DOI: https://doi.org/10.46632/daai/2/1/2

Data Analytics and Artificial Intelligence
Vol: 2(1), 2022
REST Publisher; ISBN: 978-81-948459-4-2
Website: http://restpublisher.com/book-series/data-analytics-and-artificial-intelligence/

Secure Data Deduplication and Data Portability in Distributed Cloud Server Using Hash Chaining and Lf-Wdo

***A R Athira**
Vinayaka Mission Research Foundation, Salem
*Corresponding author Email: parvathyretnakaran@gmail.com

Abstract: An application of Cloud Computing (CC) technologies for interconnecting data as well as applications served as of manifold geographic locations is the distributed cloud. For distributed CC, managing a vast measure of data along with maintaining security are vital aspects. Though lots of techniques were introduced recently to manage and protect data on the distributed cloud, those have not rendered any desirable security. Thus, this paper proposes a secure Data Deduplication (DD) and Data Portability (DP) for a distributed cloud environment. To enhance the user data's security, primarily, the inputted files are bifurcated into blocks and after that, an appropriate Cloud Server (CS) is chosen utilizing Hybrid Forest Genetic Algorithm (HFGA) centred upon split file features together with CS features. In a DD phase, the Whirlpool algorithm converts the split data into Hash Code (HC), and then, the hash chaining technique securely removes the duplicated data. Then, that split data is compressed, encrypted, together with amassed in the selected CS. Next, the Levy Flight – Wind Driven Optimization (LF-WDO) performs the DP to enhance the cloud data's security. Investigational outcomes exhibit the proposed work's potential.

Keywords: Distributed Cloud Server, Data Deduplication, Portability, Hybrid Forest Genetic Algorithm (HFGA), Levy Flight – Wind Driven Optimization (LF-WDO) Algorithm, Whirlpool Algorithm, cloud data, security.

1. Introduction

CC has the advantage of the flexibility and economic savings, which in turn motivate the companies and also individuals to farm out their data to the CS [1]. It is pressing to develop new techniques to utilize the storage space along with network bandwidth effectively due to the augmenting data volumes stored on the CS [2]. Because of this, the DD has received much interest as of academia along with the industry. Alternatively, DD techniques utilize data similarity (file-level/block level) for identifying the same data and reducing the storage space by means of storing just one single copy in the CS [3]. Cross-user deduplication is employed practically to maximize the DD benefits [4, 5]. It identifies redundant data amongst disparate users, and then, eradicates the redundancy and saves a solitary print of the unique data [6, 7]. Therefore, DD offers a unique chance to cloud storage providers to render their users with more space at a lower expense [8]. Nevertheless, the requirement for secure DD was significantly augmented for particular security goals in varied cloud application situations since data security is becoming the utmost important need for CC services [9]. Besides DD, DP is as well useful for securing data transmission. The competence to move around without any change is termed the DP [10]. Cloud portability is the procedure of moving applications along with its associated data as of one cloud supplier to another with less disruption in addition to minimal downtime [11, 12] and also encompasses the transfer of data betwixt public and private cloud environments. DP mostly enables the cloud data constituent reuse and also import-export functionality of data services [13, 14]. Lately, many researches were performed on secure Data Storage (DS) in a Distributed Cloud Server (DCS); however, those researches are not duly enough in this big data together with a DCS era. Designing secure DS aimed at the DCS is an emerging research topic. Here, a novel framework wherein data privacy on the cloud is preserved and the cloud storage supplier that performs DD devoid of compromising data privacy along with security with a DCS is proposed. The proposed work is broken into '3' stages: the selection of DCS, DD, and DP. This paper is set as: Works that are related to the proposed method are conferred in Section 2. The proposed work in tandem with its modules is elucidated meticulously in Section 3. A general discussion on the results centred on the proposed technique can well be found in Section 4. Lastly, Conclusions together with future work's discussion is exhibited in Section 5.

