Volume 5, Issue 9, September 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Proxy Based Provable Data Integrity in Multi-Cloud Using ID-DPDP

[1]Raja Rajeswari. Kandimalla, [2]Dr. K. Suresh Babu
[1](M.Tech), [2]Professor,
[1, 2] Department of CSE, Vasireddy Venkatadri Institute of Technology (VVIT)
Andhra Pradesh, India

Abstract: ***In cloud computing systems, data owners usually store huge volume of data on the cloud servers, thus clients may access the data from Cloud servers without knowing their locations. In this connection outsourcing client data among untrusted cloud servers, reliable verification, efficient data outsourcing and system performance is a challenging issue .In order to address the above issues we use Centralized Cloud Service Provider to improve the System Performance by reducing the time complexity. Therefore, every Client request is managed by centralized Cloud Service Provider. In order to provide the reliable verification during uploading and downloading User has to answer the Security Questions. Security Questions and Answers are provided by user during the registration phase. So during Uploading/Downloading operation, if user is normal then he can answer that security questions, if he/she is intruder then he/she cannot answer that questions. Thus, using this we can provide more Security. Also, we can provide the Security to uploaded data and the digest by using the encryption algorithm thus we can achieve efficient data out sourcing with data integrity. Furthermore, the integrity test protocol must be efficient in order to save the verifier's cost.***

Keywords: Provable data possession, proofs of retrievability, ID-DPDP system.

I. INTRODUCTION

Data storage on cloud is one of the well known services offered by cloud computing. Because of this service subscribers do not have to store their own data on local servers, where instead their data will be stored on the cloud service provider's servers. Cloud storage makes it possible for users to remotely store their data and enjoy the on demand high quality cloud applications without the any burden of local hardware and software management. While making clients free from data storage burdens, cloud brings new and severe security threats in user's outsourced data. The critical issue of data integrity comes whenever client uploads data on un-trustworthy servers. In such scenarios, clients need to implement strategies to prove originality of data. The client may need to access whole file to ensure data integrity, which is time and space consuming [4]. Considering the huge size of the outsourced data and the users constrained resource it is not always possible to access complete data, which boast an array of advantages like unlimited storage capability, anywhere accessibility etc. Since Cloud computing environment is constructed on open architectures and interfaces, it has the potential to incorporate multiple internal and/or external cloud services together to provide high interoperability. This type of distributed cloud environment is called as a Multi-cloud. The proverb of not putting all your eggs in one basket applies in Multi-cloud too.

A Multi-cloud approach is one where an enterprise uses two or more cloud services, therefore reducing the risk of widespread data loss or outage due to a component failure in a single cloud computing environment. Frequently, by using virtual infrastructure management (VIM) [1], a multi-cloud allows clients to easily access his/her resources remotely through interfaces such as web services provided by Amazon EC2. There exist various tools and technologies for multi-cloud, such as VMware vSphere, Platform VM Orchestrator and Ovirt. These tools help cloud providers to create a distributed cloud storage platform (DCSP) for managing clients' data. But, if such an important platform is vulnerable to security attacks, it would bring irrevocable losses to the clients. For example, the secret data in an enterprise may be illegally accessed by using remote interfaces, or organization relevant data and archives are lost or tampered with when they are stored into an uncertain storage pool outside the enterprise.

One of the biggest issues with cloud data storage is that of data integrity verification at untrusted servers. Also, there exist various motivations like maintaining reputation for cloud service providers (CSP) to behave unfaithfully towards the cloud users. For example, the cloud service provider (CSP), which experiences Byzantine failures infrequently, may decide to hide the data errors from the clients for the benefit of their own like for maintaining their reputation or for saving money and storage space. The service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Therefore, it is crucial for cloud service providers (CSPs) to provide security

techniques for managing their storage services. Provable data possession (PDP) [2] (or proofs of retrievability (POR) [3]) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. Checking proof without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Consequently, it is able to replace traditional hash and signature functions in storage outsourcing. Different PDP schemes have been recently proposed, such as Scalable PDP [4] and Dynamic PDP [5]. However, these schemes mainly focus on PDP issues at untrusted servers in a single cloud storage provider and are not suitable for a multi-cloud environment.

