Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.778226

# MULTICLOUD STORAGE WITH IDENTITY-BASED PROVABLE DATA POSSESSION

**Naveen Reddy Nemili**
Research Scholar, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, Madhya Pradesh, India

**Narender Sharma**
Supervisor, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, Madhya Pradesh, India

**Sanjeev Shrivastava**
Co-Supervisor, Professor in Dept. of Computer Science & Engineering,
Guru Nanak Institute of Technology, Hyderabad, Telangana, India

## ABSTRACT
Due to the high maintenance and estimating costs, the customer is unable to keep a big volume of data on his local system. This is why the client provides cloud storage. The customer has security concerns such data integrity, confidentiality, etc., when storing data on the cloud. Clients may need to utilize data storage across many cloud servers. Both distributed storage and integrity checking are required, although the former is more pressing due to the need to reduce verification costs. Data stored on several cloud servers may be protected using a method called distributed provable data possession. It allows customers to check the safety of their outsourced data without having to download it all. In light of the foregoing, we present a new methodology for ensuring data integrity across several cloud storage providers using identity-based distributed proofs of possession (RDI-IDPDP). Based on distributed computing and Provable Data Possession, a concrete RDI-IDPDP protocol is built. The proposed RDI-IDPDP protocol is safe when the computational Diffie-Hellman (CDH) problem is hard, the decisional Diffie-Hellman (DDH) problem is easy, and the Gap Diffie-Hellman (GDH) groups are small. Additionally, our RDI-IDPDP methodology is both effective and flexible. Client verification, third-party verification, and Identity-based public-key cryptography that does away with the need for cumbersome certificate administration are all within the purview of the proposed RDI-IDPDP protocol.

**KEYWORDS:** Distributed computing, Cloud computing, Provable data possession, Identity-based cryptography, Bilinear pairing.

## INTRODUCTION
One such technique, known as Provable Data Possession (PDP), is suggested here to guarantee that data integrity is preserved at all times. However, this technique once again raises security concerns because it requires users to download data for verification. As a result, a system that can verify information without requiring users to download any data is crucial. PDP schemes such as Scalable PDP and Dynamic PDP were developed for this purpose. These methods relied heavily on one specific cloud storage service. The next blocks' integrity may be checked with the help of an authorized skip list in methods like SPDP, DPDP, and Merkle Hash Tree (MHT).

The inability to build MHT for a multi-cloud setting renders these strategies useless. The other systems such as CPOR and PDP make use of homomorphic verification tags when downloading data for verification is not necessary. The ability to demonstrate that the client's file (data) is unaltered and complete without having the tenant to download the data itself is known as "provable data possession." Although each revision has its own set of advantages and disadvantages, the PDP schemas have effectively solved the most of the initial worries with each iteration. Retrievability proofs (POR) are conceptually similar to PDP proof schemas. PDP proves to the client that the file has not been corrupted or removed from the server's possession. With POR, the client is able to conduct an effective audit protocol in which the server demonstrates that the file can be obtained. POR systems can also utilize error-correcting codes to recover and repair files with minor file corruption.

**Multi Cloud Storage**

A multicloud architecture is one in which an enterprise makes use of the resources of many cloud service providers. Multicloud systems use several different clouds instead of just one. These clouds might be public or private or a hybrid of the two. The term "distributed computing" is used to describe any massive group effort in which many individual PC owners donate some of their PC's processing time to the service of a massive issue. Each cloud administrator in our system is made up of a collection of data blocks. The information is placed in the multi cloud by the cloud user. Open architectures and interfaces form the basis of the cloud computing environment, allowing for the seamless integration of various private and public cloud services. This type of decentralized cloud infrastructure is known as a multi-Cloud. Clients of a multi-cloud setup have simple interfaces via which to access their data from afar. With multi-cloud storage, customers may employ a range of cloud storage providers to safeguard their data. Businesses can use either a private, public, or hybrid strategy for managing data stored in different clouds. The key possibilities for multi-cloud setups include managed service providers (MSPs), native public cloud services (PCS), and third-party cloud solutions (CSS).

