International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 13th Jul 2022. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-11&issue=Issue 07

DOI: 10.48047/IJIEMR/V11/ISSUE 07/04

Title **Best performance of Cloud security uses Division and Replication of Data**

Volume 11, ISSUE 07, Pages: 22-34

Paper Authors
Jonnalagadda Sravani, V Lakshmi Chetana, D.Prasad

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Best performance of Cloud security uses Division and Replication of Data

[1]Jonnalagadda Sravani, [2]V Lakshmi Chetana, [3]D.Prasad
[1] PG Student, Department of CSE, DVR&Dr.HS MIC College Of Technology, Kanchikacherla,AP
[2] Assoc.Professor, Department of CSE, DVR&Dr.HS MIC College Of Technology ,Kanchikacherla,AP,
[3] Assoc.Professor, Dept of CSE, DVR&Dr.HS MIC College Of Technology, Kanchikacherla,AP
[1] sravanijonnalagadda1998@gmail.com, [2] lakshmichetana@mictech.ac.in, [3] prasad@mictech.ac.in

Abstract:
Security issues arise when data is outsourced to a third-party administrative authority, as is done in cloud computing. Attacks by other users and cloud nodes could lead to a data compromise. High security measures are therefore necessary to safeguard data in the cloud. The applied security method must, however, also consider how to speed up data retrieval. In this study, we propose the division and replication of data in the cloud (DROPS), which takes a combined approach to performance and security challenges. In the DROPS process, a file is divided into pieces, and the pieced-together data is replicated among cloud nodes. Each node only keeps a single piece of a specific data file, preventing the attacker from learning any useful information even in the event of a successful attack. In addition, the nodes that store the fragments are spaced out by a predetermined amount using graph T-coloring to prevent an attacker from speculating where the fragments reside. Additionally, the DROPS methodology frees the system from computationally expensive procedures by not relying on conventional cryptographic techniques for data protection. We demonstrate how unlikely it is to find and compromise every node holding a single file's pieces. We contrast the DROPS methodology's performance with that of several other techniques as well. The enhanced level of security comes with a small performance penalty.

Introduction:
The use and administration of the information technology infrastructure have been transformed by the cloud computing paradigm [7]. On-demand self-services, widespread network access, resource pooling, elasticity, and measurable services are characteristics of cloud computing [22], [8]. Because of the aforementioned qualities, cloud computing is an obvious contender for adoption by companies, organisations, and individual individuals [19].
However, increasing security risks come along with the advantages of low cost, minimal management (from a users perspective), and greater flexibility [7].

One of the most important factors preventing the widespread use of cloud computing is security [14], [19]. Cloud security issues may result from cloud characteristics (data recovery vulnerability, Internet protocol vulnerability, etc.), cloud service offerings (structured query language injection, weak authentication schemes, etc.), or cloud technology implementation (virtual machine (VM) escape, session riding, etc.). [5]. The participating entities must all be secure for a cloud to be secure. The highest level of security for any system with numerous units is equal to the level of security for the weakest entity [12]. As a result, in a cloud, asset security is not solely dependent on a person's security measures [5].The nearby entities could give an attacker a chance to get past the user's defenses.

Users of the cloud service for off-site data storage must move data in a virtualized,

shared environment, raising a number of security issues. A cloud's pooling and flexibility enable several users to share its physical resources [22]. Additionally, the shared resources may be transferred to different users at some point, which could endanger data through data recovery techniques [22]. Additionally, a multi-tenant virtualized system could cause a VM to evade virtual machine monitor's detection (VMM). The escaped VM can cause other VMs to gain access to data that is not authorised [9]. Cross-tenant virtualized network access could also jeopardise the integrity and privacy of data. Customer's private information may potentially be leaked via improper media sanitization [5].

Data that is transferred to a public cloud must be protected. It is necessary to stop unauthorised data access by other users and processes (whether intentional or unintentional) [14]. Any weak entity can endanger the entire cloud, as was previously mentioned. In this case, even after a successful intrusion into the cloud, the security mechanism must significantly increase an attacker's effort to extract a meaningful amount of data. Additionally, the likelihood of loss (due to data leakage) must be kept to a minimum.

Throughput, dependability, and security must all be ensured by a cloud [15]. Data retrieval time is a crucial aspect in determining a cloud's throughput [21]. Data replication solutions are used in large scale systems to address the issues of data dependability, data availability, and response time [3]. The attack surface for that specific data is increased whn data is replicated across several nodes. In the cloud, for example, storing m replicas of a file rather than one raises the likelihood that a node containing the file will be selected as an attack victim from 1 n to m n, where n is the total number of nodes.

