International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Division and Replication of Data in Cloud for Optimal Performance and Security

Akula Thejaswini
Department of Computer Science and Engineering, CMR Engineering College,
Hyderabad, Telangana, India.

Praveen Chouksey
Department of Computer Science and Engineering, CMR Engineering College,
Hyderabad, Telangana, India.

Mrutyunjaya. S Yalawar
Department of Computer Science and Engineering, CMR Engineering College,
Hyderabad, Telangana, India.

Abstract—

Cloud technologies offer businesses a highly flexible and scalable storage infrastructure but also have significant security issues associated with their virtualized and shared nature. Traditional data protection (via encryption) guarantees confidentiality but has a high computational overhead and does not address any of the security vulnerabilities (e.g., attacks against virtual machines, nodes being compromised) of cloud computing technologies. In this paper, we present an alternative approach called "Division and Replication of Data in the Cloud for Optimal Performance and Security," or "DROPS," that will resolve these issues. Data originating from users will be divided into smaller, non-reconstructible fragments that will reside on multiple cloud nodes. Each of these fragments will then be selectively replicated (to provide high availability) so that data is protected and will not be lost if one or more nodes go offline due to failure. An unauthorized user cannot access an original file because there is no one node that can hold enough information to recreate it. Because cryptographic operations are now simpler, overall system performance is enhanced and latency time for retrieving data is decreased because these operations have been removed from the retrieval process. The system also optimally places the file fragments using the nodes' respective storage capacities, storage loads, and network bandwidths. Experimental tests in a virtualized cloud environment confirmed that the system provided better security, scalability, and availability than traditional methods.

Keywords: Cloud Computing, Data Fragmentation, Selective Replication, Cloud Security, Fault Tolerance, Performance Optimization.

I. INTRODUCTION

In relation to how cloud computing has transformed the way we store data and access it, this has allowed organizations to utilize cheaper, fully scalable, on-demand services; however, with that much of the infrastructure in the cloud being virtualized and shared, they also have created multiple security problems relating to that shared infrastructure. Since there are multiple users of the same physical resources, sensitive information may be subject to unauthorized access and malicious activity, including node compromise. The likelihood of these security incidents increases significantly when users rely on third-party cloud providers, as they have very little control over infrastructure security. These types of environments offer attackers the opportunity to take advantage of weaknesses in a shared computing environment. The shared responsibility model for users and cloud service providers managing the security of the underlying infrastructure

(i.e., IaaS) has created a new level of complexity around ensuring digital assets are properly protected in the cloud, resulting in many potential security gaps due to humans being unable to implement effective protection measures. When either a virtual machine or a node has been compromised, this could mean that a significant amount of data may have been exposed through a number of different attacks, including a 'virtual machine escape' attack that has successfully bypassed a traditional security measure. Moreover, poor data sanitization and weak configurations only compound the risks of data loss. These limitations indicate that a system-wide approach is necessary rather than only relying upon security measures implemented by users themselves [4], [5], [6], [7], [9].

Most traditional cloud security products leverage encryption methods (i.e., AES or DES). While encryption methods do provide confidentiality for data, they also create a significant amount of computational overhead that can produce a significant impact on overall performance and increase the amount of time it takes to retrieve data. In addition, encryption does not protect against infrastructure-based types of attacks, such as side-channel attacks or VM-based exploits, as an example. Even if you encrypt your data, an attacker that targets a compromised node may be able to access your data through other methods, even if they are using a different method of access than the compromised node. Therefore, encryption by itself will not provide full protection in the cloud, especially in situations where you also require efficient performance [4], [8], [9].

To address these issues, the DROPS methodology takes a different approach to cloud storage by using data fragmentation and replication of data in a distributed manner. Instead of saving entire files (documents) on one physical node, the DROPS methodology divides data into fragmented pieces and stores and

distributes those fragments on multiple cloud service providers. This means that each piece (fragment) of data can be stored in an independent manner and replicated across multiple locations, thereby eliminating the ability for any one node to have enough data to reconstruct a complete file. This significantly lowers the chance of losing data due to a failure of a single node. Furthermore, by eliminating the complexity of encryption systems, the DROPS methodology is able to provide intelligent fragment placement, load balancing and dynamic redistribution to maintain a high level of service to users while improving the overall security level in comparison with traditional encryption methods [1], [2], [3], [10].