There are several types of companies that offer multi-cloud storage services.

- Native cloud services from big cloud providers such as AWS, Google Cloud, and Azure. Storage types including databases, objects, and blocks are frequently supported.
- Services that combine many suppliers into one package, such as managed service providers, cloud service resellers, and cloud service providers. These services can be pre-integrated with your cloud service of choice.
- Services provided by a marketplace, such as virtual machine images or apps developed by third-party services.

Part of a larger multi-cloud architecture, in which services from several cloud providers are pooled into a single infrastructure, is multi-cloud storage.

**Data Integrity**

A simple definition of data integrity is the assurance that the data will remain unchanged and accurate across time. Data should be recorded as the Original, and when it is retrieved, it must be sent in the same form as the Original recording. Information security's groundwork is

ensuring the integrity of stored data. There is no data loss because every data integrity method is used. A simple definition of data integrity is the assurance that the data will remain unchanged and accurate across time. Data should be recorded as the Original, and when it is retrieved, it must be sent in the same form as the Original recording. Information security's groundwork is ensuring the integrity of stored data. There is no data loss because every data integrity method is used. Database operations, data warehousing, and business intelligence all benefit greatly from maintaining data integrity. Because maintaining data integrity guarantees accurate, consistent, and easily accessible information.

## LITERATURE REVIEW

**Ieuan Walker et al (2022),** Storage outsourcing, or the use of a third party to manage one's data warehouse, has grown increasingly common in recent years. There are a number of intriguing privacy and security concerns raised by this movement. Lack of responsibility is a major issue with cloud storage services. This article, critically evaluates two schemas/algorithms that allow customers to validate the integrity and availability of their outsourced data on untrusted data stores (i.e., third-party data storages). Proofs of retrievability (POR) and proven data possession (PDP) are the evaluated schemas. The purpose of each of these cryptographic techniques is to reassure users that their data is safe while stored in untrusted locations. The limitations of existing storage options are also identified, and a conceptual framework is offered to address them.

**P.Vijaya Kmari et al (2018),** By storing data in the cloud, users may access their files from anywhere and take use of high-quality cloud apps without having to worry about maintaining their own gear and software. Despite its obvious benefits, this type of service also requires customers to give up physical control of their outsourced knowledge, which introduces additional security issues related to the veracity of information stored in the cloud. In this paper, we propose a model for provable data possession (PDP) that enables a client that has stored data at an untrusted server to verify that the server possesses the original data without capturing it, mitigating this new challenge and paving the way for a secure and reliable cloud storage service. Proposing a flexible distributed storage integrity auditing system, using the homomorphic token and distributed erasure-coded data, the model provides probabilistic proofs of ownership by randomly selecting groups of blocks from the server, substantially reducing I/O costs. The proposed style lets customers to audit the cloud storage with terribly light-weight connection and compute value. In addition to providing reassurance for the accuracy of data stored in the cloud, the audit's findings also allow for rapid error localization in the data, or the identification of a malfunctioning server. Since cloud data is inherently dynamic, the suggested solution also enables safe and desirable dynamic operations on outsourced data, such as block change, deletion, and append. Through testing, we have shown that our proposed design is both very effective and secure against common threats including Byzantine failure, malicious data modification assault, and continuous server collusion.

## RESEARCH METHODOLOGY & DESIGN

People are increasingly turning to cloud-based services as cloud computing gains traction. Recently, the cloud has grown to accommodate other functions, such as hosting service platforms. High availability and large storage capacities have become problems in the cloud

because of their increasing popularity. Users may access their files from any location without the hassle of carrying about a USB and external hard drive when they save their data in the cloud. Another advantage of the cloud is the relatively low price per megabyte of data storage space. Many businesses have turned to cloud storage to construct their server infrastructure and database systems because of the low costs associated with doing so. Information uploaded by users is kept in a remote server. Since the cloud does not function as a dedicated, individual data repository, its contents are accessible to other users. This can lead to issues, such as a malevolent user gaining access to another user's data and misusing it for their own gain. As a result, customer confidence in cloud service providers declines and fewer people utilize the cloud. As a result, concerns about cloud security are crucial. Consequently, solutions are required to ensure the safety of information kept on the cloud. One way is to check the integrity of stored data. An extensive search strategy for checking whether or not all data modulation is under control is the foundation of data integrity verification. Multiple methods of checking the reliability of data are now under investigation.