We can infer from the explanation above that security and performance are essential for the upcoming large-scale systems, such clouds. Because of this, we address the challenge of security and performance in this study as a secure data replication problem. We introduce Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS), which judiciously judicially fragments user files into pieces and copies them at key cloud locations. Based on specified user criteria, a file is divided into pieces so that no useful information is contained in any particular fragment. Each cloud node—we refer to computational, storage, physical, and virtual machines as nodes—contains a unique fragment to improve data security. If a single node is successfully attacked, the locations of other cloud fragments must remain a secret. We choose the nodes such that they are not contiguous and are a specific distance apart from one another to further increase security and prevent an attacker from knowing the locations of the file pieces. T-coloring is a method used to assure node separation [6]. The nodes are chosen based on the centrality measurements that provide a faster access time in order to speed up data retrieval time.

We judiciously replicate fragments over the nodes that receive the most read/write requests in order to further reduce the retrieval time. There are two stages to the nodes' selection process.
Based on the centrality metrics, the nodes are chosen for the initial placement of the pieces in the first phase. The replication nodes are chosen in the second phase. A high-level work flow of the DROPS methodology's operation is displayed in Fig. 1. In order to compare our replication strategies to the DROPS methodology, we use ten of them.

The replication strategies that have been implemented are: (a) A-star based searching technique for data replication problem (DRPA-star); (b) weighted A-star (WA-star); (c) A-star; (d) suboptimal A-star1 (SA1); (e) suboptimal A-star2 (SA2); (f) suboptimal A-star3 (SA3); (g) local min-min; (h) global min-min; I greedy (GRA). For better system performance, the aforementioned procedures are fine-grained replication approaches that determine the number and locations of the replicas. Three data centre network (DCN) architectures—the three tier, fat tree, and DCell—are used in our research. Since the aforementioned architectures make up contemporary cloud infrastructures and the DROPS approach is suggested to function for the cloud computing paradigm, we apply them.

Three data centre network (DCN) architectures—the three tier, fat tree, and DCell—are used in our research. Since the aforementioned architectures make up contemporary cloud infrastructures and the DROPS approach is suggested to function for the cloud computing paradigm, we apply them.

The following are the main contributions we make in this paper:

- We create a plan for outsourced data that considers both performance and security. The suggested method duplicates and divides the data file across cloud nodes.
- The suggested DROPS technique makes sure that, even in the event of a successful attack, the attacker receives no useful information.
- For data security, we don't rely on conventional encryption techniques. The suggested scheme's non-cryptographic nature makes the necessary operations (data placement and retrieval) faster.
- For increased security, we ensure that the file fragments are reproduced in a controlled manner, with each fragment being replicated only once.

An overview of the relevant work in the topic is given in Section 2. We offer the preliminary information in Section 3. In Section 4, the DROPS approach is described. The experimental design and findings are discussed in Section 5, and the study is wrapped up in Section 6.