II. PHASES OF SECURITY RISK IN MULTICLOUD

From different cloud service models, the security responsibility between cloud users and cloud service providers is different. In different cloud environment addresses security control in relation to physical, environmental, and virtualization security. Whereas, the users remain responsible for addressing security control of the IT system including the operating systems, applications and data according to Tabakiet al. [9], the way the responsibility for privacy and security in a cloud computing environment is shared between cloud users and cloud service providers differs between delivery models.

In SaaS, cloud service providers are more responsible for the security and privacy of application services than the cloud users. This responsibility is more relevant to the public than the private cloud environment because the clients need stricter security requirements in the public cloud. With PaaS, users are responsible for taking care of the applications that they build and run on the platform, while cloud service providers are responsible for protecting one user's applications from others.

In IaaS, users are responsible for protecting operating systems and applications, whereas cloud service providers must provide protection for the users' data [9]. Ristenpartet al. [10] claims that the levels of security issues in IaaS are different. The impact of security issues in the public cloud is greater than the impact of the private cloud. For instance, any damage which occurs to the security of the physical infrastructure or any failure in relation to the management of the security of the infrastructure will cause many problems. In the cloud environment, the physical infrastructure that is responsible for data processing and data storage can be affected by a security risk.

Confidentiality: Confidential is term in which cloud service provider also unknown to cloud users data which is uploaded on his own cloud, the cloud storage provider does not learn any information about customer data.

Integrity: Any unauthorized or illegal modification and updating the contents of client data from the cloud storage provider can be detected by the customer while retaining the main benefits of a public storage service.

Availability: Data of cloud user is available to the user at anytime, anywhere, anyplace from the cloud server. Customer data is accessible from any machine and at all-time.

Reliability: Customer data is reliably backed up.

Efficient Retrieval: Data retrieval times are comparable to a public cloud storage service.

Data Sharing: Cloud users can share data securely with trusted parties.

III. ID-DPDP SYSTEM MODEL AND SECURITY DEFINITION PRESENTED SYSTEM

3.1 Presented System

The ID-DPDP system model and security definition are presented in this section. An IDDPDP protocol comprises four different entities which are illustrated in Figure 1. We describe them below:

1) Client: An entity, which has massive data to be stored on the multi-cloud for maintenance and computation, can be either individual consumer or corporation.

2) CS (Cloud Server): An entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients' data.

3) Combiner: An entity, which receives the storage request and distributes the block-tag pairs to the corresponding cloud servers. When receiving the challenge, it splits the challenge and distributes them to the different cloud servers. When receiving the responses from the cloud servers, it combines them and sends the combined response to the verifier.

4) PKG (Private Key Generator): An entity, when receiving the identity, it outputs the corresponding private key.

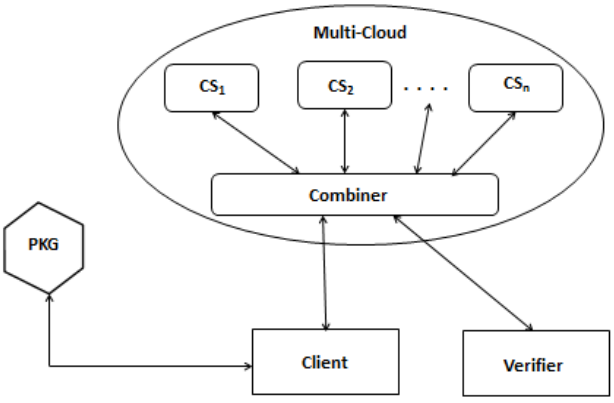

Fig 1. Presented ID-DPDP system model

3.2 Proposed System:

In order to address the issues of presented system, we use Centralized Cloud Service Provider to improve the System Performance by reducing the time complexity .Therefore, every Client request is managed by centralized Cloud Service Provider. In order to provide the reliable verification during uploading and downloading User has to answer the Security Questions. Security Questions and Answers are provided by user during the registration phase. So during Uploading/Downloading operation, if user is normal then he can answer that security questions, if he/she is intruder then he/she cannot answer that questions. Thus, using this we can provide more Security. Also, we can provide the Security to uploaded data and the digest by using the encryption algorithm thus we can achieve efficient data out sourcing with data integrity. Furthermore, the integrity test protocol must be efficient in order to save the verifier's cost.