**Provable Data Possession**

PDP is a technique for making sure that data remains unaltered. PORs are one type of integrity checking method among several. There are two stages to a PDP scheme: the preprocess and the verification. Pre-process produces and saves metadata for stored data. PDP is checked for accuracy using the metadata produced by the pre-process, query, and response statement. During the verification phase, the user makes a request to the server for details about a specific region of the file. The two stages of PDP are depicted in Figures 1 and 2.

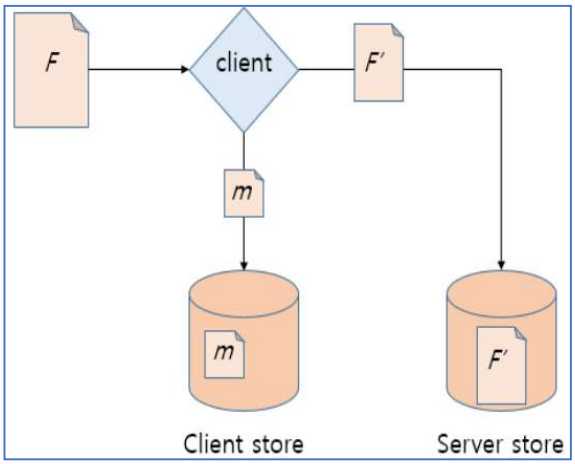


**Figure 1.1 Pre-Process in PDP**

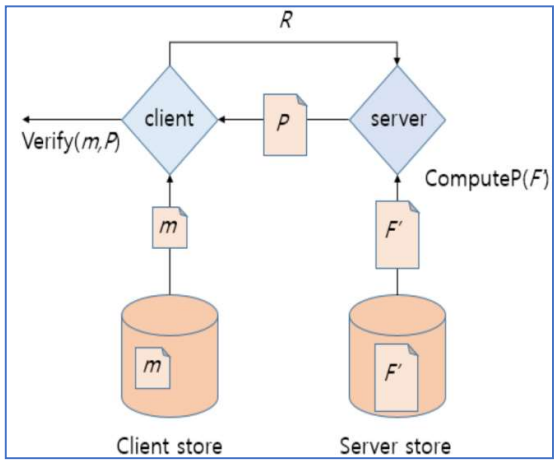


**Figure 1.2 Verification Process in PDP**

In order to retrieve a fragment of a file, a cloud storage server must first access the scattered files. PDP's strength is in its ability to detect forgeries or missing data without resorting to a sweeping search. Users save time and money by not having to download the entire database to complete a thorough search and verification. Data integrity verification techniques based on PDP have been investigated. There are two components to maximizing the PDP's impact. First, superior methods take into account the whole metadata and query statement size. When there isn't much of a size difference between the query statement and the whole file, the PDP benefit disappears. The information also includes the query statement's length. Second, complex methods should think about how much time and effort it will take to calculate the results of the pre-process calculations needed to produce the metadata and the verification process calculations needed to generate the query statements and replies. It is necessary to compute large operation expenses for each course if the PDP's efficiency is diminished because of the complexity of carrying out the operation. Cooperative Provable Data Possession (CPDP) was proposed by Y. Zhu et al. To summarize, CPDP is an approach to PDP that makes use of a bilinear group, a hash function, and an aggregation technique. Research into method-based CPDP has being place. In X. Sun and colleagues' PDP architecture, for example, the PDP is responsible for recreating the metadata whenever new data is registered or modified. A method for partially augmenting existing data with new information was suggested in this study. That procedure is feasible to partial insertion of the data. So, the recommended method is to avoid producing brand-new metadata for the modified files. A PDP program that might lessen the burden on the budget is also under investigation. Cloud storage relies heavily on remote data integrity checks. Distributed verifiable data custody is crucial for protecting off-site information in a multi-cloud setting. ID-DPDP (identity-based distributed proven data possession) is a unique methodology we propose for distant data integrity verification in multi-cloud storage. Under the difficulty assumption of the classic CDH (computational Diffi Hellman) issue, the suggested IDDPDP protocol is indisputable. The proposed ID-DPDP protocol is capable of achieving all three levels of verification: private, delegated, and public.