2. Literature Survey

Vengala et al. [15] generated a secure data transmission scheme utilizing a DCS and DD. The Hybrid Meerkatclan algorithm picked the CS that was optimally grounded on the features. Subsequently, the inputted user data's DD was carried out utilizing the SHA512. Following DD, the inputted file was compressed as well as encrypted with a '2'-stage lempel–ziv along with optimized CP-ABE-ECCs. Finally, the encrypted file was amassed in the selected CS. The outcome carried out to be fine, however, the scheme was not attentive to the split file size and also it can't check if it was probable to amass the split data into the DCS. Yibin et al. [16] concentrated on Security-Aware Effective Distributed Storage that chiefly supported by means of the algorithms, comprising Alternative Data Distributions, Secure Efficient Data Distributions, along with Conflation. The approach split the file as well as stored separately the data on the DCS. The experimentation outcomes exhibited that the approach could efficiently be needed with a satisfactory computation time for DS. However, it brought

about more storage of data because of more facets of the file along with DCS. Hua me et al. [17] produced a randomized client-side DD scheme to lessen the redundant data to construct secure cloud storage. It was centred upon a randomized DD protocol to avert the collusive authentication attack as well as offline brute-force attack commenced via the outside adversaries and amassed every data as per '2' file tags to oppose duplicate-faking attack. Additionally, it realized a more accessible ownership management in addition to data sharing utilizing dynamic Key-Encrypt Key tree. Albeit the scheme attained secure DS, this sort of DD brought about a duplicate-fake attack. S. Beulah et al. [18] perceived similar data with a light-weight procedure with pattern analysis in addition to matching. The distributed encoding was employed to inflict end-to-end security intended for the produced data with lessened communication over-head. The data attained in the processing server were decoded, examined, together with matched with patterns intended for the exclusion of similar data as well as duplicated data. Therefore, the outcomes carried out to be safer with light-weighted procedures; however, the duplicated along with similar data weren't detected effectively via the inline procedure prior the data entered into the storage. Liang et al. [19] rendered a secure DS as well as recovery system by enhancing decent ration, tampering-proof, instantaneous monitoring, along with management of storage systems; as such devise supported the dynamic storage, quick repair, in addition to update of distributed data in the DS. Centred upon the exclusive chain storage structure, the distributed code not merely mended the close by local regenerative codes in the block chain but in addition lessened the resource overhead in the industrial nodes' DS. Investigational outcomes exhibited that it achieved good Data Security. However, the technique was susceptible to the collusive substantiation along with brutes-force attack. Tchernykh et al. [20] suggested multiple-cloud storage design termed WA-MRC-RRNS that united the weight access system, thresholds secret sharing, along with superfluous residue number system with manifold failure detection or recovery mechanisms, together with homomorphic ciphers. It utilized a multiple-objective optimization to regulate redundancy, data loss likelihood along with encryption-decryption speed. Outcomes exhibited that the approach rendered a safe way to lessen DS by lessened redundant data, data loss, et cetera. However, it underwent insecure DS because of improper features extracted as of the file as well as a cloud.

3. Secure Data Deduplication and Data Portability In Distributed Cloud Server

Distributed CC proffers numerous resources to users as services, for instance, extremely accessible storage space. Managing the escalating quantity of data and averting the cloud data as of the attacks is the utmost demanding task in a DCS. This paper developed a secure DD and DP system for a distributed cloud environment to manage a large quantity of data and also enhance cloud data security. The DD method makes data management scalable in CC. Conversely, the DP system enhances data security by means of transferring data as of one cloud service to another or betwixt DCS. To achieve effective DS, the assortment of a DCS process is integrated with the proposed scheme. The proposed scheme's working flow is exhibited in figure 1.

