International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I3P214

Volume 71 Issue 3, 130-142, March 2023

Original Article

Improving File Storage Mechanism using Intelligent Data Fragmentation Model (IDFM) Algorithm and Providing Confidentiality of Data in Cloud Computing Environment

K. Rajalakshmi[1], M. Sambath[2], Linda Joseph[3]

[1,2,3]*Department of CSE, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India.*

[1]*Corresponding Author : rajee.be89@gmail.com*

Received: 27 November 2022 Revised: 28 January 2023 Accepted: 12 March 2023 Published: 25 March 2023

Abstract *- Cloud computing is a cutting-edge technology that offers users a useful environment where they can quickly and easily obtain and release reconfigurable computing resources. Cloud data security issues have attracted much attention recently since it is an integral part of cloud computing that is essential to make data and its related operations secure and safe. This study focuses on improving data confidentiality in a cloud environment by using an Intelligent Data Fragmentation Model (IDFM) that securely divides the file into different distinct fragments and generates the fact file for recovering the data in case of any catastrophic circumstances. A key problem in this research is the difficulty of ensuring confidentiality that prevents unauthorized disclosure of information, which could be achieved using fragmentation models that have their own computational overhead, key management concerns, and issues in the choice of fragment threshold. Data storage is the central part of cloud computing, where confidentiality is rendered a critical issue to be addressed. The proposed Intelligent Data Fragmentation Model (IDFM) works in two phases, wherein phase I initially concentrates on dividing the data file into smaller chunks according to some random values. Phase II concentrates on recovering the original file from the fact file in case of any disastrous situation. According to experimental results, an intermediate level of confidentiality may be reached using this model, and the efficiency of IDFM is O (n) for any number of data files.*

Keywords *- Cloud computing, Cloud service providers, Data fragmentation, Data security, Fitness value, Threshold selection.*

1. Introduction

Cloud computing can be provisioned for data hosting and management to utilize the data resources similarly. Cloud providers are hosting their data through different strategies. Businesses and individuals want to improve data reliability and service availability. In order to achieve secured data sharing, the partitioning and dynamic replication of data management in cloud storage have been proposed. A file is broken up into segments using the replication paradigm and copied from one cloud node to another. The idea of T-coloring is improved when every node stores a single fragment. Each node that stores only a single fragment improves the idea of T-coloring. In any case, the attacker does not receive any useful information, and the nodes are arranged in a specific way even if they succeed. It improves data accessibility and availability while lowering access latency. Users can process and store data in third-party data centres with the help of cloud computing and storage solutions. It can enhance the privacy and security of group-sharing data by combining proxy re-encryption, one-time password verification, and periodic file removal.

Proxy re-encryption provides data-sharing security in cloud computing. In cloud data centers, the proxy re-encryption is designed to assess security models and assess the efficiency and improvement of existing solutions. Security is the other motivation for using and expanding proxy releasable encryption. So especially interested in Proxy Re-Encryption (PRE), a security technique that allows semi-trusted proxies to re-encrypt messages encrypted with a delegate's public key using the delegate's public key without revealing either the encrypted communications or the delegate's private key. Data exchange in cloud computing can be ensured with the public key encryption provided by this technology. Since this confidential communication resides in the cloud, its owner must provide access to another authenticated person. To ensure data security in semi-trusted clouds, it is preferable that the user has access to the necessary data and can use the public key to secure personal data before uploading it. Due to the fact that sensitive data is kept in the cloud, data owners must enable other authorized user access to that data. To achieve this, users must have access to the necessary information and be able to encrypt

private data with the public key before transmitting it to a partially trusted cloud service. Upon acceptance of the shared information demand, a proxy re-encryption key is generated by using the person's secret key and the other person's public key. Here, the semi-trusted cloud provider obtains this proxy re-encryption key and uses it to transform the cypher text secured with the first user's public key into one secured with the subsequent user's public key. This way, a knowledgeable, partially trusted cloud server can take on the costly responsibility of sharing trustworthy information.

Cloud computing service providers based on information and communication technology (ICT) may encounter more difficulties in the online services sector. Using the updated Knapsack algorithm and considering reliability and load balancing, the multi-objective methodology was applied to the expense of replication from more expensive data centers to less expensive ones. As a result of file replication, reduced access latency and file service time, improvements in file availability, and load balancing, the system can be optimized for better performance.

The cloud has enabled users to have better flexibility, scalability, and transparency with good quality of service at a lower price. The Cloud also enables users to share resources like applications, software, hardware, business procedures, and any service on the network. Data is a valuable asset in every organization that is required to be kept private and secured. Data, irrespective of its form and kind, should be accurate, valid, and reliable, which is a crucial issue for any firm. Cloud computing is quickly gaining popularity as a buzzword. The field of computers is constantly changing and exhibiting persistent progress. It is used across a wide range of industries, notably, big data analytics and the internet of things, each of which has achieved significant strides. However, the security concerns and dangers that come with it are tedious. Cloud computing is a novel virtualization technology that attempts to give users access to secure, flexible, and QoS-guaranteed online data storage and processing resources.

