MEDWIN PUBLISHERS
Committed to Create Value for Researchers

Open Access Journal of Data Science & Artificial Intelligence
ISSN: 2996-671X

Authentication Scheme for Preventing Data Duplicate in Cloud Storage

Hammed M[1] and Nureni OA[2]*

Department of Statistics, Osun State University, Nigeria

***Corresponding author:** Nurain Olawale Adeboye (PhD), Department of Statistics, Faculty of Basic and Applied Sciences, College of Science, Engineering and Technology, Osun State University, Nigeria, Tel:+234-8033348141; Email: nureni.adeboye@uniosun.edu.ng / adeboye9olawale@gmail.com

Research Article
Volume 2 Issue 1
Received Date: July 19, 2024
Published Date: August 06, 2024
DOI: 10.23880/oajda-16000137

Abstract

Cloud computing is an important tools for modern business settings as a computing resource which shares and delivers over the internet and it is also based on customers' demand. Cloud storage is one of the utility services provided in a cloud computing environment by cloud service vendors to serve as a storage pool and for sharing resources. Cloud storage reduces customers' expenses for pursuing and maintaining the storage infrastructure, only the amount of storage requested can be scaled up and down in accordance with customers' demands. However, the large data size of cloud computing necessitated a reduction of data volumes which assisted cloud computing service providers in reducing the cost of maintaining large cloud storage and saving energy consumption. Deduplication methods with different concepts such as checksum, encryption techniques biometric systems etc. have been proposed by different studies. However, the literature revealed that some of the techniques used for deduplication were not adequate for the process due to different weaknesses which some of the examined studies failed to identify as a part of challenges. An authentication scheme was proposed in this study which used an Apriori associative rules mining algorithm augmented with optimization techniques that improved the processing speed of the Apriori algorithm. The method properly compared input data and the stored data to detect duplicates, thus, it enhanced and maintained storage efficiency in cloud storage.

Keywords: Data Duplication; Deduplication; Authentication Scheme; Apriori Algorithm; Breadth First-Search Algorithm

Abbreviations

IT: Information Technology; IIM: Inverted Index Method; T: Tag; L: label.

Introduction

Cloud computing is one of the new emerging technologies which has been helping several organizations to save their money and time, it also provides convenience to the cloud end users. Cloud provides many benefits in terms of cost and on demand services and real time communication [1,2]. Cloud computing provides online information technology (IT) resources which ensure the results desired by the user at any time. It allows users to borrow cloud computing resources such as server, software and storage [3]. Cloud storage is one of the cloud computing resources which provides virtual storage to cloud users based on demand and it is also used for different purposes depend on the users. Some use cloud storage as backup purposes while some users use data storage to substantiate their applications. There are some users that move their archive storage to the cloud where

more storage capacity can be achieved at lower cost, instead of wasting fund on acquiring additional physical storage. The cloud storage has becoming organizations' utility where end users can virtually store their data without bothering the entire mechanism. The increasing usage of internet and data availability and accessibility on social media continue to exert influences on very large-scale data in everyday activities. Many cloud vendors such as Netflix, Amazon and Flipkart often perform data collection, data mining, and data analysis from different sources and allow sharing of large volumes of analyzed data over the internet, this data can be accessible through cloud storage [4]. Today, the amount of data in the cloud storage continues increasing to become large volume of data and all cloud users are expected to reach the cloud services at any time irrespective of their location. Due to increasing amount of stored data in the cloud storage, the data disks are unable to recognize duplicate data that appear on the disk. Data duplicates affect storage space of the disk and more of data duplicates affect the performance and usage of the disk's speed and space [5]. When cloud storage devices are occupied with large amount of duplicate data, the deduplication techniques which are capable of detecting and removing any data of the same type or the data that are of the same content are however, needed [4].