II. Literature Survey

Historically, cloud data storage has been done by storing the entire file onto each cloud node independently with no auto duplication of data and making them highly vulnerable to security breaches. If one node was compromised, an attacker could then access the entire file resulting in an enormous amount of data being lost by the holder of that file. Furthermore, the traditional model of cloud storage whereby the data was stored centrally created the potential for cascading attacks, where compromise of one node would result in affectation of other nodes and components of the cloud storage system. These weaknesses were identified as a need for distributed mechanisms for storage, which would minimize the reliance on a single node to provide a complete dataset at one single time. DROPS solves this issue by design, in that no node contains a complete dataset; thus, the potential for loss due to breaches is mitigated [2], [5], [11], [12].

There have been various attempts to create trusted third-party solutions utilizing PKI (Public Key Infrastructure) in order to improve the security of these cloud systems. These solutions relied on Certification Authorities to provide management of keys and to ensure secure communications. Even though the use of Certification Authorities has improved the authentication and integrity of data, they also created centralized failure points; that is, should the Certification Authority or Key Management System be compromised, all users connected to the cloud would be at risk. There is nothing worse than to have an external trust entity create greater complexity and reduce reliability for a given system. An example of how to overcome the dependency on a centralized trust model is through the use of the DROPS methodology that does not rely on a single entity and decentralizes data using distributed fragments of the data across multiple systems [5], [8], [9].

Public key cryptography can also be implemented in conjunction with techniques like threshold-based secret sharing, where keys can be split among numerous nodes. Multiple nodes can also perform computations on their individual portions, and then later be used together to recreate the same data. In this manner, complexity increases in an environment with multiple types of security, and therefore the processing load imposed upon the system is increased, as well as the amount of latency associated with completing other actions associated with a given application or service. It should also be noted that techniques of this kind are typically not practical for use in real-time systems due to the latency imposed between the original request to complete a form of computation and when that form of computation is actually completed. The split distribution/security method via DROPS, on the other hand, will allow for maintaining high levels of security while not adding latency associated with processing in real-time and providing increased efficiency [10], [13].

Some current systems that use Data Management Techniques together with Encryption to increase confidentiality and storage efficiency. Even though these are very good approaches to protecting data, they still depend on encryption and, therefore, will have high processing overhead and impact scalability. Additionally, they do not adequately address the risks that virtualization and infrastructure attacks pose to the system. Even strong encryption cannot stop an attack that is targeted at the infrastructure where the cloud is hosted. In contrast, the DROPS approach solves this problem by assuring that each piece of data fragment (source node only) does not possess a meaningful value when it exists independently of the other fragments. DROPS uses a combined fragmentation-selective replication-intelligent node selection technique to represent files as non-reconstructable fragments that are distributed according to specific factors (e.g., each node's reliability, load, or performance). The use of replication provides information availability when nodes fail, while the use of dynamic redistribution allows for system stability and performance improvement following node failure events. Unlike traditional approaches where encryption is used as the only means of securing information, DROPS does not use encryption; thus significantly reducing programmatic and computational overhead while improving overall performance. Finally, the integration of all of these features into one cohesive framework yields a balanced solution which will help improve data security, availability, and efficiency in cloud-computing environments [1], [2], [3], [4].

III. Proposed Methodology

The DROPS (Division and Replication of Data in the Cloud for Optimal Performance and Security) solution is able to effectively and securely manage data from users that are stored in Cloud environments by means of combining data fragmentation and data selective replication techniques. While traditional methods of managing user data in Cloud environments commonly use (encryption) and use Centralized storage, whereas DROPS uses (fragmenting) users' data into many smaller, non-reconstructible fragments, which are then distributed to different Cloud nodes. The fact that no single node holds enough of the original file to allow the file to be reconstructed means that the risk of a breach is greatly reduced. Each fragment has an identical copy so that another node may have access to that fragment if the first copy becomes unusable.

There are criteria to determine the location of each fragment in the system once the fragments are physically separated across many different nodes (The amount of storage space available on the node, the workload on the node, and the bandwidth of the network). These criteria optimise resource use and maintain balance within the overall system.