The cloud environment offers storage space for data related to financial records, telecom systems, utilities, media and entertainment, retail, public sector etc. A Cloud Service Provider has to ensure privacy by preventing data loss. Cloud is cited as a top priority by most of the global financial services as the support is extended through cloud infrastructure and software as a service by 2019. Cost reduction, IT flexibility, competitiveness, and time to market are some of the attractive features of the cloud. In the realm of clouds, data residency and data security are the main concerns. The concern could be expanded to who manages the data and has accessed it, the law of regulation related to where the data is stored and what kind of laws are applied, and the presence of data on the provider’s end even after termination of service.

The proposed architecture is designed to focus on various challenges, including decisions about the file fragment size and the number of file fragments, including the minimum data equivalence between fragments. Choice of finding the centermost stable nodes to store file fragments using a node selection mechanism and designing an intelligent data fragmentation model to maintain confidentiality. The problem addressed here is that in large-scale systems, the problem of data reliability, confidentiality, availability and response time are dealt with by privacy-preserving mechanisms and replication strategies. In addition to moving data offsite to a cloud storage service, the data is virtualized and shared in a shared environment, increasing the attack surface for remotely stored data. The entire cloud environment is at risk if any entity is weak. When such a scenario occurs, the security mechanism implemented should significantly reduce the attacker's efforts to obtain a reasonable amount of data, reducing data loss.

Cloud computing data storage is the centermost part of cloud computing, where confidentiality is rendered a critical issue to address. The proposed Intelligent Data Fragmentation Model (IDFM) works in two phases, where phase I initially concentrates on dividing the data file into smaller chunks according to some random values. Phase II concentrates on recovering the original file from the fact file in case of any disastrous situation. Cloud service providers must assure customers that their data and applications are secure. Customers must confirm that the provider has implemented security measures to safeguard their data. For users of a public cloud service, extensive virtualization also raises special security risks. The interaction between the operating system and the underlying hardware is changed by virtualization. Computing, storage, and networking are all types of hardware. Virtualization layers need to be appropriately managed, secured and configured.

Further, the paper is consolidated as part 2 related works, part 3 describes the proposed work on Threshold selection and generation of Fact files for data recovery, part 4 explains the implementation of the Intelligent Data Fragmentation Model and its algorithm, and part 5 is about the conclusion of this work.

2. Related Work

2.1. Cryptographic Mechanism

Kaaniche N. [3], The author, originally put out a novel ID-Based Cryptography technique in which each client serves as a Private Key Generator (PKG). Suganya S et al. [10], This research paper focuses on data replication knowledge in cloud computing. AES encryption is employed here. Elmubarak S. A et al. [17], This research intends to provide consumers with higher results ranking approach to assist them in selecting the best service. The applicability of the suggested paradigm has been examined using the SMI Cloud Toolkit.

Hasan M et al. [27], This survey article compares and contrasts various fault tolerance frameworks, enabling researchers to choose the framework of their choice. Sajay K. R et al. [31], This report recommends encrypting cloud data and employing a hybrid method to boost data security. To strengthen security procedures, this process combines homographic encryption with blowfish encryption. Beckham O et al. [32], a novel architectural model described in the article to develop and extend security tools for cloud computing. This architecture offers protection to many cloud service providers. In this study, the author discussed a two-tier architecture for multi-cloud security, one on the client and one on the server. Kara M et al. [29], This article presents a method for performing Leveled Fully Homomorphic Encryption (LFHE), an asymmetric homomorphic encryption strategy that depends on translating base numbers to a certain base. This simple method can be used in a variety of disciplines, particularly those that call for great speed and safety. It is very simple to implement.

Barhoom T.S et al. [47], This work is to create a cloud confidentiality model that effectively addresses known encryption method shortcomings by combining text file fragmentation and encryption. The text files were split into two triangles using the axis in the fragmentation procedure. On the other hand, the Blowfish algorithm was utilized for encryption. According to the research, creating a multi-layer model allows for the achievement of great confidentiality. Seth B et al. [51], Researchers use double encryption and segmenting methods to prove the security of information transmission over multitenant cloud infrastructures. K. Rajalakshmi et al. [53], Based on the author's conclusion, the RSA algorithm performs better in relation to security purposes than any other cryptographic algorithm regarding cloud data security.