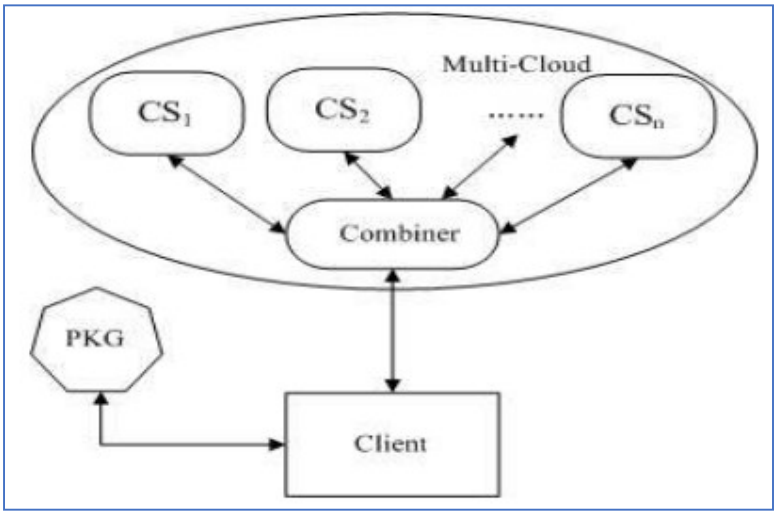


**Figure 1.3 The System Model of ID-DPDP**

This section describes the ID-PDP framework's model and security specification. There are four unexpected components to an ID-PDP gathering. Below, we usually illustrate them as follows: Client: AN element, that has expanding information to be put away on the multi-cloud for maintenance and processing, may be either lone client or partnership. The term "cloud server" (or "CS") refers to a piece of hardware managed by a cloud service provider and equipped with storage and processing power for use in handling customers' data. A component known as a "combiner" collects the required storage space and then sends the corresponding label sets out to the various cloud servers for comparison. It divides the exam into sections and sends them to several cloud servers after receiving the test. When accepting responses from remote servers, it combines them into a single response before sending it on to the protagonist. After receiving the character, a PKG (Private Key Generator) material produces the corresponding non-public key. This study explores the concept of proven data possession across clouds using identity-based public key cryptography. The certificate management can be removed to increase the protocol's efficiency. IDDPDP is the new paradigm for distant data integrity checking that we propose. There is a formal proposal for the system model and the security model. The concrete ID-DPDP protocol is then developed using the bilinear pairings. Our ID-DPDP protocol provides provable security in the random oracle setting. On the other hand, our protocol is more adaptable besides the high efficiency. The proposed IDDPDP protocol allows for private verification, delegated verification, and public verification based on client authorization.

## CONCLUSION

This paper provides a formalization of the ID-DPDP system concept and security model for use in multi-cloud storage. Concurrently, we present the first provably secure ID-DPDP protocol in the context of the CDH problem being assumed to be hard. Our ID-DPDP protocol is highly efficient and adaptable, and it does not require the administration of certificates. The proposed ID-DPDP protocol supports private verification, delegated verification, and public verification all at once, depending on the client's preferences. To facilitate dynamic scalability across numerous storage servers, we have presented a cooperative PDP technique based on

homomorphic verifiable response and the hash Index hierarchy. Furthermore, we demonstrated that our scheme had the security features necessary for a zero knowledge interactive proof system, making it secure enough to withstand attacks even when used as a public audit service on the cloud. Our studies also showed that our methods barely added any extra time to computations or communications. As a result, our method might be considered a promising alternative for ensuring data safety in cloud-based archives. We presented a model for verifiable information possession in which reducing file block visits, server processing, and client-server communication are all desired. Network traffic is reduced since the challenge/response protocol only requires a continual delivery of a small quantity of data. Therefore, huge data sets in widely dispersed storage systems are suitable for the PDP model of distant information verification. We offer an efficient and flexible distributed scheme with explicit dynamic knowledge support, such as block update, delete, and append, to accomplish the guarantees of cloud knowledge integrity and availability and enforce the quality of reliable cloud storage service for customers. An innovative explanation mechanism that uses a high rate of accuracy in the case of any business method with human-understandable can also improve the efficiency for secure storage and analysis of complex datasets is in our future plans.