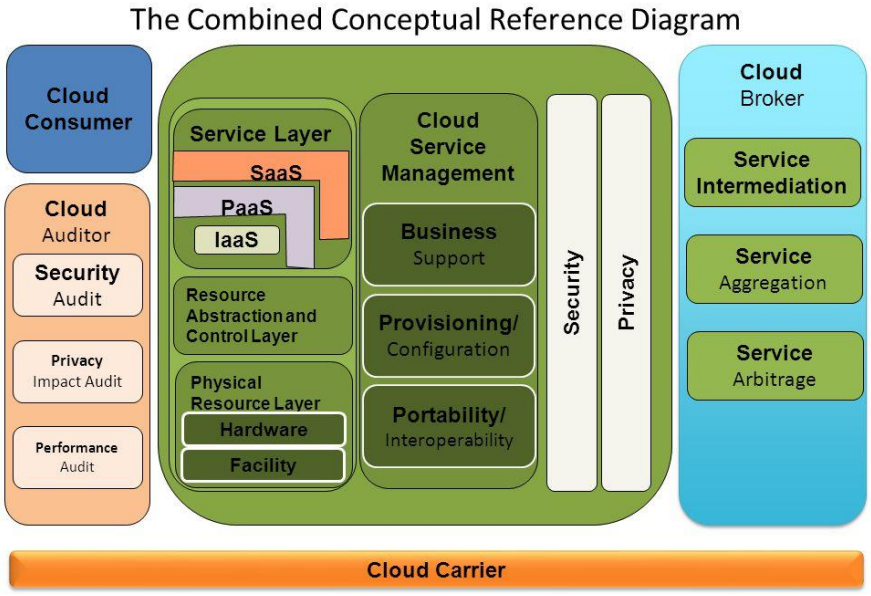
Related Work

A method to guarantee the availability, integrity, and freshness of data in a cloud was presented by Juels and Opera [10]. The Iris file system is in charge of performing the data migration to the cloud. The company has created and implemented a gateway application that uses a Merkle tree to guarantee the data's integrity and freshness. At various layers of the tree, the file blocks, MAC codes, and version numbers are kept. The proposed method in [10] significantly relies on the user's chosen data secrecy strategy. Furthermore, it is impossible to reduce the likelihood of loss in the event of data tampering due to breach or access by other VMs. For data security, our suggested system does not rely on conventional encryption methods. Additionally, the DROPS approach avoids storing the entire file on a single node to prevent data compromise in the event of a successful assault on the node.

The authors of [11] used consolidated storage and native access control to address concerns with virtualization and multi-tenancy in cloud storage. It is suggested to use the Dike authorization architecture, which combines native access control with tenant name space isolation. The suggested system is created for and

functions with object-based file systems. However, the loss of important data due to insufficient sanitization or malicious VMs is not dealt with. The DROPS approach uses many nodes to store a single file and data file fragmentation to prevent the leakage of crucial information. [22] recommends using a dependable third party to handle cloud security services. The level of trust in the authentication, integrity, and confidentiality of data as well as the communication between the parties involved was increased by the authors using the public key infrastructure (PKI).

The certifying authorities create and maintain the keys. At the user level, it was suggested to store the keys on devices that could withstand temperature changes, like smart cards. Similar to this, Tang et al. used public key cryptography and a reliable third party to secure data in cloud environments [20].



The PKI infrastructure, however, has not been utilised by the authors of [20] to save overhead. Public/private key generation and administration are done by the trustworthy third party. It's possible for the trusted third party to consist of one or more servers. By combining public key cryptography and (k, n) threshold secret sharing techniques, the symmetric keys are safeguarded. Nevertheless, because to problems with virtualization and multi-tenancy, such techniques do not shield the data files against tampering and loss.

It provides a safe and effective location for data objects in a distributed system. On various sites throughout the network, an encryption key is partitioned into n shares. The (k, n) threshold secret sharing system allows for the partition of a key into n shares. Clusters have been created within the network. Heuristics are used to decide the placement and quantity of replicas. In each cluster, a primary site is chosen to distribute the cluster's replicas. The replication problem is combined with security and access time enhancement in the technique reported in [21].

However, the technique only concentrates on the safety of the encryption key. The data files are processed as a single file and

are not fragmented. On the other hand, the DROPS approach fragments the file and stores the fragments on other nodes. Additionally, [21] does not take into account the DROPS methodology's focus on the security of data within the cloud computing realm.

In contrast to traditional infrastructure solutions that cannot meet the cost, scalability and other requirements for supporting these modern applications, Cloudian's HyperStore offers an ideal data repository or lake for Vertica environments:

- Drop-in integration: Seamlessly integrate with Vertica using Vertica's capability to use S3 as the main repository for the hybrid cloud.
- Modular, limitless elasticity: Start with three low-cost nodes and expand simply by adding devices to the cluster without disrupting analytics workloads.
- Multi-tenancy: Allow multiple users to analyze data sitting on a single, shared data lake or data warehouse—without compromising security—while also employing multi-tenant billing, metering and quality of service (QoS) controls.
- Hybrid-cloud readiness: Employ policy-based tools to replicate or tier data to AWS, Google Cloud Platform, Microsoft Azure, or to another HyperStore cluster for offsite disaster recovery, capacity expansion or data analysis in the cloud.
- Data resiliency: Get up to 14 nines of resiliency along with administrator-defined storage policies for implementing it based on replication or erasure coding.
- Ransomware protection: Through S3 Object Lock, prevent malware from encrypting data—enabling quick, easy recovery of an unencrypted backup copy without paying ransom—and meet governance and legal hold demands.
- Military-grade security: Further secure data with features such as secure shell, integrated firewall, RBAC/IAM access controls, AES-256 server-side encryption for data at rest and SSL for data in transit, as well as certification with the most stringent government security requirements.

"With enterprises looking to leverage analytics applications on-prem to gain greater insights from the data stored there, having a modern storage infrastructure is critical," said Jeff Healey, vice president of marketing of the Vertica Product Group at Micro Focus. "By partnering with Cloudian, we're enabling our customers to capitalize on a leading object storage platform and maximize the value of their digital assets."