Deduplication is the process that enhances cloud storage efficiency which aims to manage storage space for only one replicate of the data in the storage (SNIA). However, aiming to address the data duplication challenges, many studies have proposed different deduplication methods and the literature revealed that some of the studies focus on hash functions such as Inverted Index Method and cryptographic algorithms such as AES, DES and Blowfish etc. to secure files over cloud. But notwithstanding, deduplication with user authentication has not been given much attention as numbers of studies have proposed different techniques to reduce or eliminate data duplication in cloud storage. This study used an Apriori algorithm augmented with optimization techniques to authenticate and validate every data that were stored in the cloud storage. The technique used in this study guaranteed that every data in the cloud storage (datacenter) has no duplicates. That is, the authentication method used eliminates set of duplicates data that might come from either adversaries or even come by error with the intension to occupy the storage space with redundancy data. The method used also keeps only unique and essential data, thus it is significantly clearing the storage space for incoming data. The optimization and breadth first-search techniques used enhanced the system throughput when unnecessary processing that might come through Apriori algorithm was avoided.

Large amount of cloud storage infrastructures is filled with duplicate data records; however, aiming to address the data duplication issues, literature revealed that many studies have proposed different techniques. Many studies have focused on techniques for deduplication using biometric user authentication while many studies focused on cryptographic and hash function techniques to authentication user so as to remove duplicate data in the cloud storage. Literature revealed that some of the studies that used cryptography techniques for deduplication were not practically implemented for authentication scheme and while some of the techniques are weak due to information leaks. Public-key cryptography is exposed to attacks such as phishing, spoofing, shoulder surfing and other attacks. In the situation of exposure to attacks, a malicious third party intercepts a public key when it is sent to one of the parties involved. Moreover, other problems of using public-key encryption system for deduplication is the processing speed, the encryption and decryption processes which involved in cryptography creates non-scalability of the method. Achieving the system throughput which is one of the metrics of the deduplication might be difficult. Among the studies that were reviewed include the works of Assam A, et. al. [6]; Wong and Kim [7]; Yujuan T, et al. [8]; Maharasi M, et al. [9]; Aghili H [10]; Akhila K, et al. [11]; Maragatharajan and Prequiet [12]; Yang Z, et al. [13]; Ryan NSW, et al. [14]; Qinlu and Bilin [15]; AbdulSalam and Assmaa [16].

Assam A, et al. [6] examined different authentication techniques in cloud environments. But it was observed that some of the deduplication using simple password-based authentication lacks positive recognition as far as cloud data deduplication is concerned. However, there is need for a strong deduplication with authentication scheme to prevent data duplication in the cloud storage. This study used associative data mining (Apriori) algorithm which capable of comparing input and stored data ($X \cup Y$) whether they appear together in the data storage to detect duplicate using S (for support) and C (for confidence). The study also used an optimization technique to avoid unnecessary processing which may cause delay when computing support.

Wong and Kim [7] proposed a concept of biometric-based authentication scheme in cloud computing system, the study discussed the challenges and other limitations of traditional methods alongside with different attacks scenarios, such as tracking of an individual to leak his/her confidential data, misuse of biometric data etc. The study further explained that the privacy of cloud-based biometric authentication might not capable to resolve some technical challenges of authentication scheme cloud system. But, the Apriori algorithm proposed in this study used associative rules to discover patterns in the datasets which systematically detect duplication between input data and the stored data when they were compared. Results include finding the relationship among the data items with the aid of breadth first-search

algorithm which improved the performance of Apriori to quickly remove any duplicate data in the cloud storage.

Yujuan T, et al. [8] proposed a system which its architecture consists of three-tier systems: file agent, master server agent and storage server agent. Whenever the clients sign up for substitute services, then file agents will be installed on the clients' machines, while service providers produce master server agent and storage server agent in cloud data center to service the substitute services request from the clients. But, cost and time to transfer file (transfer overhead) from one server to another is one of the challenges of their architecture which composed three subsystems. An optimization technique was used to augment the Apriori algorithm to eliminate unnecessary processing which can cause delay when searching for item sets and computing the support.

