



**RESEARCH ARTICLE**

# **Identity-Based Integrity Verification using PDP in Multi Cloud Storage**

**P.Janani Chandra**, M.E(SCHOLAR), Department of Computer Science and Engineering, N.P.R.C.E.T

Under the guidance of **Mrs. N.Dhanalakshmi**, Head of IT Department, N.P.R.C.E.T

*Abstract: Cloud computing has become an important thing in computer field. Cloud computing takes information processing as a service, such as storage and computing. Data integrity is important thing in cloud storage. In certain situations, clients should store their data such as image or text in multi cloud. When the client stores his/her data on multi-cloud servers, the distributed storage and integrity checking is very important. Here we propose an Identity Based Distributed Provable Data Possession (ID-DPDP) protocol for multi-cloud storage. Remote data integrity checking is important in cloud storage. It can make the clients verify whether their data is kept as it is without downloading the entire data. In some application scenarios, the clients have to store their data on multi-cloud servers. At the same time, the integrity checking protocol must be efficient in order to save the verifier's cost.*

*Keywords: Cloud computing, integrity, multcloud, data possession*

---

## **INTRODUCTION**

A protocol (ID-DPDP-Identity-based distributed provable data possession) is proposed to store data in multi cloud .ID-DPDP protocol eliminate the certificate management. In this system, the client's data is distributed to multi cloud servers based on

type of the data and size of the data. Private key generator generates the private key for the client, it contains the client unique id. Client's data is transferred to combiner; the combiner distributes the data according to the size and type of data. Verifier sends the challenges to the combiner, the combiner transfer the challenge to the respected cloud. Afterwards, combiner aggregates the result and check whether it is valid or not. If it is valid, allow clients to store the data in multi cloud. In the phase Extract, PKG creates the private key for the client. The client creates the block-tag pair and uploads it to combiner. The combiner distributes the block-tag pairs to the different cloud servers according to the storage metadata. The verifier sends the challenge to combiner and the combiner distributes the challenge query to the corresponding cloud servers according to the storage metadata. The cloud servers respond the challenge and the combiner aggregates these responses from the cloud servers. The combiner sends the aggregated response to the verifier. Finally, the verifier checks whether the aggregated response is valid. The concrete ID-DPDP construction mainly comes from the signature, provable data possession and distributed computing. The signature relates the client's identity with his private key. Distributed computing is used to store the client's data on multi-cloud servers. At the same time, distributed computing is also used to combine the multi-cloud servers' responses to respond the verifier's challenge. Based on the provable data possession protocol, the ID-DPDP protocol is constructed by making use of the signature and distributed computing. This paper is organised as follows: I.Scope, II.RelatedWork, III.System Model, IV.Contributions, V.Conclusions, VI.Future Work.

## **I.SCOPE**

PDP is a technique for validating data integrity over remote servers. In a typical PDP model, the data owner generates some metadata/information for a data file to be used later for verification purposes through a challenge response protocol with the remote/cloud server. The owner sends the file to be stored on a remote server which may be untrusted, and deletes the local copy of the file. As a proof that the server is still possessing the data file in its original form, it correctly computes a response to a challenge vector sent from a 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; One of the core design principles of outsourcing data is to provide dynamic scalability of data for various applications. This means that the remotely stored data can be not only accessed by the authorized users, but also updated and scaled by

the data owner. PDP schemes presented and focus on static or warehoused data and do not consider the case of dynamic data that are usually more prevailing in practical applications. Dynamic provable data possession (DPDP) constructions reported in the literature focus on the provable possession of a single copy of a dynamic data file. Although PDP schemes have been presented for multiple copies of static data, PDP scheme exists for multiple copies of dynamic data.

## II.RELATED WORK

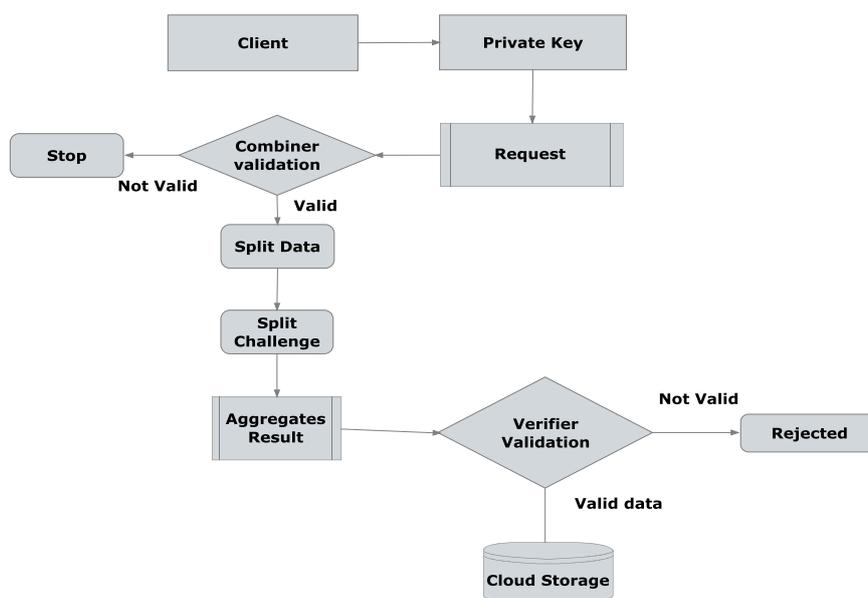
RDPC allows a client that has stored data at a public cloud server (PCS) to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof.