System Functions:

1) PKG (Private Key Generator). Entity, trusted by the clients and the PCSs, that generates the public parameters Params, the master public key mpk, the master secret key msk and the private key of the Client which helps to protect user privacy as well provide data integrity .

2) Client: Entity which has massive data to be stored on the public cloud for maintenance and computation. Clients can be either individual consumers or group consumers, e.g., the departments of the company in the motivated scenario.

3) Cloud Server: Entity, managed by the cloud service provider that has significant storage space and computational resources to maintain the clients' data. In the cloud paradigm, by putting the large data files on the remote cloud servers, the clients can be relieved of the burden of storage and computation. As the clients no longer possess their data locally, it is of critical importance for them to ensure that their data are being correctly stored and maintained. That is, clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies.

4) Centralized CSP: To reduce the complexity we can use the Centralized Cloud Service Provider. Therefore, every request is managed by centralized Cloud Service Provider in order to reduce the time complexity thus to improve the system performance. Here every client outsource data will managed by Centralized CSP in secured manner data will not reviled at Centralized CSP Level. It will distribute Encrypted data over Multiple Cloud severs as Network code based (spitted data among servers) manner. Hence it helps data availability and security.

5) User: An Entity which can access the data from multi-cloud, where it has to take the acceptance from PKG to view files.

6) Verifier: An Entity periodically checks for data integrity.

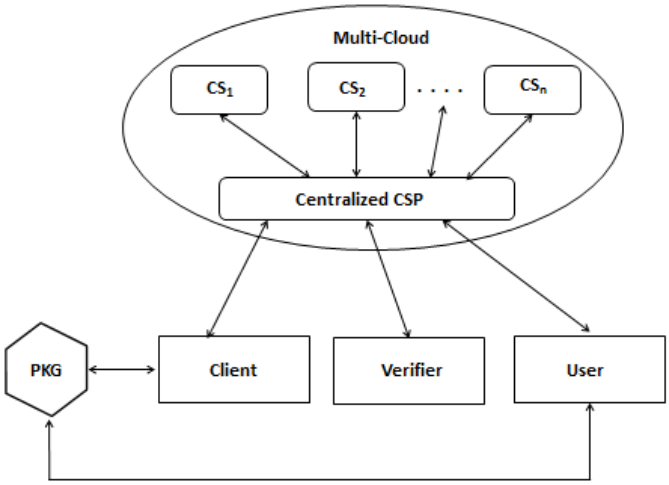

Fig 2. Proposed System

IV. PERFORMANCE ANALYSIS

We give elaborate performance analysis in order to show the efficiency of the proposed scheme. In our experiment, the process of the user, the server and the TPA are implemented on a windows 7 system with an Intel i5 CPU running at 2.53 GHz, 2 GB DDR 3 of RAM(1.74 GB available). All algorithms are implemented by C language, and our code uses the MIRACL library version 5.6.1. The elliptic curve we used is a MNT curve, where the base field size is 159 bits and the embedding degree is 6. The security level is chosen to be 80 bits, it means that | vi | = 80 and | p | = 160. All the results of experiment are represented the average of 30 trials.

In the following, we emphasize on reporting our performance results from computational overhead, and we also give a performance comparison. According to the comparison, we can see that our scheme retains the efficiency, while fixing its security flaw.

The Proposed Scheme's Computation Overhead

Comparing, our scheme just additionally calculates an inverse element in cloud server side. The operation of computing inverse element is too small to be ignored. Therefore, in a practical system, our scheme is of high efficiency.

