JOURNAL OF OPTOELECTRONICS LASER ISSN: 1005-0086
Volume 41 Issue 11, 2022

For Secure Dynamic Data Deduplication With Cloud Storage: Augumented Collinear Encryption Key Generation

Dr. R. Murugadoss
Professor, Department of Computer science and Engineering, V.S.B. College of Engineering Technical Campus, Coimbatore- 642109

ABSTRACT
Today's corporate environment heavily relies on cloud computing since it allows for the on-demand delivery of computing services to consumers via the Internet. From the perspective of the customer, the fundamental benefit of adopting cloud storage is that they can spend less money on buying and maintaining data storage and only spend for the requested quantity of storage, which can be scaled up or down as needed. A decrease in data quantities could assist providers in lowering the expenses of operating huge storage systems and lowering energy usage in light of the expanding size of data of cloud computing. To improve the storage efficiency of cloud storage, techniques for deduplication have been developed. Key generation and management for convergence encryption and deduplication procedures are handled by this mechanism. The way that data is used changes over time as a result of the fluid nature of cloud storage. At one time, some data chunks might be regularly accessed, but not at others. Due to reliability requirements, some databases may need a lot of redundancy, but others may be frequently accessed or changed by many people. Therefore, it is crucial that cloud services have dynamic capabilities. Although most current solutions are static, this limits their ability to fully apply to the fluid nature of online storage data. In this research, we present a dynamic deduplication approach for cloud services with the objective of enhancing storage efficiency while retaining variety for fault tolerant.

Keywords: Avaiability, Cloud storage; Cloud Computing; Dependability; Deduplication.

I. INTRODUCTION

A form of Internet-based computing known as "cloud computing" makes information and shared computational capabilities available on demand to computers and other devices. [1] The actual labour that an instance or group of instances will undertake is represented by the concept of workload. Since the cloud is made up of several large servers, load balancing methods are utilised to distribute the load across them.

[2] The cloud backup services have two issues. There is network bandwidth and storage space. The amount of digital data has grown significantly in recent years. One storage technology that allows for several copies of the same data to be stored is data redundancy. Data backups that may be useful for disaster recovery. For the purpose of storing the data content, a lot of storage space is required. [4] This causes an increase in storage costs, particularly for IT organisations that store a lot of data. Data backup evolved into a cost-effective strategy. Due to the limited network bandwidth between the user and service provider, the other difficulty is a wide backup window.

[5] One compression method that can get rid of multiple data copies is data deduplication. Only unique content is allowed in the cloud storage due to data deduplication. Moreover, broaden the network's bandwidth. [6] The volume of information that must be sent across a network. Large data objects are divided into discrete processes known as blocks or chunks. Create a distinct key for each block using the fingerprint cryptographic hash function. [7] Then, using the index lookup table, substitute the duplicate blocks with their hash fingerprints. Finally, send the distinctive pieces for communication or data storage in the cloud.

[8] Data deduplication is implemented using two methods. There are finger prints and deduplication methods based on deltas. The oldest way for chunking is the delta deduplication strategy, although it does not seek for identical but similar data blocks. [9] All of the chunks are fingerprinted using the cryptographic hash function in the fingerprint base data deduplication approach. After that, it can look for the same file in the index lookup table. It cannot be stored if the file already exists. Since it is not similar, it can be transferred across the network after being put in the index lookup table.

[10] The information that is kept in a cloud backup may be safe. because personal information cannot readily be downloaded and is not accessible by others. Deduplication is followed by encryption and decryption on the data for safe backup. Cloud innovation offers a wide range of potential for daily use. [11] Cloud is named for its ability to handle requests. The cloud illustration is a creation. It can be accessible from anywhere at any time that is necessary. Modern consumers demand innovation wherever it is possible. Information is stored and processed on cloud

DOI: 10050086.2022.12.05

innovation external server farms, and cloud computing enables users and activities across their memory configurations.