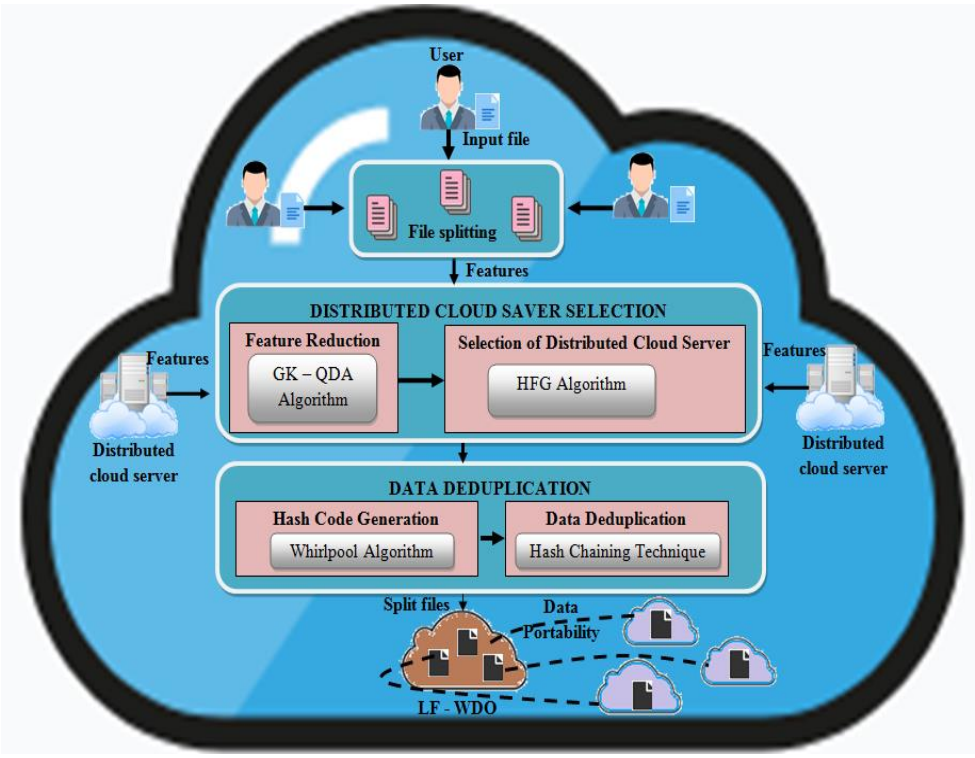


FIGURE 1. Working Flow of the Proposed Methodology

Input file splitting primarily, the inputted user files are bifurcated into blocks. This procedure is utilized to guard the user files as of data transmission attacks and also ameliorate the user files' security level. If the file is bifurcated into blocks, the

attackers can't access the complete file. Regard F_i is the inputted user files along with the split file blocks are mathematically expressed as,

$$(F_b)_i = \{B_1, B_2, ..., B_n\} \quad (1)$$

Wherein, $(F_b)_i$ alludes to the split blocks of the inputted file and B_n signifies the n-th block of $(F_b)_i$.

Distributed cloud server (dcs) distributed cc generalizes the cc design to position, process, along with the serve data and applications as of geographically distributed sites to fulfil requirements for performance, redundancy, along with regulations. Whilst amassing the data in the dcs, it is vital to choose an appropriate cs for the data to enhance the system's performance. The proposed work carries out feature extraction as well as feature reduction (fr) for choosing the appropriate cs. Feature extraction here, the significant features of the inputted file and the cs features are extracted. The features, say task cost, weight, speed, in addition to file size, are extracted as of the split file. Similarly, the resource features, say disk space, cloud memory, together with bandwidth, are as well extracted as of the disparate cs. The features are going through fr to attain effectual cs selection with minimal process time. Feature reduction fr is the procedure of lessening the number of features in computation devoid of losing vital information. The proposed work utilizes a Gaussian kernel-centred qda (gk-qda) to lessen the features. The conventional qda has a drawback of data loss. The Gaussian kernel method is integrated with qda to diminish the loss of data. Let $E(f_{uv}) = [E(f_{11}), E(f_{12}), ...E(f_{ww})]$ be the extracted features on kernel space R^S in addition to M be the kernel matrix, explicitly, $\mathrm{M} = (\mathrm{M}_{mn})_{m=1,....,w}^{n=1,....,w}$, M_{mn} implies a $w_m \times w_n$ sub-matrix of M.

$$\mathrm{M}_{mn} \to (\mathcal{G}_{uv})_{u=1,....,w_m}^{v=1,....,w_n} = \mathcal{G}_{uv}\left(E(f_{mu}), E(f_{nv})\right) \quad (2)$$