"Over the past two years, Cloudian has introduced a range of new solutions to support organizations' modern application demands," said Jon Toor, chief marketing officer, Cloudian. "With data gravity and data sovereignty driving the move toward hybrid cloud models, the Cloudian-Vertica solution combines the benefits of cloud-based data warehouses with the security, control and other advantages of keeping data behind the firewall."

Preliminaries

For the researchers' convenience, we discuss the associated concepts in the following before delving into the specifics of the DROPS methodology.

Data Fragmentation

A large-scale system's security, like the security of the cloud, is dependent on both the overall system's security and the security of each individual node. A successful intrusion into a single node could have detrimental effects on the other nodes as well as the data and applications on the victim node. Because the entire file is present, the information on the victim node may be fully disclosed [17]. A

software or administrative vulnerability may be the cause of a successful intrusion [17]. In homogeneous systems, it is possible to use the same weakness to attack more system nodes. Less effort will be needed to successfully attack succeeding nodes compared to the initial node. Comparatively, heterogeneous systems call for greater work. However, only one node needs to be breached in order to compromise a single file. Making copies of a data file and keeping them on different nodes might lower the amount of communicated data [17], [21]. A successful intrusion will only grant access to a small subset of data that might not be of any importance.

Furthermore, there is a very little chance that an attacker will detect fragments on every node if they are unsure about the positions of the fragments. Consider a file with z number of fragments and a cloud with M nodes. The number of successful intrusions on different nodes, s, must be more than z. The likelihood that all z sites hosting the file fragments represented by P(s,z) are present on s number of victim nodes is given as:

$$P(s,z) = \frac{\binom{s}{z}\binom{M-s}{s-z}}{\binom{M}{s}}.$$

If M 1/40, S 1/40, and Z 1/47, then P 10; 7 1/4 0:0046. But if we select M 1/40, s 1/40, and z 1/45, then P 20; 15 1/4 0:000046. The likelihood of a state decreases further as M rises. Therefore, we can state that the likelihood of an attacker obtaining the data file decreases as M increases. A considerable reduction in the likelihood that an attacker will be able to access a sizable amount of data is seen in cloud systems with thousands of nodes. The time it takes to get the data will grow if each fragment is only stored once in the system. Fragments can be replicated to speed up data retrieval time in a way that doesn't raise the overall retrieval time.

Centrality

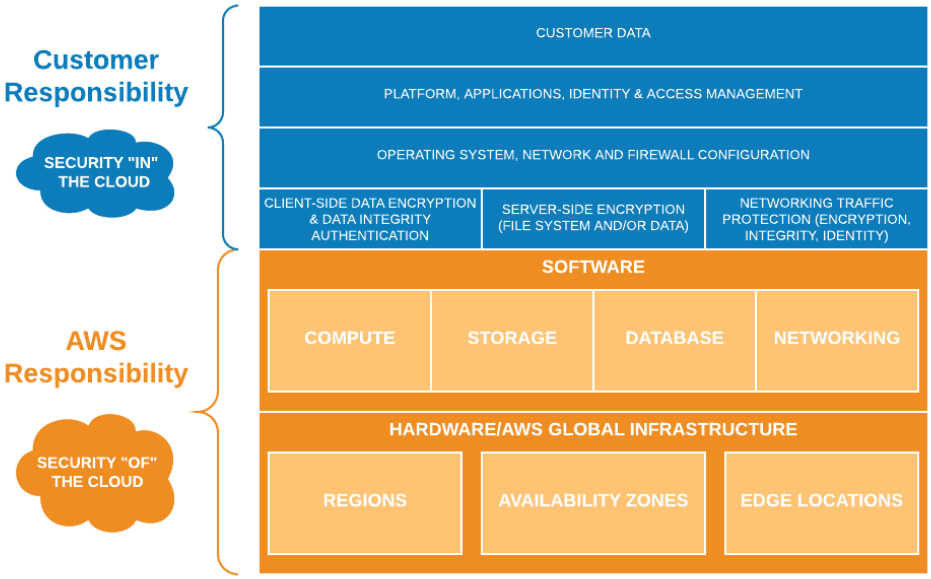
The measure of a node's relative importance in a network is given by its centrality in a graph. The objective of faster retrieval times in replication increases the significance of the centrality measures. There are several different centrality metrics, including proximity, degree, betweenness, eccentricity, and eigenvector centrality. Because we are using the aforementioned three centralities in this work, we only go into detail about the proximity, between-ness, and eccentricity centralities. We recommend that readers review [24] for the remaining centralities.