[12] To address consistency and cost-effectiveness in volume marketplaces, cloud computing exchanges assets. The cloud is a very big idea that is supported by extensive collaboration and numerous specialised administrations. It is possible to access mutually adjustable computer assets, like administration, servers, storage solutions, and software, through a good on-demand platform using the cloud. Assets are released with little administrative work and processed quickly. The benefits of high computational power, low administration costs, elite, flexibility, adaptability, and availability make cloud registration a huge assist or benefit nowadays. For traders, the cloud is advancing slowly at a 50%–55% range.

[13] To make cloud advantages more effective and clear, there are a few initial challenges that need for fair thought. Data deduplication is a fantastic cloud support method that also scoops up programmes for shortening the fortification duration, enhancing the storage system's effectiveness and limiting usage.

[14] Parity with huge data volumes is a problem in VM (Virtual Machine) and HPC systems, according to recent studies. Such trials saved 30% of the typical storage capacity by up to 90% in memory space and 70% in high-performance constrained structures, which were utilised to create data replication in massive data sets. For instance, by advancing the deduplication of data, the flow time for the actual VM movement in the cloud could be reduced.

Erasure coding is frequently utilised in warehouse systems to maintain minimum overhead storage while delivering excellent accessibility and ongoing effectiveness. Eradication coding was a process that compromises and destroys software security in different systems. The information cannot be divided into smaller pieces, enlarged, coded, or kept at a safe distance from extra information bits thanks to this technique. Utilizing metadata from data stored in a cluster's circle storage phase, deletion coding seeks to reproduce debased data. Erasure coding is useful when used with extensive knowledge endeavors and any projects or strategies. [15] The system must be deceptive, such knowledge grids, circular clusters, recorded and dispersed storage, and object warehousing. In object-based cloud storage, deletion coding was applied. Storage frames quickly change as a result of the rising use of several applications, using ever-larger rings and networking devices. Data security methods are becoming more and more important because of the constantly new technology as well as the greater risk of failing parts.

II. PROPOSED SYSTEM MODEL

Overall Architecture
Our system currently relies on full file hashing for client-side deduplication. The client-side hashing process links to any available deduplicator depending on its current load. The deduplicator then locates the duplicate by comparing it to the hash values already present in.

Metadata Server.
When a file is uploaded to a file server using a typical deduplication system, the file's logical path and new hash value are both recorded in the metadata server. The file will have more references if it does in fact exist.

Some systems might maintain a fixed multiple copies of each file. However, in order to increase availability, the files with a lot of references could need more copies. Some recent works added a level of redundancy to deduplication systems to address this problem. We suggest a deduplication system that takes into account the dynamic nature and Quality of Service (QoS) of the Cloud system in order to increase reliability while retaining storage effectiveness. In our simulation environment, the Redundancy Manager first determines the duplication, and then, depending on the number of referrals and required degree of QoS, it determines the best number of copies for the file. Based on the fluctuating number of references, the degree of QoS, and the need for the files, the number of copies is dynamically modified. The redundancy manager will recalculate the ideal amount of copies when the modifications are detected, such as when a file is removed by a user or the file's quality of service level is changed. Figure 1 depicts the system model we've suggested. The following components make up the system:

DOI: 10050086.2022.12.05

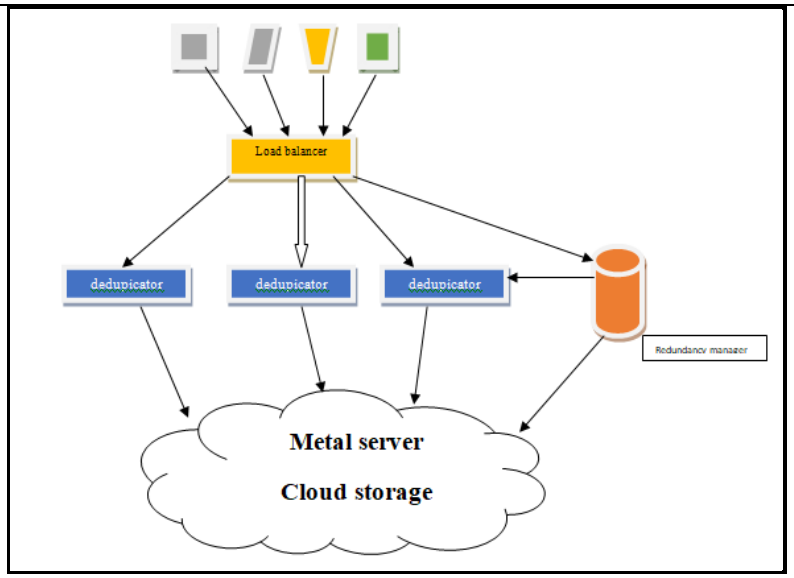

Figure 1: Proposed System Model

***Load Balancer*:**
After SHA-1 hashing, has sent a fingerprint (hash function) to a deduplicator through the load balancer. The balancer replies to queries clients send to each of the deduplicators in accordance with their current loads.

***Deduplicators*:**
A component created to locate duplicates by comparing them to the hash values already present in the metadata server.

***Cloud Storage*:**
Several file servers to store actual files and duplicates of them, as well as a metadata server to keep metadata.

***Redundancy Manager*:**
A part that counts the initial number of copies and keeps track of the QoS's fluctuating level.

Simulation Environment
Both HDFS Simulator and CloudSim are Java-based toolkits with distinct simulation objectives. Although CloudSim has some storage-related classes that can be expanded, the current architecture is still in its infancy and necessitates the installation of a module that enables cloud storage modeling in order to test out novel replica management techniques. Even though the replication level is a predetermined and static amount, HDFS Simulator already offers replication mechanisms, making it more appropriate to our work. Nevertheless, replica dynamicity can be added by altering the source code. Additionally, we can do trials by mimicking scenarios like the varying QoS level.

Through the use of modeling, the process of this work is assessed, giving academics the chance to model large-scale cloud settings, specifically failure occurrences in the cloud, and also aid in the assessment of QoS metrics like performance and availability. To mimic our suggested model of the system, the HDFS Simulation ideas were modified. 5 Datanodes are created as File servers, and one Namenode is created as a Metadata server. The metadata server stores data in XML format. The versions of files are kept on file servers. We simulated the following three activities: update,delete, and upload.

Upload
A hash value from the client-uploaded file is called by the deduplicator, which then searches the metadata server for any copies with the same hash value. A new file will have its new metadata included to the system and be uploaded to a file server if it is one. Depending on the quality of service (QoS) of the uploaded file, duplicates of the file will be made.

DOI: 10050086.2022.12.05

Update

Existing files will have their metadata changed, and depending on the file's max QoS value, the system could need to produce or delete copies of the file.

Delete

The deduplicator counts the number of files that the user chooses to delete and which share the same hash value. All copies of the file would be removed if the hash was only mentioned once. In contrast, just the metadata would be upgraded if any other files make reference to the hash, and the number of copies of the file would have to be reduced in accordance with the optimum amount of QoS.

Figure 2 depicts an abstract prototype cloud system model with KGMaaS,Cloud Storage (CS), and Users (U), and explains how deduplication is accomplished with concurrent encryption in the subsequent steps..