Where, $\mathcal{G}(\bullet)$ signifies the Gaussians kernel function. The gk-qda centred fr function is implied as,

$$R(f_{uv}) \to \tilde{Q}^T E(f_{uv}) = \Lambda_y^{-1/2} \vec{\mathrm{N}}_{yE_v}^T \Phi_y^T E(f_{uv}) \quad (3)$$

where, $R(f_{uv})$ denotes the dimension reduced features, $\tilde{Q}^T$ is the transformed matrix, Λ_y signifies the eigen values' diagonal, $\vec{\mathrm{N}}_{yE_v}^T$ is the transformed m-number of eigen vector set and Φ_y denotes the diagonal of the feature's mean value. As of this stage, the dimension reduced features $R(f_{uv})$ are attained, which is rendered to the cs selection phase.

Selection of distributed cloud server here, the appropriate cs for the split file is chosen utilizing the hybrid forest genetic algorithm (hfga) centred on lessened features. The fga is enthused by the procedure of a few trees in the forests. The fga is facing some downsides, such as low precision of optimization, slow convergence in the later phase, in addition to effortless to fall into a local optimum. To trounce such issues of fga, the proposed work hybrid the genetics algorithm to the fga. The hfga steps are elucidated as,

(i) Initialize the forest by randomly generating the trees. In each tree, initialize a variable with '0' or '1' and assign the "age" to 0. Select the optimal solution at random at the first iteration.

(ii) Employ the local seeding (ls) to each tree. Choose some random variables utilizing ls-change parameters and vary them from 0 to 1 or 1 to 0.

(iii) Neglect the 2 series of trees with "age" greater than the lifetime parameter and the area limit parameter and develop a candidate solution.

(iv) Select the trees from a candidate solution cantered on the tree's transfer rate utilizing ls-changes. Now, vary each selected variable from 0 to 1 or 1 to 0. Then, include the new solution in the candidate solution.

(v) Implement the crossover as well as mutation operators, which choose the best solution and neglect the worst solution for effective selection.

(vi) Sort the solution utilizing fitness value (fv), and the solution bearing the topmost fv is concerned as an optimal solution. Till the stop criterion is satisfied, perform all the aforesaid steps iteratively.

During cs selection, the trees are concerned as css. Also, the selected optimal solution is the best cs, that is, the selected cs for storing the split data. Data deduplication dd process eliminates the duplicate blocks of data, enhances the bandwidth over the network, and speeds-up the transfer of data, which brings about effectual data transmission. During dd, the cs creates a hash code (hc) for every split data with the aid of the whirlpool algorithm and verifies the existence of hash value with the utilization of the hash chaining technique (hct) in the dcs. The whirlpool algorithm is centered on the utilization of a block cipher for the compression function. To determine the hash value of data, it performs 4 steps: i) appending padding bits, ii) appending length, iii) initializing hash matrix, and iv) processing message in 512bit blocks. It takes the split file data $(F_b)_i$ for generating the hash values and is written as,

$$Wh\tilde{\mathrm{H}}_f : (F_b)_i \to \tilde{\mathrm{H}}_{Inter} \to \tilde{\mathrm{H}}_i \quad (4)$$

$$\tilde{\mathrm{H}}_{Inter} = E(\tilde{\mathrm{H}}_{i-1}, (F_b)_i) \oplus \tilde{\mathrm{H}}_{i-1} \oplus (F_b)_i \quad (5)$$

Where,

$Wh\tilde{\mathrm{H}}_f$ - whirlpool hashing function,

$\tilde{H}_{Inter}$ - intermediate value

$\tilde{H}_i$ - generated hc values for the corresponding input $(F_b)_i$

Input: Split Files $(F_b)_i$
Output: Remove Redundant Data from Split Files

Begin
  Initialize n number of split files
  for n number of split files **do**
    {
      $Wh\tilde{H}_f : (F_b)_i \rightarrow \tilde{H}_{Inter} \rightarrow \tilde{H}_i$ *// Hash code generation*
    }
    if ($\tilde{H}_i$ == *hash chain*) **then** *// Data Deduplication*
      Remove the corresponding file
    else
      Store the file to the cloud server
    end if
  end for
End

FIGURE 2. Pseudocode for the Proposed DD Scheme.