Maharasi M, et al. [9] proposed attribute-based encryption system for removal of data duplicates in the cloud-based storage. When a file storage request is sent, each data provider firstly creates a tag (T) and a label (L) connected with the data, and then the data is encrypted underneath of an access structure over a set of attributes. But, large number of tags and labels constituted system difficult when retrieving data. However, the ABE systems do not support secure deduplication which leads to the application of the system to be more expensive for some business storage services. Moreover, the technique is not scalable for large amount of data. The Apriori augmented with optimization technique used in this study is a simple association technique with feasible implementation which removes duplicates in the cloud storage. It is less expensive which is easy to be deployed for any business storage services. More so, the system used in this study achieved scalability when large amount of data were involved.

Aghili H [10] did a comparative analysis of symmetric algorithms like DES, triple-DES, AES and Blowfish. The analysis proved that Blowfish takes less time for execution. The author used Blowfish for encryption, applied IBM deduplication service on JPG image, and it was concluded that blowfish is a best suitable algorithm with respect to security and the algorithm processing. The weaknesses of blowfish are that the encryption key may get to the receiver through unsecured transmission channel. In blowfish encryption algorithm each user must have a unique key, so as number of users increasing, to manage the key also becoming very difficult. Blowfish decryption process is also times consuming and processing speed is very slow. An optimization technique was used to augment the Apriori algorithm to eliminate unnecessary processing which can cause delay when searching for item sets and computing the support.

Akhila K, et al. [11] examined various deduplication techniques such as ClouDedup, DupLESS, HEDup, and SecDup etc. and most of the algorithms used were based on convergent key method. One of the drawbacks of convergent key is compromising of the key which hindered the quality of the algorithms. The Apriori augmented with optimization technique used in this study is a simple association techniques with feasible implementation which removes duplicates in the cloud storage. It is less expensive which can also be deployed for any business storage services.

Maragatharajan and Prequiet [12], proposed encrypted data using proxy re-encryption method. The user chooses random number to constitute the symmetric key and the encryption key is communicated to the cloud service provider to check for duplicate data with the help of tokens. If any duplicate file is found, it will be communicated to an authorized party. The scheme allows uploading and downloading of files, checking for duplicates, and deleting the duplicate files. This scheme provides security and privacy of files. But, the drawback of the method is that when it checks and duplicate files were found, this will be communicated to an authorized party. However, the author failed to realize that information communicating to the authorized party can leaks to an unauthorized third party who may compromise the information before getting to an authorized party. The Apriori algorithm used aims to discover patterns which technically detect duplication between input data and the stored data when they were compared. Finding the pattern between input data and the stored data helped the system to quickly remove any duplicate data.

Yang Z, et al. [13] proposed a method which is called droplet strips input data streams onto multiple storage servers, the method capable to limit number of stored data clusters on each server and also ensure that fingerprint index is fitted into the system memory. It was revealed that buffering layer in droplet performs better for only small data clusters and hash function crashes is likely an issue with droplet deduplication. It was also observe that if fingerprint is compromised, it is not possible for the fingerprint identifiers to be replaced. The association augmented with optimization that was used in this study enhanced the system searching for duplicates process and the distribution and finding relationship among data items is not differ from one another.

Ryan NSW, et al. [14] proposed a rapid asymmetric maximum. The method uses bytes value to declare the threshold instead of using hashes. The technique utilizes a fix-sized and a variable-sized window to find a maximum-valued byte which is the point of entry for the system to detect and prevent duplicate. But the drawback is that it likely delivers more metadata, and the process will make the system slow. An optimization technique was used to augment

the Apriori algorithm to eliminate unnecessary processing which can cause delay when searching for data item sets and computing the support to the relationship between input data and the stored data.