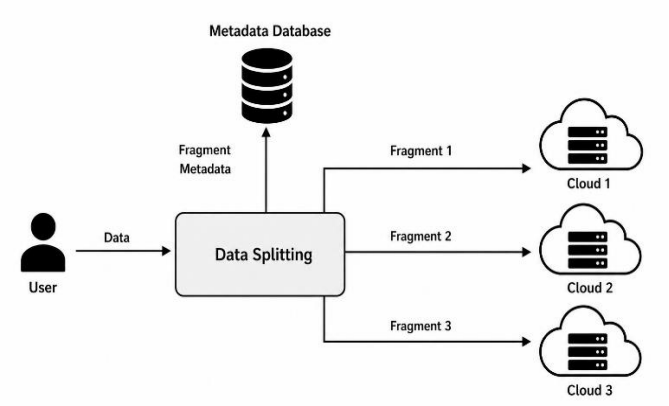

Fig. 1. Block diagram of the proposed DROPS system.

Showing how the DROPS system works; Fig 1 shows that a user's data is processed with three different types of operations: fragmenting the user data, placing them in storage, and rebuilding the user files. The DROPS system has three components: data splitting module, metadata repository, and multiple cloud nodes. The data splitting module takes a user's input file and breaks it down into smaller sized pieces of data. The metadata repository provides information about the different locations where each piece of data has been stored and the DROPS exposes each separate piece of data in different cloud nodes. Because each node does not have an entire copy of a user's data, if an unauthorized person gets access to any one of the nodes, they are only able to see a small amount of data, which may not provide any useful information to them about the value of the user's data. When the user requests to retrieve their data, the DROPS system will gather the required information from multiple cloud nodes and reassemble the original user file; completing the secure and efficient access to the user's data.

A. Data Fragmentation Mechanism

The fragmentation process starts when a user file is divided into small pieces called fragments using a predefined criteria. Each fragment is created to be non-reconstructive on its own meaning that there isn't enough information in the fragment to reconstruct the original user file. This eliminates the risk of having an entire file stored on one cloud node. Distributing fragments instead of original data decreases the probability of loss if a cloud node is compromised and therefore prevents any unauthorized attempts to reconstruct information.

B. Metadata Management and Mapping

Once documents have been broken up into smaller pieces (the fragments), the system will have a metadata database. This database has information on where each fragment is located at each of the cloud nodes, which is crucial for being able to find out where the fragments are when you are retrieving them.

By storing this metadata about where the fragments are and allowing them to be found and reassembled without having the entire file stored together in one place, it will ultimately lead to more efficient retrieval from the system while continuing to have a fully distributed system.

C. Fragment Distribution and Selective Replication

Distributed around different nodes that can be used to store data in accordance to the choice of nodes adjacent to other nodes based on capacity of the node, load of the node and state of the network can lead to multiple copies of the same data being stored in different parts of the cloud by applying a single strategy for choosing where to place fragmentation once it has been generated, as having copies of the same data will be treated as though they have been duplicated and there should only ever be one copy of each fragmentation. By ensuring that neither copy of the original is placed onto the same node as the original there is ensured both data security and volume redundancy.

D. Data Retrieval and Reconstruction

When retrieving data, the system will reference metadata to gather all of the necessary parts from various cloud nodes. The necessary parts will then be put together to recreate the original file through data reconstruction. The parts are scattered over multiple nodes, thereby minimizing delays and enabling efficient access throughout the process. The replicated pieces will enable the system to retrieve a data item, regardless of whether or not any single node is down or damaged, thus preserving both availability and reliability.

IV. Results and Discussion

The DROPS system has been assessed, as stated in a later section of the text, through a series of experiments that demonstrate the functional ability to successfully execute four key operations: (1) data fragmentation; (2) distribution of fragments across cloud nodes; (3) storage of the respective data fragments; and (4) reconstitution of the original file from fragment downloads. Results of the implementation of the DROPS system show that files are fragmented into new smaller components, then distributed to multiple cloud nodes and can be successfully restored when requested. The DROPS system design feature of multiple cloud nodes not hosting complete files enhances the security of the data stored within the system, while still allowing high availability of files through selective replication.

A. Fragment Generation

Fig. 2. Fragment generation result of the DROPS system.

Fig 2 provides evidence that the file has successfully been split up into numerous pieces (fragments). A total of ten fragments were produced from the input file. Each one of them represents only part of the original data; therefore, there is not enough information in each fragment to reassemble or recreate the original file by itself. Therefore, the fragmentation has been implemented correctly; and guarantees that no one location will contain all of the complete data for the file.