The number of shortest paths that travel through a node n and connect it to other nodes is known as its between ness centrality [24]. Formally, any node v in a network has the following between ness centrality:

$$C_b(v) = \sum_{a \neq v \neq b} \frac{\delta_{ab}(v)}{\delta_{ab}},$$

where dab is the total number of shortest paths between a and b, and dabðvÞ is the number of shortest paths between a and b passing through v. The variable CbðvÞ denotes the between ness centrality for node v.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



Closeness Centrality

If the sum of a node's distances from all of the other nodes in a network is less than the sum of the distances of other potential nodes from all of the other nodes, the node is said to be closer to all of the other nodes [24]. The node is more central if its total distances from the other nodes are lower. In a network, a node's formal closeness centrality is described as:

$$C_c(v) = \frac{N-1}{\sum_{a \neq v} d(v,a)},$$

where N is total number of nodes in a network and dðv; aÞ represents the distance between node v and node a.

DROPS

A file stored at a node in its entirety creates a single point of failure in a cloud system [17].

If a node is successfully attacked, the confidentiality, integrity, or both of the data may be at risk. The aforementioned scenario may happen as a result of incursion or unintentional mistakes. By using replication algorithms in these systems, performance in terms of retrieval time can be improved. Replication, however, multiplies the amount of file copies in the cloud. Consequently, the likelihood that the node hosting the file will fall prey to an assault, as mentioned in Section 1, increases. A large-scale system like the cloud requires security and replication because both are used to deliver services to the end user. Replication and security must be balanced so that neither service degrades the other's service quality.

We recommend against storing the full file at a single node when using the DROPS approach.

The file is fragmented and replicated via the cloud according to the DROPS approach. No node in a cloud can contain more than one fragment due to the distribution of the fragments, meaning that even a successful attack on a node won't provide any substantial information. To increase security, the DROPS approach employs controlled replication, where each fragment is copied just once in the cloud. Despite not increasing retrieval times to the same extent as full-scale replication, managed replication greatly increases security.

The user uploads the data file to the cloud using the DROPS approach. After receiving the file, the cloud manager system—a user-facing server that responds to user requests—performs the following operations: (a) fragmentation; (b) first cycle of node selection and storage of one fragment over each selected node; and (c) second cycle of node selection for fragment replication. The cloud manager is thought to be a secure entity that maintains track of the placement of the fragments.

The file owner specifies that the data file's fragmentation threshold will be generated. The threshold for file fragmentation can be set by the file owner in terms of either percentage or the total number and size of pieces. For example, the percentage fragmentation threshold may specify that each fragment must be 5% of the file's overall size. Alternately, the owner may create a new file that contains details on the fragment number and size, such as fragment 1 being 5,000 bytes in size and fragment 2 being 8,749 bytes in size.

We contend that the appropriate person to determine the fragmentation threshold is the file's owner. As the owner is aware of all the details relevant to the data, they may best divide the file so that each fragment does not include a substantial quantity of information. If the user does not indicate the fragmentation threshold while uploading the data file, the default % fragmentation threshold may be made a part of the service level agreement (SLA). In this study, we primarily concentrate on the security of the storage system on the presumption that the communication path between the user and the cloud is secure.

Following file fragmentation, the DROPS methodology chooses the cloud nodes where the file fragments will be placed. Security and performance in terms of access time are both equally important while making the choice. To improve access times, we pick the nodes that are closest to the centre of the cloud network. The DROPS technique employs the idea of centrality to shorten access times towards the aforementioned goal. As was covered in Section 3.2, the centralities determine how central a node is depending on several metrics.

We create the set T starting at zero and working our way up to the created random number after creating a non-negative random number. To limit the node selection to nodes that are at hop-distances outside of T, the set T is employed. For the aforementioned reason, we provide the nodes colours in such a way that they all start out with the open color. All of the nodes in the neighbourhood at a distance belonging to T are given close color after a fragment has been added to the node. We lose some of the central nodes in the aforementioned process, which can lengthen retrieval times, but we also raise security levels.

The location of the other fragments cannot be known if the intrusive party compromises a node and takes a fragment. The attacker can only continue making educated guesses about where the other components are. But as was already mentioned in Section 3.1, the likelihood of a successful coordinated strike is incredibly slim. Until all of the fragments are positioned at the nodes, the procedure is repeated. The method for placing fragments is represented by Algorithm 1.

To improve data availability, reliability, and retrieval speed, we undertake a controlled replication in addition to putting the fragments on the central nodes. With the aim of reducing access costs and increasing retrieval times for accessing fragments needed for file reconstruction, we place the fragment on the node that offers them. The separation of fragments, as described in the placement procedure

through T-coloring, is also taken care of while reproducing the fragment.