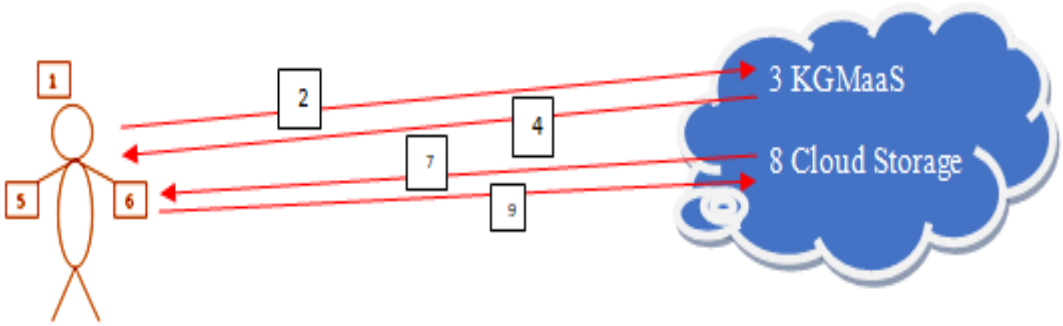

Figure 2. Abstract Clouds Infrastructure with Deduplication Methodology Diagram

1. Users who want to upload data (UD) to the cloud must first create tokens (TKN) from the user's data using the primitive function TKN=Tkn Gen (UD).
2. In order to obtain a key for creating a convergent encryption key, the token TKN is sent to the KGMaaS across a secure channel (CEK).
3. KGMaaS checks the metadata to see if the specified TKN already exists in the database. If it does, KGMaaS sends the user the corresponding key that was previously produced for the same TKN. In the event that the TKN is missing from the metadata, KGMaaS produces a key (GCK) for the TKN and sends it to the user. Each user's TKN and the related key information are maintained in metadata by KGMaaS.
4. KGMaaS provides the user with the relevant key GCK so they can create a convergent encryption key CEK.
5. Using the key GCKreceived from the KGMaaS and the suggested algorithm CEK = CEkey Gen(GCK), users construct the convergent encryption key.
6. Using the cryptography encryption method ED=CEnc Alg, UD is encrypted with the convergent key following key creation (UD, CEK). Following encryption, a Tag (TG) is created from the encrypted data (ED) using the formula TG = Tag Gen(ED) in order to confirm whether the data is duplicated.
7. Users send the TG to the cloud to see if the info has already been saved there or not.

Improved Divergent Encryption Key Generation Proposed (ECEKGA)

Convergent encryption uses the recipient's original data to generate a one-of-a-kind (hash code), which is then used as the secret to encrypt the data. The converging encryption public key algorithm has the following formula: CEK = CEkey Generation (GCK). Convergent encryption keys are also referred to as CEK (convergent encrypted key), CEkey Gen() (basic function for divergent encryption generation), & GCK (key generated by KGMaaS again for token TK provided by the users). The KGMaaS sends the user a key GCK, and the user can produce a CEK by using the value and the recommended. When consumer gets a secret GCK from the KGMaaS, and utilising the key and the recommended convergent encryption key generation method, the user can generate a CEK. The proposed convergent encryption key construction algorithm employs a digesting algorithm that consumes user data and executes the necessary steps to create the CEK. The following figure shows the suggested convergent encryption key generation algorithm.

DOI: 10050086.2022.12.05

```
1. U_D←user's data
2. N←sizeof(U_D)
3. U_D[N]←array(U_D)
4. for i← 1 to N
     Asc U_D[i]←ascii(U_D[i])
   next i
5. j←1
6. k←1
7. for i← 1 to N
     if (i%2==0) then
       E_BLOCK[k]←Asc U_D[i]
       k←k+1
     else
       O_BLOCK[j]← Asc U_D[i]
       j←j+1
   end if
   next i

8. Mid←N/2
9. for i← 1 to Mid
       S_BLOCK[i]←E_BLOCK[j]+O_BLOCK[j]
   next i
10. for i← 1 to Mid
       BinU_D←append(binary(S_BLOCK[i]))
   next i
11. BinN←sizeof(BinU_D)
12. Blck←BinN/256
13. m←1
14. Bin←block of 256 0's
15. for i← 1 to Blck
     Binbk[i]←split(BinU_D, m, m+255)
     m←m+256
     Bin←binary_addition(Bin,Binbk[i])
     next i
16. BinU_D←Bin ⊕ GC_K
17. Blck←sizeof(BinU_D)/8
18. m←1
19. for i←1 to Blck
      DecU_D[i] ←decimal(split(BinUp, m, m+7))
      AscU_D[i] ←ascii(DecU_D[i])
      AscBuff←append (AscU_D[i])
      m←m+8
     next i
20. CE_K←AscBuff
21. End sub
```