B. Fragment Distribution and Replication

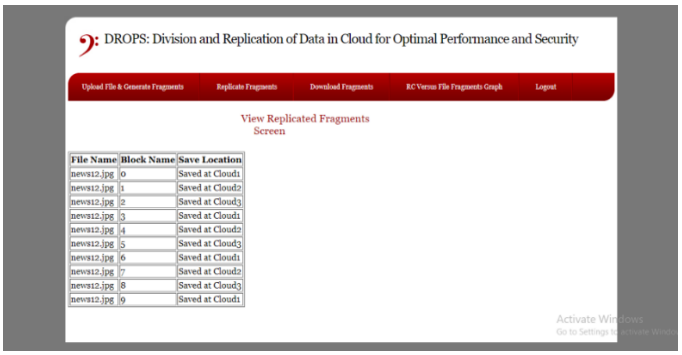

Fig. 3. Fragment replication and distribution across cloud nodes.

The distribution of fragments across Cloud 1 - 3 is illustrated in Fig. 3. Each fragment is recorded to be contained on its own individual node and has a replica on a different node. This verifies that the application has implemented selective replication such that no single node will possess all data at one time, but will still provide for data availability in the event of a node failure.

C. Fragment Storage in Cloud Nodes

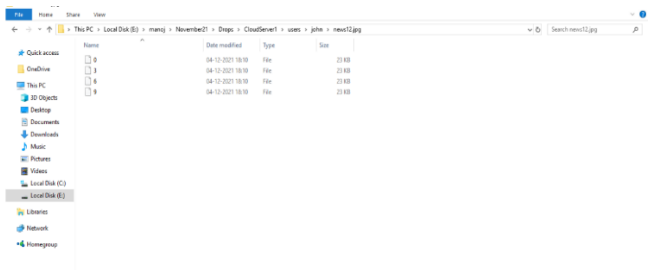

Fig. 4. Stored fragments in a cloud node.

Fig 4 shows how only a part of the fragments can be held in any one cloud node, and hence the system has been designed to place fragments in different cloud nodes (i.e., not all fragments of a file will be on the same node). It also shows that the originals and their copies of the same segment are stored separately, which increases the overall level of data safety.

D. Data Retrieval and Reconstruction

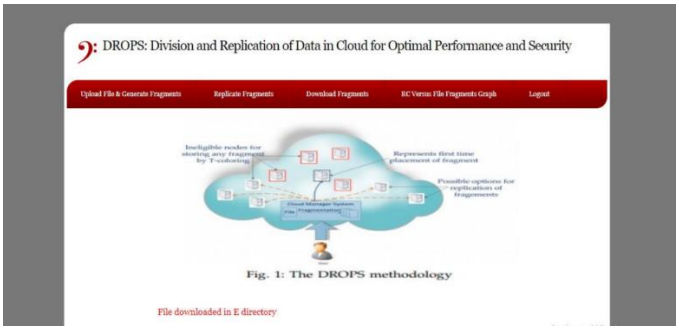

Fig. 5. Successful data retrieval and reconstruction.

The original file is reconstructed from fragmented parts in Fig 5. The system utilizes metadata information to find where the fragments are stored and recover them from the various cloud nodes, where they have been stored. After retrieving all of the fragments, they are restored together as a whole. The figure also indicates that the reconstruction of the original file is accurate, thereby confirming that the retrieval process was successful.

E. Performance Analysis

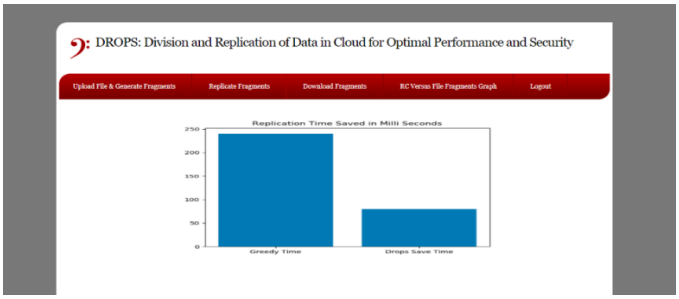

Fig. 6. Performance comparison of replication time.

The data for comparison between the proposed DROPS method to a traditional approach for snapshot-based replication is shown in Fig. 6. The results indicate that DROPS is more efficient than a traditional approach at reducing the time to replicate data, through optimized replication of data distribution and selective replication of files. Therefore, the performance of the proposed system has improved, while maintaining the security of the data being replicated and the availability of the replicated data.