It is also possible that part of the fragments are left without being replicated due to the T-coloring in cases of a high number of fragments or few nodes. As was previously mentioned, T-coloring forbids storing a fragment nearby a node that is already storing one, which eliminates a number of nodes that could have been used for storage. In this scenario, only the remaining fragments are stored on the nodes that aren't currently carrying any fragments.

Discussion

With a certain amount of an attacker's work, a node gets compromised. A successful assault on a cloud node will result in the compromise of an entire data file if the affected node saves the file entirely. However, if the node only keeps a portion of a file, then a successful assault only retrieves a portion of the data file. An attacker would need to gain control of a significant number of nodes in order to successfully compromise the DROPS approach, which saves data file fragments across various nodes. Because each compromised node may not provide a fragment in the DROPS methodology because the nodes are rated separately based on T-coloring, the number of compromised nodes must be bigger than n.

where EConf stands for the amount of work needed to compromise confidentiality, EAuth for the amount of work needed to compromise authentication, and EBreakIn for the amount of work needed to break into a single node. The security of the authentication method is not a concern in this study because our main focus is on the security of cloud-based data. Therefore, we may argue that an attacker's effort must rise by a factor of n in order to obtain n fragments. Additionally, the attacker must properly guess the nodes that are holding file fragments when using the DROPS approach. Therefore, in the worst situation, all of the nodes storing the file fragments will be present in the set of nodes compromised by the attacker. Equation (1) demonstrates that there is very little chance that the worst-case scenario will come to pass. In contrast to the worst case scenario, there is a substantial possibility that some of the machines (average case) containing the file fragments will be chosen. The damaged fragments, however, won't be sufficient to recreate the entire dataset. The number of nodes storing pieces chosen for an assault determines the worst, average, and best cases in terms of likelihood. Equation thus encompasses all three instances (1).

The DROPS approach may handle assaults that target a compromised node in addition to generic attacks where the attacker gains access to user data by evading or thwarting security protections. Some of the assaults that the DROPS approach can handle are listed in Table 2.

The attacks that are being discussed are cloud-specific and are based on cloud-core technologies. It is important to note that even in the case of successful assaults (such as those discussed), the DROPS approach assures that the attacker receives just a portion of the file because only one fragment is stored on the node. Additionally, the node that holds the fragment must be the target of the successful assault.

Experimental Setup and Results

The data centre network serves as the foundation for cloud computing's communication system [2]. Three DCN architectures—the three tier, the fat tree, and the DCell—are used in this article [1]. The three-tier DCN architecture is the original design. However, the Fat tree and Dcell architectures were introduced [2] in order to address the rising demands of

cloud computing. Therefore, we assess the effectiveness of our plan on both legacy and cutting-edge architectures using the three aforementioned architectural styles.

Switch-centric networks include the three-tier and fat tree topologies. The access layer switches are used to connect the nodes. Aggregate layer switches are used to connect many access layer switches. The switches on the core layers connect to the switches on the gate layers. The Dcell is a server-centric network architecture that employs both servers and switches to facilitate network communication [1].

In the Dcell architecture, a server is linked to other servers using a switch. The higher level dcells are constructed recursively from the lower level dcells. The identical-level cells are completely connected. The readers are invited to study [1] and [2] for more information on the aforementioned designs and their performance analyses.

Comparative Techniques

The following fine-grained replication strategies were used to compare the DROPS methodology's findings: (a) DRPA-star, (b) WA-star, (c) A-star, (d) SA1, (e) SA2, (f) SA3, (g) Local Min-Min, (h) Global Min-Min, I Greedy algorithm, and (j) Genetic Replication Algorithm. A data replication algorithm called DRPA-star is based on the A-star best-first search algorithm. The root node, or null solution, is where the DRPA-star begins. Costn 14 gn hn, where g(n) is the path cost for reaching n and h(n) is referred to as the heuristic cost and is an estimate of the cost from n to the goal node, is how the communication cost at each node n is calculated.