III. RESULTS Of EXPERMIENTATIONS

We run tests on a simulation of the model we've suggested. One, five, and 10 deduplicators are used in the experiments. The contents and attributes of every file utilised in the tests are random. The documents used for this experiment come in a variety of sizes: 100, 150, 200, 250, 300, 500, 800, 1 MB, and 2 MB are examples of data sizes. On ten files, one hundred files, one 1,000 documents, and ten thousand files, we evaluate upload, modify, and delete events. Each file's degree of QoS has been assigned randomly in order to test the shifting level of QoS. (1-5). Each file's level of redundancy is represented by a separate QoS number between 1 and 5. Higher level QoS files will receive more replication than lower level files.

DOI: 10050086.2022.12.05

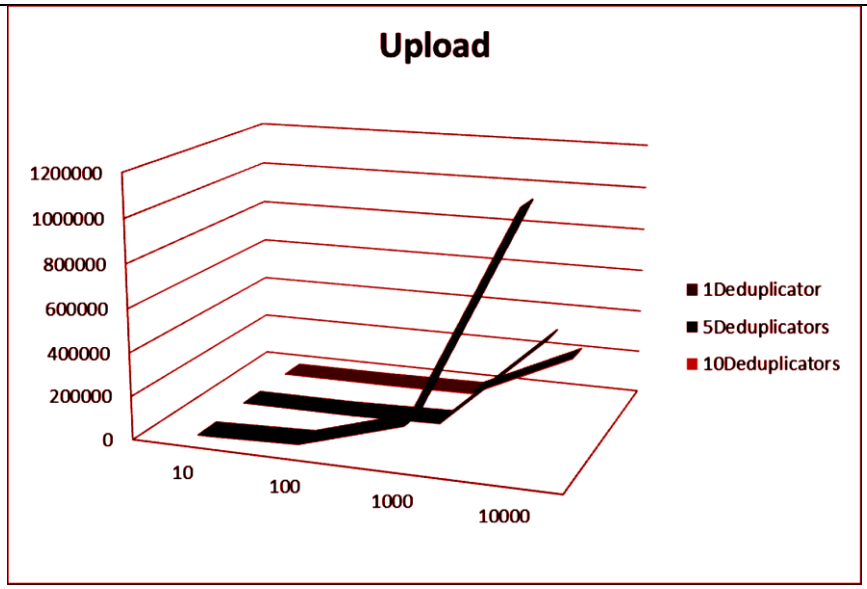

Figure 3: Experimental results (UPLOAD)

We run studies for the situation where the level of QoS changes after files have already been posted to the system, which implies the amount of copies of documents in the system may change based on the maximum level of QoS. Update files' results demonstrate that increasing the number of deduplicators can speed up processing whenever the quantity of files increases. Five deduplicators may reduce the processing time required by one deduplicator by 41.78%, 61.79%, 63.78%, and 75.25% when there are 10, 100, 1,000, and 10,000 files, respectively. Ten deduplicators could save even more work at 75.02%, 75.34%, 82.09%, and 96.17%.