Once the HC is generated, HCT is employed. In HCT, the HCs are normally stored in blocks and developed as hash chains. If an HC value gets into the hash chain, the CS verifies whether the hash chain already contains that HC. If yes, the address of the data block pointer that holds the single data segment across the cloud storage replaces the equivalent blocks of data in the files. Else, the CS executes compression and encryption and stores the file to the selected CS. The proposed DD scheme is detailed in the above pseudo-code (figure 2). Data Portability The cloud DP process securely migrate the cloud data as of one cloud provider to another or between a DCS at a particular interval. If a file is moved from its location and saved in another location, the malicious owners could not get that file even if they find any file location by cheating the CS, since no files are now present on that location. For a DP process, the proposed system utilizes the Levy Flight – Wind Driven Optimization (LF-WDO) Algorithm. As the WDO algorithm randomly fixes the position as well as velocity of the air parcel, it does not render an exact result. So, in LF-WDO, the levy flight distribution (LFD) is done in place of this randomness to acquire effective DP. The LF-WDO algorithm trails the path of atmospheric motion of infinitesimal small air parcels that navigates over an d -dimensional search domain. Primarily, generate the population of air parcels, and assign the position (x) and velocity (v) of the air parcels grounded on LFD.

$$L(x) = t(-x) \quad 0 < x \leq 2 \tag{6}$$

$$L(v) = t(-v) \quad 0 < v \leq 2 \tag{7}$$

Where, $L(\bullet)$ signifies the LFD, t indicates the time of task completion. The pressure (P_i) of every air parcel at the current iteration s is evaluated as,

$$\mathrm{P}_{i,s} = \rho' \alpha \mathrm{T} \tag{8}$$

Where,

ρ' - Density of the air parcel

T - Temperature

α - Universal gas constant

This pressure value is concerned as FV; and grounded on these FVs, rank each air parcel. Afterward, update the velocity of each air parcel as,

$$\hat{v}_{new} = (1-\lambda).L(v) - g.L(x) + \left(\alpha \mathrm{T} \left| \frac{1}{r} - 1 \right| (L'(x) - L(x)) \right) + \left(\frac{c.\tilde{v}}{r} \right) \tag{9}$$

Wherein, $\hat{v}_{new}$ implies the velocity at iteration $s+1$, g signifies the gravitational acceleration, $L'(x)$ implies the air parcel's optimum location, r signifies the rank betwixt the entire air parcels, $\tilde{v}$ implies the velocity influence as well as c indicates the coefficient ($c = -2\phi\alpha\mathrm{T}$). Here, ϕ implies the earth's rotation. If $\hat{v}_{new} > \hat{v}_{MAX}$ ($\hat{v}_{MAX} = 0.3$), then the velocity is limited as per the following condition,

$$\hat{v}_{new}^{*} = \begin{cases} \hat{v}_{MAX} & if\ \hat{v}_{new} < \hat{v}_{MAX} \\ -\hat{v}_{MAX} & if\ \hat{v}_{new} < -\hat{v}_{MAX} \end{cases} \tag{10}$$

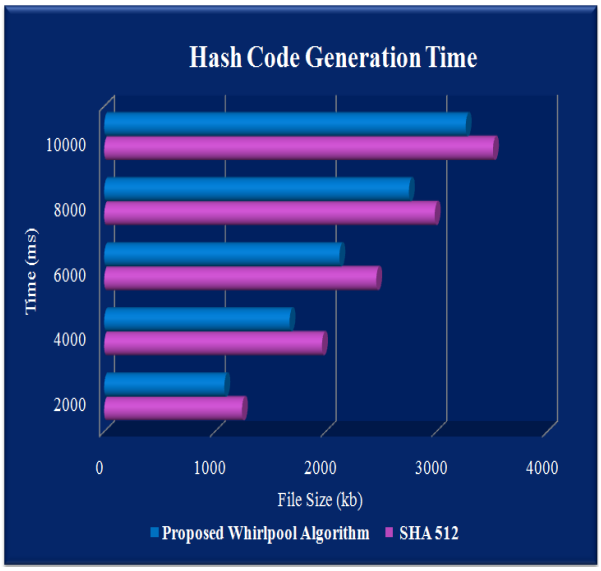
Wherein, $\hat{v}_{MAX}$ implies the maximum velocity. Subsequent to updating velocity, an air parcel's position is updated as,