Qinlu and Bilin [15] proposed RFD-HDFS and FD-HDFS which was based on hash function, the method proved that both RFD-HDFS and FD-HDFS framework not only implement deduplication function but also effectively reduces the disk utilization of duplicate files. But, one of the drawbacks is that the applications require hash function and probability of hash colliding in SHA cannot be overlooked. Therefore, hash crashes is likely to be an issue when using the method for deduplication because in a situation where a bit of information gets hash number that contrasted with the record of existing hash numbers, the bit of information is viewed as a duplicate. The Apriori algorithm used discovered patterns which technically detect duplication between input data and the stored data when they were compared. Finding the pattern helped the system to quickly remove any duplicate data.

AbdulSalam and Assmaa [16] proposed content-based TTTD chunk algorithm as a deduplication system that was decomposed into three stages which include chunking, hashing and indexing, and matching. The hashing and indexing stage computes the hash value for the whole chunk and the value is added to the index table. The technique computes the hash values for each chunk to search for the similarity of the chunks using a file name or file type in the dataset. The literature revealed that the process is a time consuming to complete the matching operations. Furthermore, the content-based TTTD used the histogram to represent the chunk of dataset while using statistical distribution to representing the chunks in one dataset may differ from one another. The association augmented with optimization used in this study enhanced the searching process and the distribution of data items will not differ from one another.

Materials and Methods

In this study, let X denotes input data while Y denotes the stored data. The association rule $X \to Y$ is interpreted as a set of data that satisfies the conditions in X which are also likely to satisfy the conditions in Y. This shows that X and Y are conjunctions of attribute value-pairs and S (for support) is the probability that X and Y appear together in data storage and C (for confidence) is the conditional probability that X appears in data storage when Y is present. The association rule used in the study is an If-THEN rule which uses two measures to quantify the support and confidence of the rule to find the relationship between input data and the stored data to predict whether the input data has occurred in the data storage. The association rule used in the study uses inference engine to make decision based on support and used the measure's engine to ascertain the reliability of the decision made. The association rule detects duplication when input data and the stored data were compared, the conditions in input data must satisfy the conditions in stored data., the input data and the stored data are conjunctions of attribute value-pairs where S (for support) indicates that input data and the stored data appear together in the data storage ($X \cup Y$). Both equation 1 and 2 and the algorithm 1 were used to model the association rule for support which found relationship between the input data and the stored data.

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y) \quad (1)$$

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N} \quad (2)$$

Algorithm 1

Step1: $X \subseteq \sum, Y \subseteq \sum$

Step2: $X \cap Y = \varnothing$

Step3: $\exists I_i \in X, 1 \leq i \leq n$

Step4: $\exists I_j \in Y, 1 \leq j \leq n$

The C (for confidence) which indicates that conditionally X appears in the data storage when Y is present and the rule shows the level of correlation between an input data and the stored data (X and Y) in the data storage. If the C (for confidence) is low, this indicates that the input data is already occurred in the data center storage but, if the C (for confidence) is high, it indicates that the input data has not occurred in the data storage. The equation 3 modelled confidence by dividing the support for whether the input data has duplicate in the data storage using correlation for measurement.

$$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \quad (3)$$

The equation 4 depicts correlation measurement between input data and the data stored to determine whether the C (for confidence) is high or low before decision is made that input data has occurred or not occurred in the data storage.

$$C(x, y) = \frac{\sum_{i=1}^{m}(x_i - \mu_i)(y_i - \mu_i)}{\sqrt{\sum_{i=1}^{m}(x_i - \mu_i \sum_{i=1}^{m}(y_i - \mu_i)^2}} \quad (4)$$

Decision Theory for Apriori Support Strategy

The decision is made based on support strategy at inference engine; thus, the decision was strictly made based on support strategy which finding the relationship between input data and the data stored in the cloud storage or data center (i.e., $X \cup Y$). The number of characters and the

nature of data were used as metrics to determine whether the confidence C is low or high, that is duplication occurred or it has not occurred in the data storage. An apriori rule S which support strategy subdivides input data set into a set of subsets ($x_1,x_2.....,x_n$) where each set of subsets are correspond to parent input data . The number of characters in which belong to class of stored data is $X \cup Y$, thus, the relationship determine whether the confidence is low or high which shows whether the duplicates are detected or not.