The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. In order to achieve secure RDPC implementations, Ateniese *et al.* proposed a provable data possession (PDP) paradigm [1] and designed two provably-secure PDP schemes based on the difficulty of large integer factoring. They refined the original paradigm and proposed a dynamic PDP scheme in [2] but their proposal does not support the insert operation. In order to solve this problem, Erway *et al.* proposed a full-dynamic PDP scheme by employing an authenticated flip table [3]. Following Ateniese *et al.*'s pioneering work, researchers devoted great efforts to RDPC with extended models and new protocols [4], [5], [6], [7], [8], [9], [10], [11], [12]. One of the variations is the proof of retrievability (POR), in which a data storage server cannot only prove to a verifier that he is actually storing all of a client's data, but also it can prove that the users can retrieve them at any time. This is stronger than the regular PDP notion. Shacham presented the first POR schemes [15] with provable security. The state of the art can be found in [16], [17], [18], [19] but few POR protocols are more efficient than their PDP counterparts. The challenge is to build POR systems that are both efficient and provably secure [14]. Note that one of benefits of cloud storage is to enable universal data access with independent geographical locations. This implies that the end devices may be mobile and limited in computation and storage. Regular RDPC protocols are more suitable for cloud users equipped with mobile end devices. Our ID-RDPC architecture and protocol are based on the PDP model.

### III.SYSTEM MODEL

The ID-DPDP system model and security definition are presented in this section. An ID-DPDP 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 System model of ID-DPDP Protocol**

### IV.CONTRIBUTIONS

In identity-based public key cryptography, this paper focuses on distributed provable data possession in multi-cloud storage. The protocol can be made efficient by eliminating the certificate management. We propose the new remote data integrity checking model: ID-DPDP. The system model and security model are formally proposed. Then, based on the

bilinear pairings, the concrete ID-DPDP protocol is designed. In the random oracle model, our ID-DPDP protocol is provably secure. On the other hand, our protocol is more flexible besides the high efficiency. Based on the client's authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.

## **V.CONCLUSION**

We presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphic verifiable response and hash Index hierarchy, we have proposed a cooperative PDP scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds.

Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly Demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems. As part of future work, we would extend our work to explore more effective CPDP constructions. Finally, it is still a challenging problem for the generation of tags with the length irrelevant to the size of data blocks. We would explore such a issue to provide the support of variable-length block verification.

## **VI.FUTURE WORK**

We would extend our work to explore more effective CPDP constructions. Finally, it is still a challenging problem for the generation of tags with the length irrelevant to the size of data blocks. We would explore such an issue to provide the support of variable-length block verification.

In multi-cloud storage, this paper formalizes the ID-DPDP system model and security model. At the same time, we propose the first ID-DPDP protocol which is provably secure under the assumption that the CDH problem is hard. Besides of the elimination of certificate management, our ID-DPDP protocol has also flexibility and high efficiency. At the same time, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification based on the client's authorization.

## REFERENCES

1. Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 12, DECEMBER 2012.
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," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.
3. A. Juels and B.S.K. Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
4. H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.
5. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm '08), pp. 1-10, 2008.
6. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession", In CCS '09, pp. 213-222, April 24,2012.
7. Feifei Liu, Davu Gu, Haining Lu,"An Improved Dynamic Provable Data Possession", Proceedings of IEEE CCIS2011, pp 290-295, 2011.
8. Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", The University of Alabama, Tuscaloosa, 24 March 2012.
9. Venkatesa Kumar V, Poornima G, "Ensuring Data Integrity in Cloud Computing", Journal of Computer Applications ISSN: 0974 – 1925, Volume-5, Issue EICA2012-4, February 10, 2012.
10. 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," Proc. ACM Symp. Applied Computing, pp. 1550-1557, 2011.
11. Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative Integrity Verification in Hybrid Clouds," Proc. IEEE Conf. Seventh Int'l Conf. Collaborative Computing: Networking, Applications.
12. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.