$$\hat{x}_{new} = L(x) + \hat{v}_{new}.s \tag{11}$$

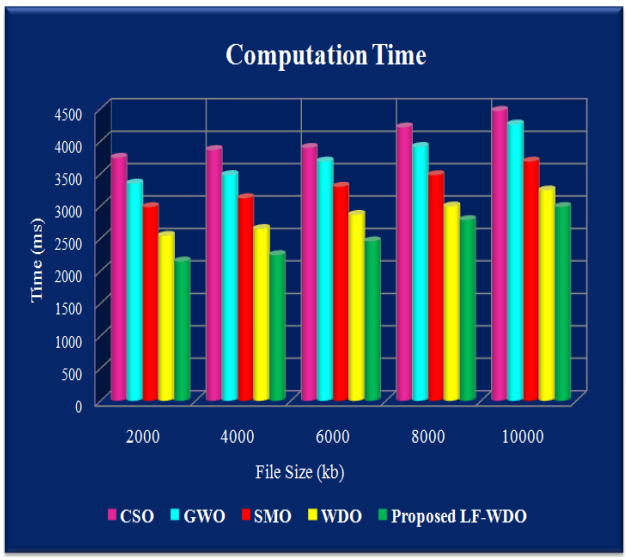
Wherein, $\hat{x}_{new}$ implies the updated position. After that, the fitness of every solution is estimated at this new position. The procedure as of pressure assessment to populace updation is repetitive until the maximal iteration is attained. Subsequent to achieving maximum iterations, air parcel that encompasses the best pressure (fitness) is chosen as an optimum solution. Similar to optimum air packet selection, a CS location is chosen at a specific interval time. Subsequent to choosing a location, the files are moved as of its current location to the chosen location. This DP process is repetitive at a specific interval time.

4. Result and Discussion

The proposed secure DD system with a DCS is analyzed in this section. The proposed approaches like the whirlpool algorithm, HFGA, and LF-WDO are contrasted to the existing algorithms centered on their performance. This comparison is performed by considering HC generation time, computation time, and FV, which is elucidated further using the below figures. Here, JAVA is the platform utilized for system implementation.



(a) Proposed whirlpool algorithm based on hash code generation time



(b) Proposed LF-DWO based on computation time

FIGURE 3. Performance analysis

Figure 3 contrasts the proposed secure DD system with a DCS and certain existing methodologies for proving the proposed system's efficiency. Figure 3(a) graphically analyzed the proposed Whirlpool algorithm and the existing SHA512 algorithm in respect of HC generation time. Generally, the hash value is quickly generated only by a pre-eminent hash function algorithm. According to that, for the average file size of 6000kb, the proposed whirlpool algorithm takes an average HC generating time of 2180 ms, but, the SHA512 algorithm consumes a more average time of 2432ms. From this analysis, the proposed whirlpool algorithm is perceived to generate HC much faster compared to others, and therefore, it is an excellent hash function algorithm. Figure 3 (b) compares the proposed LF-DWO optimization algorithm and the existing cuckoo search optimization (CSO), grey wolf optimization (GWO), spider monkey optimization (SMO), along with wind-driven optimization (WDO) in respect of the computation time. For the average file size of 4000kb, the existing CSO, GWO, SMO, and WDO acquires an average computation time of 4037, 3736, 3313, and 2857ms, respectively, but the proposed LF-DWO takes 2524 ms, which is two times faster when contrasted to the existing optimizations. Likewise, for other file sizes, the proposed LF-DWO takes less time when analogized to the existing optimizations. Figure 4 compares the proposed HFGA algorithm and the existing Particle Swarm Optimization (PSO), Ant Bee Colony (ABC), Genetic Algorithm (GA), along with Forest Optimization Algorithm (FOA) in respect of the FVs generated at different iterations. For 15 iterations, the proposed HFGA shows an average FV of 85, whereas, the existing PSO, ABC, GA, and FOA optimizations acquire 53, 60, 67, and 77– average FVs, respectively. From this analysis, the FV is higher for the proposed HFGA when analogized to others. On that account, the proposed HFGA is confirmed to render effective data transmission by choosing a suitable CS for file storing.