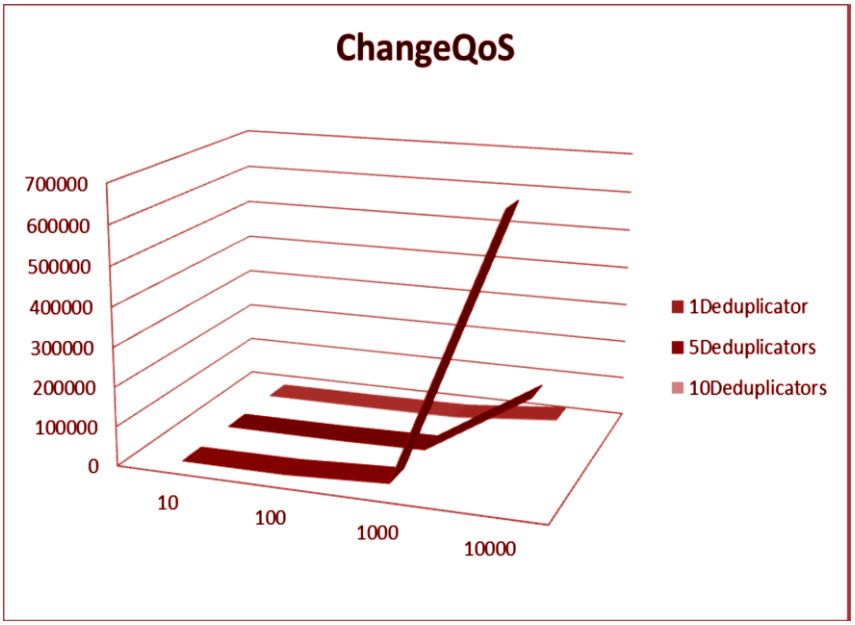

Figure 4: Experimental results (CHANGE QOS)

When there are ten, one hundred, or a thousand files, we discovered that the time saved with adding more deduplicators is not as great as the time saved by upload situations. In contrast to upload scenarios, the time savings by five and 10 deduplicators continue to grow as the number of files is increased to one million and ten thousand. To remove files, we do trials. Increasing the number of deduplicators can also speed up processing, however delete files provide significantly different outcomes from upload and update scenarios. According to the results, utilising 5 deduplicators can reduce the processing time required by one deduplicator by 93.42%, 69.31%, 40.74%, and 90.28%

DOI: 10050086.2022.12.05

when there are 10, 100, 1,000, and 10,000 files, respectively. Ten deduplicators can cut processing time by 98.68%, 90.59%, 85.87%, and 90.03%.

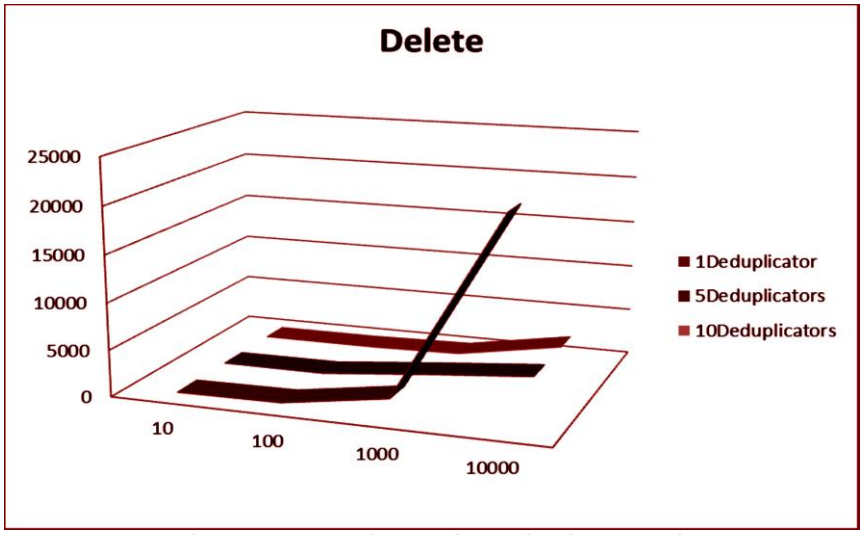

Figure 5: Experimental results (DELETE)

As demonstrated in Figure 2, when only one deduplicator is utilised, the system has scalability issues and becomes slower even as number of files grows. This is due to the fact that a solitary deduplicator cannot keep up with the enormous strain brought on by increasing requests and users. The results indicate that adding five and ten more deduplicators helps to shorten processing times.

Table I displays the findings. When there are between ten and one hundred upload files, utilising five deduplicators can cut processing time by between 85.75% and 94.20% and ten deduplicators can save even more time between 92.85% and 95.55%. Five or ten deduplicators still can help to speed up processing when there are 1,000 upload files, but their effectiveness drops to 91.40% and 95.58%, respectively. Whenever the amount of upload files is increased to 10,000, however, time savings become much less because ten and five deduplicators may respectively cut processing time by 60.10% and 79.71%.