## REFERENCES

[1] Huaqun Wang, Identity-Based Distributed Provable Data Possession in Multicloud Storage, IEEE Transactions on Services Computing, (Volume:8 , Issue: 2 )

[2] G. Ateniese, R. DiPietro, L. V. Mancini, G. Tsudik. Scalable and Efficient Provable Data Possession.SecureComm 2008.

[3] C. C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia. Dynamic Provable Data Possession.CCS'09, 213-222, 2009.

[4] F. Seb ´ e, J. Domingo-Ferrer, A. Mart´ ınezBallest ´ e, Y. Deswarte, J. Quisquater. Efficient Remote Data Integrity checking inCritical Information Infrastructures. IEEE Transactions on Knowledge and Data Engineering, 20(8):1034-1038, 2008.

[5] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu, S. S. Yau. Efficient Provable Data Possession for Hybrid Clouds.CCS'10, 756-758,2010.

[6] Y. Zhu, H. Hu, G.J. Ahn, M. Yu. Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage IEEE Transactions on Parallel and Distributed Systems, 23(12):2231-224, 2012.

[7] R. Curtmola, O. Khan, R. Burns, G. Ateniese. MRPDP: Multiple-Replica Provable Data Possession. ICDCS'08, 411-420,2008.

[8] A. F. Barsoum, M. A. Hasan. Provable Possession and Replication of Data over Cloud Servers. CACR, University of Waterloo,Report2010/32,2010.

[9] H. Wang. Proxy Provable Data Possession in Public Clouds.IEEE Transactions on Services ComputingTo appear, available on-line at http://doi.ieeecomputersociety.org/10.1109/TSC.2012. 35

[10] Z. Hao, N. Yu. A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability. 2010 SecondInternational Symposium on Data, Privacy, and E-Commerce, 84-89, 2010.

[11] A. F. Barsoum, M. A. Hasan, On Verifying Dynamic Multiple Data Copies over Cloud Servers. IACR eprint report 447, 2011.Available at http://eprint.iacr.org/2011/447.pdf
[12] H. Wang, Y. Zhang. On the Knowledge Soundness of a Cooperative Provable Data Possession Scheme in Multicloud Storage.IEEE Transactions on Parallel and Distributed Systems. To appear, available on-line at http://doi.ieeecomputersociety.org/10.1109/TPDS.201 3. 16
[13] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li. Enabling Public Auditability and Data Dynamics for Storage Security in CloudComputing. IEEE Transactions on Parallel And Distributed Systems, 22(5):847-859, 2011.
[14] A. Juels, B. S. Kaliski Jr. PORs: Proofs of Retrievability for Large Files. CCS'07, 584-597, 2007.
[15] H. Shacham, B. Waters. Compact Proofs of Retrievability.ASIACRYPT 2008, LNCS 5350, 90- 107, 2008.
[16] K. D. Bowers, A. Juels, A. Oprea. Proofs of Retrievability: Theory and Implementation. CCSW'09, 43-54, 2009.
[17] Q. Zheng, S. Xu. Fair and Dynamic Proofs of Retrievability.CODASPY'11, 237-248, 2011.
[18] Y. Dodis, S. Vadhan, D. Wichs1, Proofs of Retrievability via Hardness Amplification, TCC 2009, LNCS 5444, 109-127, 2009.
[19] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu. Zero Knowledge Proofs of Retrievability. Sci China InfSci, 54(8):1608-1617, 2011.
[20] D. Boneh, M. Franklin. Identity-based Encryption from the Weil Pairing.CRYPTO 2001, LNCS 2139, 213-229, 2001