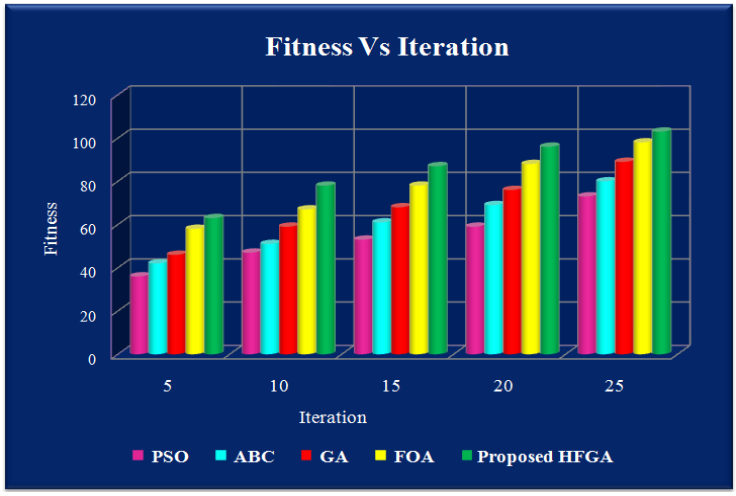


FIGURE 4. Fitness Vs Iteration

5. Conclusion

In a distributed cloud environment, it is vital to manage the cloud data and also prevent it as of the attacks. To attain these '2' objectives, this paper developed '2' novel techniques: Whirlpool algorithm and hash chaining based DD and LF-WDO based DP. The proposed work maintains cloud data, support portability, as well as render effective sharing of files amongst users. And also for verifying the proposed work's potential, the proposed techniques' performance is compared with the prevailing techniques. The investigational outcomes exhibited that the proposed work renders an excellent performance than those prevailing techniques. In terms of disparate file sizes, the processing time along with the HC generation time of the proposed and the existing algorithm is computed. Within 3265ms, the proposed Whirlpool algorithm renders an HC for the 10000kb file. The proposed LF-WDO achieves a lower process time of 2987ms, which is above 5% smaller than the existent techniques. The security of file data augmented swiftly via moving the files quickly from one site to another. Therefore, as of the outcomes, it is illustrated that the proposed work is highly effective as well as safe than the other techniques. However, still, the data owner suffers as of a lack of confidentiality in the distributed cloud setting; thus, to enhance the data security along with confidentiality, in the future, effectual authentication schemes will be integrated with it.

References

[1]. Ruhul Amin, Neeraj Kumar, G. P. Biswas, Rahat Iqbal, and Victor Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment", Future Generation Computer Systems, vol. 78, pp. 1005-1019, 2018.

[2]. Fan Wu, Xiong Li, Lili Xu, Arun Kumar Sangaiah, and Joel JPC Rodrigues, "Authentication protocol for distributed cloud computing: An explanation of the security situations for Internet-of-Things-enabled devices", IEEE Consumer Electronics Magazine, vol. 7, no. 6, pp. 38-44, 2018.

[3]. Priyanka Maharu Salunke, and Vishal V. Mahale, "Secure Data sharing in Distributed Cloud Environment", In 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on, IEEE, pp. 262-266, 2018.

[4]. Muhammad Usman, Mian Ahmad Jan, and Xiangjian He, "Cryptography-based secure data storage and sharing using HEVC and public clouds", Information Sciences, vol. 387, pp. 90-102, 2017.

[5]. Haoran Yuan, Xiaofeng Chen, Jin Li, Tao Jiang, Jianfeng Wang, and Robert Deng, "Secure cloud data deduplication with efficient re-encryption", IEEE Transactions on Services Computing, 2019.