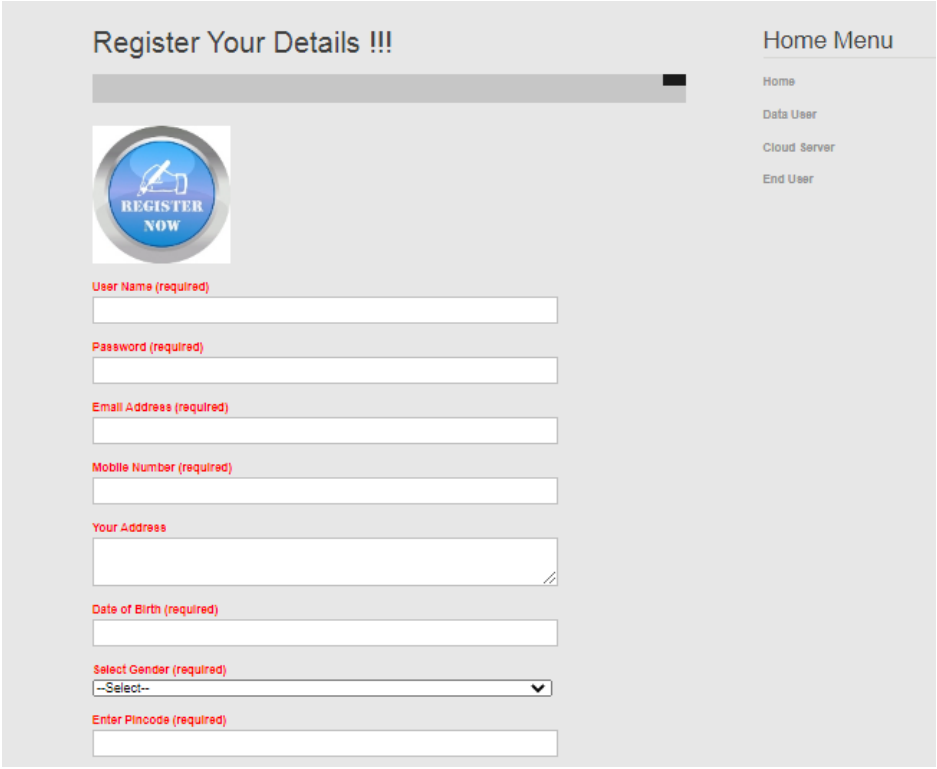
The DRPA-heuristic star's is specified as hn 14 max0; mmk(n), where mmk(n) is the least expensive replica allocation or the max-min RC. Readers are urged to read [13] for further information on DRPA-star. A weighted function is used by the WA-Star, a development of the DRPA-star, to assess cost. The function is written as follows: fn 14 fn hn 1 dn=Dhn. The node n's depth is represented by the variable d(n), while the expected depth of the destination node is indicated by the letter D [13].

The A-star is another DRPA-star variant that use the two lists OPEN and FOCAL. Only nodes from the OPEN list with f larger than or equal to the lowest f by a factor of 1 + are included in the FOCAL list. Instead of using the OPEN list, the FOCAL list is used to execute the node expansion. [13] provides more information on WA-Star and A-star. The DRPA star-based heuristics SA1 (sub-optimal assignments), SA2, and SA3 are used. Only the best descendants of node n with the lowest expansion cost are chosen in SA1 at level R or below.

Only chooses the best node n successors for the first time when it reaches depth level R. All additional successors are eliminated. The SA3 functions similarly to the SA2, with the exception that all nodes but the lowest-cost node are removed from the OPEN list. For more information on SA1, SA2, and SA3, readers are urged to read [13]. The bin packing method can be thought of as a specific case of the LMM. Based on the RC of the file fragments that will be stored at a node, the LMM sorts the file fragments.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Figuare 1. Home Page



Figuare 2: Registration Page

Conclusions

We put forth the DROPS technique, a security plan for cloud storage that addresses both security and performance in terms of retrieval time. The data file was broken up into pieces, and the pieces are scattered across many nodes. The nodes were divided using T-coloring. In the event of a successful attack, the adversary would not have access to any significant information thanks to the fragmentation and dispersal. No node in the cloud kept more than one piece of a single file. The effectiveness of the DROPS process was evaluated in comparison to full-scale replication methods. The simulation results showed that when security and performance were prioritised simultaneously, the level of data security increased and performance slightly decreased. With the DROPS approach as it stands, a user must download the file, update its contents, and then upload it once more. It is wise to create an automatic updating system that updates only the necessary parts. The time and resources used for downloading, updating, and uploading the file again will be saved by the aforementioned future work. Additionally, it is important to research the effects of TCP incast on the DROPS approach as it relates to distributed data access and storage.

References

[1] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," Concurrency Comput.: Prac. Exp., vol. 25, no. 12, pp. 1771–783, 2013.

[2] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya,"On the characterization of the structural robustness of data center networks," IEEE Trans. Cloud Comput., vol. 1, no. 1, pp. 64–77, Jan.–Jun. 2013.

[3] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," in Proc. IEEE Globecom Workshops, 2013, pp. 446–451.

[4] Y. Deswarte, L. Blain, and J.-C. Fabre, "Intrusion tolerance in dis- tributed computing systems," in Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy, Oakland, CA, USA, 1991, pp. 110–121.