Reliability Measurement of Decision based on Apriori Confidence

This study used a model to determine the reliability of decision made based on confidence C which first compares the relative advantages of the metrics that determined whether the C confidence is low or high; that is, duplication has occurred or not occurred respectively. Proving these two results, we considered the dual model using sigma notation in equation 5, 6 and 7. The architecture in Figure 1 depicts the logical flow of information in the system.

Minimize

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad (5)$$

Subject to the constraints

$$\sum_{j=1}^{n} x_{ij} = a_i, \ \forall i = 1,2,...m \ \text{(for confidence is low)} \ (6)$$

$$\sum_{i=1}^{m} c_{ij} = b_j, \ \forall j = 1,2,...n \ \text{(for confidence is high)} \ (7)$$

$$x_{ij} \geq 0 \ \text{for i and j}$$

Minimizing Time Complexity for Computing Apriori Algorithm Support

There must be a considerable time taken for an Apriori algorithm to computing support for a given system to avoid unnecessary processing which delays the computational time of the algorithm. This study used an optimization technique to modeling the computational time of the Apriori algorithm. The techniques enhanced the processing time of the algorithm when computing support for the conjunctions of attribute value-pairs between the input data and the stored data. The optimization technique used eliminates unnecessary searches and computation for support S is faster. Thus, for the determination of the support of z item sets in the data storage we have the time complexity of equation 8.

$$T = 3\tau\sum_{j=1}^{d}\ \sum_{i=1}^{n} x_j^{(i)} z_j \ (8)$$

Thus, equation 10 modelled the total time for the determination of the support of all az items to be

$$T = 3\tau\sum_{k}\ \sum_{s=1}^{m_k}\ \sum_{j=1}^{d}\ \sum_{i=1}^{n} x_j^{(i)} z_j^{(s,k)} \ (9)$$

The Apriori algorithm visits the general storage which contains item of datasets in a level-wise style to increase the processing time of the algorithm as it shown in the algorithm 2.

Algorithm 2

Step1: $C_1 = A(X)$ is the set of all one-item sets, k = 1
Step2: while Ck ≠ ∅ **do**
Step 3: scan database to determine support of all a_y with y ∈ C_k
Step4: extract frequent item sets from C_k into L_k
Step5: generate C_k+1
Step6: k:= k + 1.
Step7: end while

Apriori Augmented with Breadth-first search Techniques

Breadth-first search algorithm that was used in the study allows uncomplicated method of duplicates detection system when Apriori is searching for duplicate data in the cloud storage. The breadth-first search algorithm starts at a root data item and critically examined all the neighbouring data items and then examines each of those neighbour data items around the axis. It examines the neighbouring data items that have not been visited, and continue the process until all data items are visited. Each time a breadth-first search algorithm examines the root data item, it checks whether data item is a duplicate of neighbouring data items. In addition, it checks the neighbouring data items that have not been visited whether they are duplicates of visited data items. The procedure for breadth-first search algorithm is shown in algorithm 3.

Algorithm 3: Breadth-first search Procedure for augmenting Apriori Rule

Step 1: *Create a set of data item-list and set it as initial state.*
Step 2: *Repeat until the goal state is found or data item-list is empty.*

a. *Remove the first element, say Z, from the data item-list, if data item-list was empty **then** stop.*
b. *For each way that each rule can match the state described in Z **do**:*

1. *Apply the rule to generate a new state.*
2. *If the new state is the goal state, stop and return this state.*
3. *Otherwise add this state to the end of data item-list.*

This approach drastically enhanced the performance of Apriori when searching for duplicates in the datacenter, thus unnecessary delay that may hindered the performance of Apriori was reduced. The system architecture for deduplication is shown in Figure 1.