[6]. Milind Waghmare B., and Suhasini V. Padwekar, "Survey on techniques for Authorized Deduplication of Encrypted data in Cloud", In International Conference on Computer Communication and Informatics (ICCCI), IEEE, pp. 1-5, 2020.

[7]. Haoran Yuan, Xiaofeng Chen, Jin Li, Tao Jiang, Jianfeng Wang, and Robert Deng, "Secure cloud data deduplication with efficient re-encryption", IEEE Transactions on Services Computing, 2019.

[8]. Shunrong Jiang, Tao Jiang, and Liangmin Wang, "Secure and efficient cloud data deduplication with ownership management", IEEE Transactions on Services Computing, 2017.

[9]. Samiksha Chavhan, Pragati Patil, and Gajanan Patle, "Implementation of Improved Inline Deduplication Scheme for Distributed Cloud Storage", In 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, pp. 1406-1410, 2020.

[10]. Esteban Lopez-Falcon, C., Vanessa Miranda-López, Andrei Tchernykh, Mikhail Babenko, and Arutyun Avetisyan, "Bi-objective Analysis of an Adaptive Secure Data Storage in a Multi-cloud", In Latin American High Performance Computing Conference, Springer, Cham, pp. 307-321, 2018.

[11]. Zheng Yan, Mingjun Wang, Yuxiang Li, and Athanasios V. Vasilakos, "Encrypted data management with deduplication in cloud computing", IEEE Cloud Computing, vol. 3, no. 2, pp. 28-35, 2016.
[12]. Hui Tian, Fulin Nan, Chin-Chen Chang, Yongfeng Huang, Jing Lu, and Yongqian Du, "Privacy-preserving public auditing for secure data storage in fog-to-cloud computing", Journal of Network and Computer Applications, vol. 127, pp. 59-69, 2019.
[13]. Yuqing Mo, "A data security storage method for iot under hadoop cloud computing platform", International Journal of Wireless Information Networks, vol. 26, no. 3, pp. 152-157, 2019.
[14]. Xiaoyu Zheng, Yuyang Zhou, Yalan Ye, and Fagen Li, "A cloud data deduplication scheme based on certificateless proxy re-encryption", Journal of Systems Architecture, vol. 102, pp. 101666, 2020.
[15]. Dilip Venkata Kumar Vengala, D. Kavitha, and A. P. Kumar, "Secure data transmission on a distributed cloud server with the help of HMCA and data encryption using optimized CP-ABE-ECC", Cluster Computing-The Journal Of Networks Software Tools And Applications, 2020, 10.1007/s10586-020-03114-1..
[16]. Fizza Shahid, Humaira Ashraf, Anwar Ghani, Shahbaz Ahmed Khan Ghayyur, Shahaboddin Shamshirband, and Ely Salwana, "PSDS–Proficient security over distributed storage: a method for data transmission in cloud", IEEE Access, vol. 8, pp. 118285-118298, 2020, 10.1109/ACCESS.2020.3004433.
[17]. Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, and Hui Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing", Information Sciences, vol. 387, pp. 103-115, 2017, 10.1016/j.ins.2016.09.005.
[18]. Beulah, S., and F. Ramesh Dhanaseelan, "Detection of duplicated data with minimum overhead and secure data transmission for sensor big data", Cluster Computing, vol. 22, no. 5, pp. 10467-10479, 2019, 10.1016/j.ins.2016.09.005.
[19]. Wei Liang, Yongkai Fan, Kuan-Ching Li, Dafang Zhang, and Jean-Luc Gaudiot, "Secure data storage and recovery in industrial blockchain network environments", IEEE Transactions on Industrial Informatics, 2020, 10.1109/TII.2020.2966069.
[20]. Andrei Tchernykh, Mikhail Babenko, Nikolay Chervyakov, Vanessa Miranda-López, Arutyun Avetisyan, Alexander Yu Drozdov, Raul Rivera-Rodriguez, Gleb Radchenko, and Zhihui Du, "Scalable data storage design for non-stationary iot environment with adaptive security and reliability", IEEE Internet of Things Journal, 2020, 10.1109/JIOT.2020.2981276.