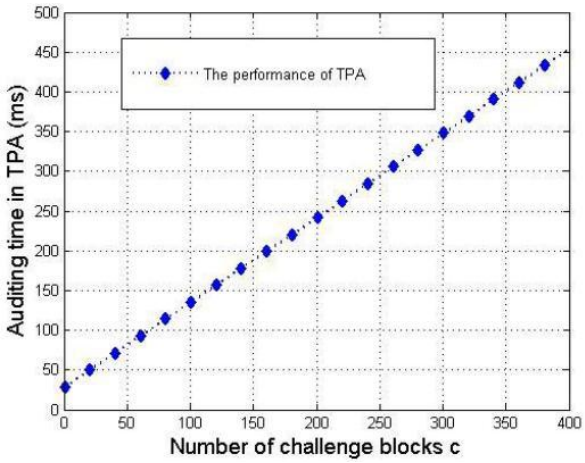

Fig. 3 Shows the performance of TPA in different challenge blocks c .'

Batch Auditing Overhead with Its Advantage

Fig. 4 and Fig. 5 respectively indicate the efficiency comparison on auditing time between batch auditing and basic auditing in c = 300 and c = 500 . The experiment shows that batch auditing improves efficiency a lot.

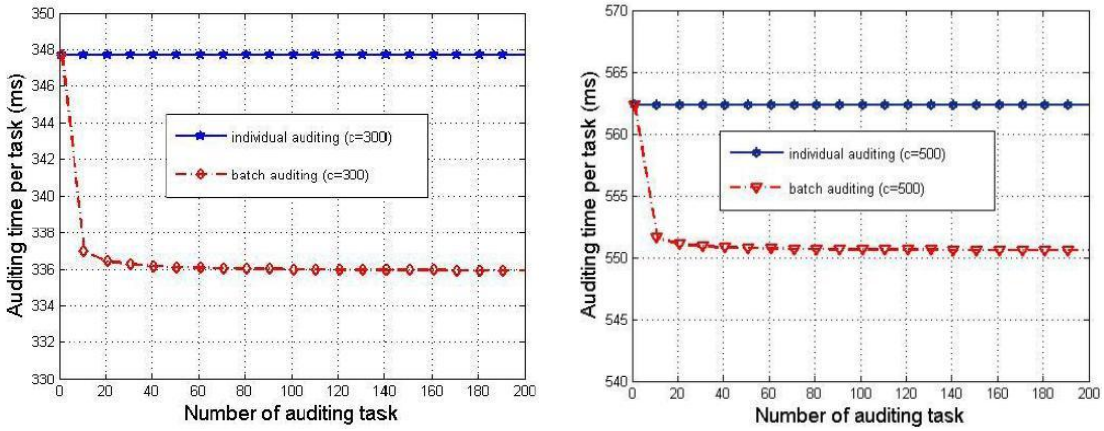

Fig 4 & 5 Comparison on auditing time between batch and individual auditing (c=500)

Through above formal security proof and performance analysis, we can see that our scheme fixes the aforementioned security flaw while ensuring the same efficiency. Therefore, the proposed scheme has advantages over the existing in a practical application.

V. CONCLUSION AND FUTURE WORK

This paper address various challenging issues which are related to access controlling, data integrity, data availability, security of data and system performance with respect to multicloud data storage and sharing by the clients. These are the 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.

REFERENCES

[1] 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, pp. 14–22,2009.

[2] 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.

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

[4] 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 netowrks, SecureComm, 2008, pp. 1–10.

[5] C. C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213–222.

[6] H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5350. Springer, 2008, pp. 90–107.

[7] Q. Wang, C.Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in ESORICS, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355–370.

[8] Q. Wang, C.Wang, J. Li, K. Ren, and W. Lou, "Enabling public Verifiability and data dynamics for storage security in cloud Computing," in ESORICS, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355–370.

[9] H. Tabakiet, J.B.D. Joshi and G.-J. Ahn, "Security and Privacy Challenges in Cloud Computing Environments", IEEE Security & Privacy, 8(6), 2010, pp. 24-31.

[10] T. Ristenpart, E. Tromer, H. Shacham and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds", CCS'09: Proc. 16thACM Conf. on Computer and communications security, 2009, pp. 199-212.

[11] Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 109–127.

[12] L. Fortnow, J. Rompel, and M. Sipser, "On the power of multiprover Interactive protocols," in Theoretical Computer Science, 1988, pp.156–161.

[13] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced Storages in clouds," in SAC, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.

[14] 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.

