

View File Owners

Registered File Owners are viewed by the Administrator. Admin can have the facility to contact the file owners and can monitor the storage space used by the file owners.

VI. CONCLUSION

The aim of this article entitled "*Co-Operative Provable Data Possession For Integrity Verification In Multi-Cloud Storage*" was to demonstrate the combined effect of two efficient and well known concepts CPDP and MR-PDP. This could achieve the user's requirements in the multi cloud environment such as availability and security of data. These are the two major concerns in a distributed environment.As we are using multi cloud, so there are multiple cloud service provider's for multiple clouds. As we want to store block in each cloud so the request has to go from each Cloud Service Provider, so to reduce the complexity we can use the Centralized Cloud Service Provider. Therefore, every request is managed by centralized Cloud Service Provider. This research can be treated as a new technique for data integrity verification in data possession. As part of future enhancement, I would like extend my work to explore more effective MR-CPDP constructions. Finally, it is still a challenging problem for the generation of tags with the length irrelevant to the size of data blocks and various file formats.

VII. ACKNOWLEDGEMENT

I would like to thank GOD Almighty for the valuable support throughout this research phase to complete this paper successfully. I would like to express my sincere thanks to my Guide for his valuable suggestions. I also like to thank my family members and my college authorities as well as co-workers for their consistence support and motivation.

REFERENCES

[1] O. Rahamathunisa Begam,, T. Manjula,, T. Bharath Manohar,, B. Susrutha, "Cooperative Schedule Data Possession for Integrity Verification in Multi-Cloud Storage" in International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue. 5, Sep - Oct. 2013, pp.2726-2741.

[2] Yan Zhu ,Beijing Key Lab. of Internet Security Technol., Peking Univ., Beijing, China; Hongxin Hu ; Gail-Joon Ahn ; Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage" in Parallel and Distributed Systems, IEEE Transactions on (Volume:23 , Issue: 12), Dec. 2012,pp.2231 – 2244.

[3] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, "Provable data possession at untrusted stores," in ACM Conference on Computer and Communications Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598–609.

[4] 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 computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb 2009.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology (CRYPTO'2001), vol. 2139 of LNCS, 2001, pp. 213–229.

[6] Megha Patil , Prof. G.R.Rao, "Integrity Verification in Multi-Cloud Storage Using Cooperative Provable Data Possession " in Megha Patil et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, pp.982-985

[7] Vaibhav Bharati and M R Patil., "Advanced Cooperative Provable Data Possession based Data Integrity Verification for Multi-Cloud Storage" in International Journal of Computer Applications, Foundation of Computer Science, New York, USA, ,November 2013,pp.25-28

[8] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, "Provable data possession at untrusted stores," in ACM Conference on Computer and Communications Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598–609.

[9] A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for Large files," in ACM Conference on Computer and Communications Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 584–597.

[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th international conference on Security and privacy in Communication networks, SecureComm, 2008, pp. 1–10.
B. Sotomayor, R. S. Montero, I. M. Llorente, and I. T. Foster,"Virtual infrastructure management in private and hybrid Clouds," IEEE Internet Computing, vol. 13, no. 5,2009, pp. 14–22.

[11] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 187–198.