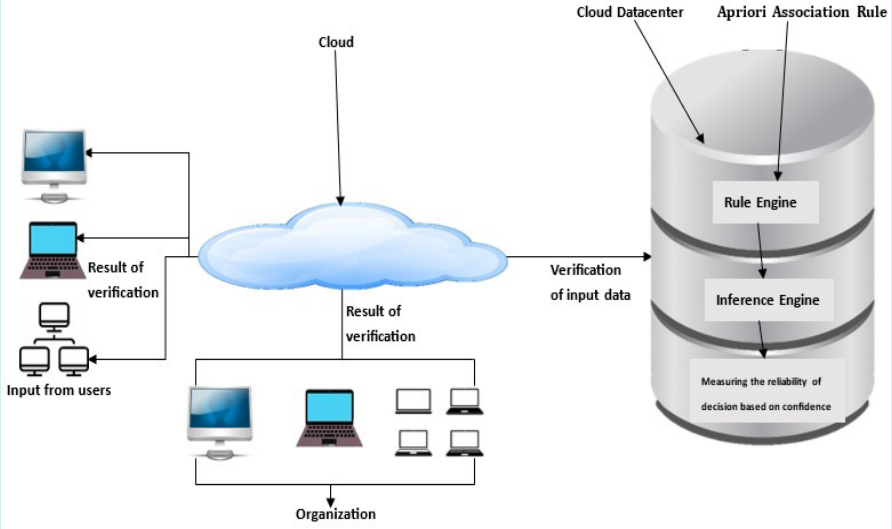

Figure1: Deduplication System Architecture.

Discussion of Results

Usage of system across enterprise might be varied and if a storage system accommodates multiple virtual machines for different users, a deduplication system should provide significant storage savings. A deduplication system in this study was implemented and tested at Federal Polytechnic, Ilaro. Authentication scheme for data deduplication scheme using Apriori augmented with optimization techniques was implemented in JavaScript on the portable windows-based systems. The study focused on the usage of Apriori algorithm to find the relationship among the data items to detect duplicate and a single edge server was used as an external storage system. Implementation was done on a single storage connector which provides the software functionality through a directory on the disks and also manages external storage systems which includes databases and servers. The JavaScript file system library which was used can be deployed on any file system compatible with the library. The system is flexible to detect duplicates in cloud system and as well as capable of detecting duplicates in any local storage. The same two files were copied into the software directory and the system detected that the two files were the same as it is shown in the Figure 2. As the system used machine learning approach, it did not rely on dataset like other different machine learning approaches which have been used by other studies but it mapped the incoming data with stored data to check for duplicate.

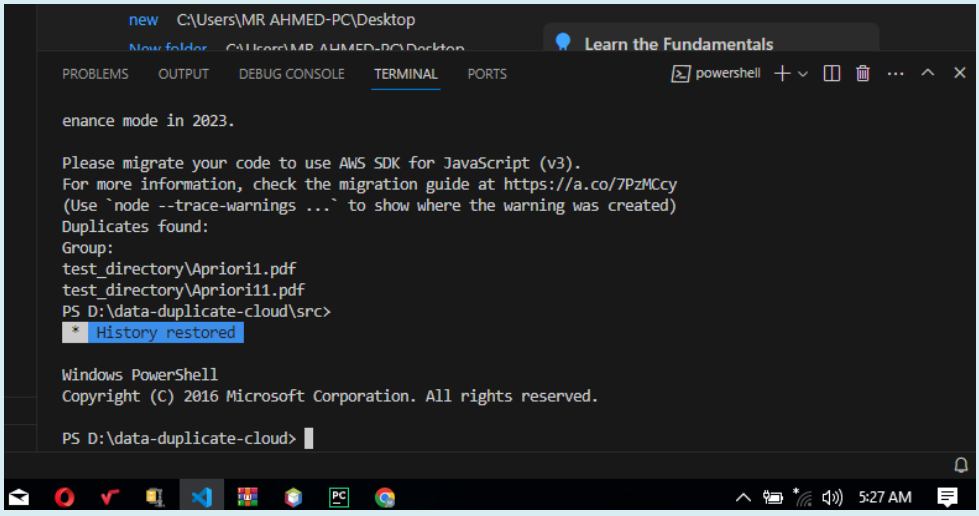

Figure 2: Deduplication Implementation.