TABLE I. Generating Data Proportion Between 5 and 10 deduplicators

Number of files	upload		update		delete	
10	96.86	101.96	52.89	86.13	104.53	109.79
100	105.31	108.66	72.8	86.45	70.42	101.6
1000	102.51	106.69	74.89	93.11	51.85	96.98
10000	106.69	80.82	86.36	107.28	101.39	101.14

IV. CONCLUSION

The use of fog storage solutions, which are offered via cloud computing, has grown in popularity. It provides on-demand virtualised capabilities and users just pay for the actual amount of room they utilise. Due to the increased need for cloud storage, data deduplication is one technique utilised to improve efficiency levels. The current data duplication techniques in cloud storage are static, which limits their capacity to fully address the changing nature of the information there. In this work, we provide a fluid deduplication technique for cloud storage to achieve a balance among rising efficiency levels and fault-tolerant needs, in addition to boost efficiency in cloud-based storage. We constantly change the number of duplicates of files based on the altering degree of QoS. Additionally, we want to monitor any changes in users' get the over time.

REFERENCES

1. Pooranian, Zahra, et al. "RARE: Defeating side channels based on data-deduplication in cloud storage." *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018.

DOI: 10050086.2022.12.05

2. Chhabra, Nipun, and Manju Bala. "A Comparative study of data deduplication strategies." *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018.
3. Barik, Rabindra K., et al. "GeoBD2: Geospatial big data deduplication scheme in fog assisted cloud computing environment." *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2021.
4. Reddy, B. Tirapathi, and MVP Chandra Sekhara Rao. "Filter based data deduplication in cloud storage using dynamic perfect hash functions." *International Journal of Simulation Systems, Science & Technology* (2018).
5. Shin, Hyungjune, et al. "Privacy-preserving and updatable block-level data deduplication in cloud storage services." *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 2018.
6. Yang, Xue, et al. "Achieving efficient and privacy-preserving multi-domain big data deduplication in cloud." *IEEE Transactions on Services Computing* 14.5 (2018): 1292-1305.
7. Yu, Chia-Mu, et al. "Privacy aware data deduplication for side channel in cloud storage." *IEEE Transactions on Cloud Computing* 8.2 (2018): 597-609.
8. Kaur, Ravneet, Inderveer Chana, and Jhilik Bhattacharya. "Data deduplication techniques for efficient cloud storage management: a systematic review." *The Journal of Supercomputing* 74.5 (2018): 2035-2085.
9. Singh, Priyanka, Nishant Agarwal, and Balasubramanian Raman. "Secure data deduplication using secret sharing schemes over cloud." *Future Generation Computer Systems* 88 (2018): 156-167.
10. Prajapati, Priteshkumar, and Parth Shah. "A review on secure data deduplication: Cloud storage security issue." *Journal of King Saud University-Computer and Information Sciences* (2020).
11. PG, Shynu, et al. "A secure data deduplication system for integrated cloud-edge networks." *Journal of Cloud Computing* 9.1 (2020): 1-12.
12. Zhang, Yinghui, et al. "Secure deduplication based on Rabin fingerprinting over wireless sensing data in cloud computing." *Security and Communication Networks* 2018 (2018).
13. Uma, G., and L. Jayasimman. "Enhanced convergent encryption key generation for secured data deduplication in cloud storage." *Journal of Physics: Conference Series*. Vol. 1142. No. 1. IOP Publishing, 2018.
14. Yuan, Haoran, et al. "DedupDUM: secure and scalable data deduplication with dynamic user management." *Information Sciences* 456 (2018): 159-173.
15. Noshy, Mostafa, Abdelhameed Ibrahim, and Hesham Arafat Ali. "Optimization of live virtual machine migration in cloud computing: A survey and future directions." *Journal of Network and Computer Applications* 110 (2018): 1-10

DOI: 10050086.2022.12.05