[5] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," IEEE Secur. Privacy, vol. 9, no. 2, pp. 50–57, Mar./Apr. 2011.

[6] W. K. Hale, "Frequency assignment: Theory and applications," Proc. IEEE, vol. 68, no. 12, pp. 1497–1514, Dec. 1980.

[7] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Info. Sci., DOI: 10.1016/j. ins.2015.01.025, 2015.

[8] M. Hogan, F. Liu, A. Sokol, and J. Tong, "NIST cloud computing standards roadmap," NIST Special Publication, Jul. 2011.

[9] W. A. Jansen, "Cloud hooks: Security and privacy issues in cloud computing," in Proc. 44th Hawaii IEEE Int. Conf. Syst. Sci., 2011, pp. 1–10.

[10] A. Juels and A. Opera, "New approaches to security and availability for cloud data," Commun. ACM, vol. 56, no. 2, pp. 64–73, 2013.

[11] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, "Virtualization-aware access control for multitenant filesystems," in 30th IEEE Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–6, 2014.

[12] L. M. Kaufman, "Data security in the world of cloud computing," IEEE Secur. Privacy, vol. 7, no. 4, pp. 61–64, 2009.

[13] S. U. Khan and I. Ahmad, "Comparison and analysis of ten static heuristics-based Internet data replication techniques," J. Parallel Distrib. Comput., vol. 68, no. 2, pp. 113–136, 2008.

[14] A. N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," Future Gener. Comput. Syst., vol. 29, no. 5, pp. 1278–1299, 2013.

[15] A. N. Khan, M. L. M. Kiah, S. A. Madani, and M. Ali, "Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing," J. Supercomput., vol. 66, no. 3, pp. 1687–1706, 2013.

[16] T. Loukopoulos and I. Ahmad, "Static and adaptive distributed data replication using genetic algorithms," J. Parallel Distrib. Comput., vol. 64, no. 11, pp. 1270–1285, 2004.
[17] A. Mei, L. V. Mancini, and S. Jajodia, "Secure dynamic fragment and replica allocation in large-scale distributed file systems," IEEE Trans. Parallel Distrib. Syst., vol. 14, no. 9, pp. 885–896, Sep. 2003.
[18] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the place- ment of web server replicas," in Proc. IEEE Comput. Commun. Soc. 20th Annu. Joint Conf., 2001, vol. 3, pp. 1587–1596.
[19] M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K. Li, and A. Y. Zomaya, "SeDaSC: Secure data sharing in clouds," IEEE Syst. J., DOI: 10.1109/JSYST.2014.2379646, 2015.
[20] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Trans. Dependable Secure Comput., vol. 9, no. 6, pp. 903–916, Nov. 2012.
[21] M. Tu, P. Li, Q. Ma, I.-L. Yen, and F. B. Bastani, "On the optimal placement of secure data objects over Internet," in Proc. 19th IEEE Int. Parallel Distrib. Process. Symp., 2005, p. 14.
[22] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Gener. Comput. Syst., vol. 28, no. 3, pp. 583–592, 2012.
[23] J. J. Wylie, M. Bakkaloglu, V. Pandurangan, M. W. Bigrigg, S. Oguz, K. Tew, C. Williams, G. R. Ganger, and P. K. Khosla, "Selecting the right data distribution scheme for a survivable storage system," Department of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-01-120, May 2001.
[24] M. Newman, Networks: An Introduction. London, U.K.: Oxford Univ. Press, 2009.
[25] V. S. Rao, V. Mounika, N. R. Sai and G. S. C. Kumar, "Usage of Saliency Prior Maps for Detection of Salient Object Features," *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2021, pp. 819-825, doi: 10.1109/I-SMAC52330.2021.9640684
[26] G. S. C. Kumar, D. Prasad, V. S. Rao and N. R. Sai, "Utilization of Nominal Group Technique for Cloud Computing Risk Assessment and Evaluation in Healthcare," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 927-934, doi: 10.1109/ICIRCA51532.2021.9544895
[27] Mandru D.B., ArunaSafali M., Raghavendra Sai N., Sai Chaitanya Kumar G. (2022) Assessing Deep Neural Network and Shallow for Network Intrusion Detection Systems in Cyber Security. In: Smys S., Bestak R., Palanisamy R., Kotuliak I. (eds) Computer Networks and Inventive Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 75. Springer, Singapore. https://doi.org/10.1007/978-981-16-3728-5_52
[28] N. R. Sai, G. S. C. Kumar, M. A. Safali and B. S. Chandana, "Detection System for the Network Data Security with a profound Deep learning approach," 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 1026-1031, doi: 10.1109/ICCES51350.2021.9488967.