TABLE I
RESULT ANALYSIS OF THE DROPS SYSTEM

Operation	Observation	Outcome
Fragmentation	File divided into 10 fragments	No complete data in a single node
Distribution	Fragments stored across Cloud 1, Cloud 2, and Cloud 3	Data distributed across multiple nodes
Replication	Each fragment replicated once on a different node	Ensures availability during node failure
Storage	Partial fragments stored in each node	Prevents full data exposure
Retrieval	Fragments combined using metadata mapping	Successful file reconstruction

The data in TABLE I all demonstrate the operation of the proposed DROPS system for the three non-traditional data fragmenting processes: (1) fragmentation; (2) distribution; and (3) replication. For example, the Fragmentation result indicates that the input file has been disassembled into (or fragmented into) several smaller file fragments, so no contiguous space exists for storing the entire input file on a single storage node. Similarly, the Distribution result shows how the file fragments have been distributed across many data storage cloud nodes so as to eliminate any potential for centralized data storage. The Replication result shows that each file fragment has been replicated once and stored on a separate storage node so as to guarantee the availability of the file fragment in the event of a storage node failure. Further evidence of the above reassures the storage results as to the fragmentation, distribution, replication, and retrieval processes being performed correctly.

The results support the conclusion that the proposed DROPS system can be used for data fragmentation, partitioning in multiple cloud nodes, and optional replication of data. The results from this study show that no single cloud node holds all copies of data while providing access to the data when a cloud node fails. The fact that the original file can be reconstructed shows that the retrieval process using metadata mapping is correct. In addition, the decrease in time to replicate the data shows an increase in performance of the system compared to traditional methods.

Overall, these findings show that the DROPS methodology affords a viable method for distributed storage and retrieval of data in cloud computing environments.

V. CONCLUSION

The DROPS methodology is an effective method for managing data in a cloud environment through the use of fragmentation along with selectively replicating data across clouds. The data created by users is fragmented into smaller fragments that cannot be reconstructed and are defined by a set of rules. The usage of fragmentation and selective replication allows for a distribution to various cloud nodes in a way that the end user does not have access to a complete data set from one location. Each fragment of data is replicated onto a different node of the cloud therefore providing the end user with a complete file should the original node on which that fragment is stored fail. The use of metadata in the retrieval of the data provides the means for reconstructing the original data files. The experimental results have shown that the system performs in accordance with the process of fragmentation, distribution, replication and reconstruction of the data. The results have also demonstrated that a decrease in time for the replication of data has also improved the overall performance of the system when compared to traditional methodologies. The DROPS methodology provides a practical means to increase the efficiency and availability of distributed data storage.

Future work may explore adaptive fragmentation strategies and lightweight security enhancements for deployment across diverse cloud environments.

REFERENCES

[1] F. Castro-Medina, L. Rodriguez-Mazahua, A. López-Chau, M. A. Abud-Figueroa, and G. Alor-Hernández, "FRAGMENT: A Web Application for Database Fragmentation, Allocation and Replication over a Cloud Environment," *IEEE Latin America Transactions*, vol. 18, no. 6, pp. 1126–1134, 2020, doi: 10.1109/TLA.2020.9099751.

[2] C. A. Ramirez-Gutierrez, M. Morales-Luna, and O. Rodríguez-Hernández, "Application of Data Fragmentation and Replication Methods in the Cloud: A Review," in *Proc. IEEE ICEV*, 2019, doi: 10.1109/ICEV.2019.8673249.

[3] M. Ali, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, "DROPS: Division and Replication of Data in the Cloud for Optimal Performance and Security," *IEEE Systems Journal*, 2015, doi: 10.1109/JSYST.2014.2379646.

[4] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in Cloud Computing: Opportunities and Challenges," *Information Sciences*, vol. 305, pp. 357–383, 2015, doi: 10.1016/j.ins.2015.01.025.

[5] D. Zissis and D. Lekkas, "Addressing Cloud Computing Security Issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012, doi: 10.1016/j.future.2010.12.006.

[6] S. Subashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011, doi: 10.1016/j.jnca.2010.07.006.

[7] M. Armbrust et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010, doi: 10.1145/1721654.1721672.

[8] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9, doi: 10.1109/INFCOM.2010.5462173.

[9] H. Takabi, J. B. D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010, doi: 10.1109/MSP.2010.186.

[10] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010, doi: 10.1109/TIT.2010.2054295.

[11] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," in *Proc. ACM CCS*, 2009, pp. 187–198, doi: 10.1145/1653662.1653686.

[12] G. Ateniese et al., "Provable Data Possession at Untrusted Stores," in *Proc. ACM CCS*, 2007, pp. 598–609, doi: 10.1145/1315245.1315318.

[13] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979, doi: 10.1145/359168.359176.