Conclusion

Deduplication is a promising technique to optimize storage system. It can greatly minimize the amount of data, energy consumption in the data center, and thereby reduce drastically the storage cost. Based on the extensive review of deduplication techniques, the literature revealed that authentication scheme using Apriori algorithm which was used in this study is a powerful tool for deduplication processes. Ability of the Apriori to compare and finding the relationship between all incoming data and stored data has strengthened the system to remove all duplicate data for the large cloud data center.

References

1. Leesakul W, Townend P, Xu J (2014) Dynamic Data Deduplication in Cloud Storage. Proceedings IEEE 8th International Symposium on Service Oriented System Engineering pp: 320-325.
2. Shobana R, Shantha KS, Leelavathy S, Sridevi V (2016) De-Duplication of Data in Cloud. International Journal of Chem. Science 14(4): 2934-2938.
3. Hakjun L, Dongwoo K, Youngsook L, Dongho W (2021) Secure Three-Factor Anonymous User Authentication Scheme for Cloud Computing Environment. Hindawi Wireless Communications and Mobile Computing pp: 1-20.
4. Vinoth KM, Venkatachalam K, Prabu P, Abdulwahab A, Mohamed A (2021) Secure biometric authentication with de-duplication on distributed cloud storage. Peer J Computer Science 7: e569.
5. Ajahar IP, Liladhar MK, Rijavan AS, Dheeraj BS (2018) Removing Duplicate Data in Cloud Environment using Secure Inverted Index Method. International Research Journal of Engineering and Technology 5(9): 157-161.
6. Assam H, Hassan W, Zeadally S (2019) Automated Biometric Authentication with Cloud Computing. In: Obaidat M, et al. (Eds.). Biometric-based physical and cybersecurity systems. Cham pp: 455-475.
7. Wong KS, Kim MH (2012) Secure Biometric-based Authentication for Cloud Computing. In: Ivanov I, et al. (Eds.). Cloud Computing and Services Science. CLOSER 2012. Communications in Computer and Information Science pp: 367.
8. Yujuan T, Hong J, Dan F, Lei T, Zhichao Y, et al. (2010) SAM: A Semantic-Aware Multi-tiered Source De-duplication Framework for Cloud Backup. In Parallel Processing (ICPP), 2010 39th International Conference pp: 614-623.
9. Maharasi M, Keerthiga S, Kiruthika P, Nivetha R, Priya S (2018) Removal of Duplicate Storage of Encrypted Data in Cloud Computing Environment. International Journal of Engineering Research in Computer Science and Engineering 5(3): 643-647.
10. Aghili H (2019) Improving Security Using Blow Fish Algorithm on Deduplication Cloud Storage, Fundam. Res Electr Eng 480: 723-731.
11. Akhila K, Ganesh A, Sunitha C (2016) A Study on Deduplication Techniques over Encrypted Data. In Procedia Computer Science 87: 38-43.
12. Maragatharajan M, Prequiet L (2017) Removal of duplicate data from encrypted cloud storage. Proceedings of the 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS pp: 1-5.
13. Yang Z, Yongwei W, Guangwen Y (2012) Droplet: A Distributed Solution of Data Deduplication. ACM/IEEE 13th International Conference on Grid Computing pp: 114-121.
14. Ryan NSW, Hyotaek L, Mohammed A (2017) A new Content-Defined Chunking Algorithm for Data Deduplication in Cloud Storage. Future Generation Computer Systems 71: 145-156.
15. Qinlu HG, Genqing B, Bilin S, Weiqi Z (2020) Data Deduplication Technology for Cloud Storage. Technical Gazette 5: 1444-1451.
16. Abdulsalam HJ, Assmaa FA (2018) New Techniques to Enhance Data Deduplication using Content based-TTTD Chunking Algorithm. International Journal of Advanced Computer Science and Applications 9(5): 116-121.