2.2. Data Storage Security

Zhao J et al. [1], This paper includes batch verification task capability in their scheme, allowing Third Party Administrators to conduct simultaneous public audits among several users at once. P.M et al. [4], Merkle Hash Tree (MHT) and the AES algorithm are used in the suggested approach to preserve data integrity on the untrusted server. Alani M.M et al. [6], The author of this report, explain how cloud security differs from traditional system security. A summary of the most frequent assaults on clouds is also offered, along with a short list of general security advice for the cloud. Patni P.D et al. [9], Here, nodes are assigned to the fragments and replicas using T-Coloring. In this instance, the overall processing time is fairly constant. Jegadeeswari S et al. [11], They concentrate on the technological facts of cloud computing in this study article. This research activity is suggested as an avenue for further study in the field of cloud security and is suitable for beginners. Almorsy M, et al. [12], The cloud architecture, cloud services, cloud stakeholders, and cloud service delivery models were only a few topics the author examined. Based on this study, they precisely characterise the cloud security issue and the essential components that any suggested security solution must include. Polu S.K [28], The data provider, will greatly benefit from this suggested method for protecting data from attackers. Additionally, provide a heuristic auditing approach (HAS) that increases searches to find any violations that might be anticipated given the conditions.

2.3. Data Replication Model

Chavan M.R et al. [7], The author provides an overview of cloud computing, using binary fragmentation to achieve fragmentation. T-Colouring is employed in this case. Singh J et al. [16], Here, fragmentation and dispersion ensured no severe storage capacity violations occurred. Storage capacity restrictions that made it impossible to store some fragments at nodes with the best retrieval times are to blame for the performance drop. Mansouri N et al. [19], The author introduce a concurrent upload and download method that replicates sets of information and concurrent download files repeated data fragments in order to improve overall performance and assess optimal network utilization rate, mean job completion, hit percentage, the overall amount of replications, and proportion of space to store filled. Omana M.U et al. [21], The FS algorithm is employed in this paper. Here, the files are split up into pieces, and the data from those pieces are replicated over the cloud node in random order. Mohan T et al. [22], The data was split up and spread over several nodes created by a random number generator. Compared to full-scale reproduction processes, DROPS technology offers high efficiency. The speed of data retrieval and performance were both improved. The number of nodes and their capacity can be changed to improve performance further. Divya S.V et al. [25], The method delivers great data security using T-coloring techniques and elliptic curve cryptography.

Additionally, the technique enforces collusion resistance and protects user privacy. Nashat D et al. [23], This article attempts to complete this stage by presenting a thorough taxonomy of the fragmentation and allocation mechanisms accessible in distributed database design. The article also analyses a few case stories to understand these approaches' successes and limits fully. Nithya L.M et al. [35], The data approach calculates the limited fingerprint for each and every data item using hashing methods like MD5 and SHA. The found pattern is then cross-referenced with other readily available bits in a database designed specifically for the storage of the pieces. Shakarami et al. [44], To deal with fragmented databases during synchronous replication, this work introduce the Grid Quorum technique. Using this approach, a database can be divided into several sections. Tos U et al. [45], Without a framework in place, database managers find it challenging to do fragmentation, and doing it dynamically is even more challenging. The dynamic fragmentation approach proposed in this work significantly

simplifies the challenge of precise and appropriate fragmentation. Waseem Q et al. [46], The results of encryption and decryption can be communicated and securely transferred to authorized recipients using identity-based encryption (IBE). The findings also indicate that more data is hidden in the image as optical distortion decreases. Sharma D.K et al. [54], The database is effectively fragmented into clusters utilizing the Fibonacci heap and an upgraded CRUD matrix in the proposed technique. Following that, the pieces are distributed and copied throughout various network nodes based on the manipulation and reading operations carried out at the individual sites while taking into account the cost factor.

2.4. Data Leakage Detection

Hussein N.H et al. [13], There is a need to ensure that users' sensitive data stays secure and private in the cloud. This article examines various issues and solutions related to these concerns. The challenges posed by the security of cloud computing and the potential solutions have been the focus of this study. Geetha S et al. [14], They recommended the DROPS approach, a framework for secure cloud storage that solves the security and efficiency that speed up retrieval. There are many nodes where fragments are scattered. Kumar P.R et al. [26], The deployment methodologies and service delivery models for cloud computing are also discussed in this study. Whenever there is a compromise of data or corruption of information, all public trust is undermined, ultimately resulting in the failure of the business or cloud computing as a whole. Mokadem R et al. [38], The safeguarding of authorized users from any false identities is one of this architecture's most crucial components.

Periyanatchi S et al.[41], Regarding both security and performance problems, in this study, the author proposes the Lion Optimization Algorithm (LOA). Data duplication and fragments throughout the cloud are measured using optimized centrality. It was seen that there was less performance overhead and a higher level of security. Shajan A et al. [48], The main online dangers to a cloud computing system are covered in this essay. It also discusses the safeguards and procedures a cloud provider should employ to ensure the cloud environment is secure and impenetrable. Vulapula S.R et al. [49], In this work, a hybrid cloud architecture-based data storage system for unstructured data is suggested. Yuvaraman S et al. [52], The author is implementing cutting-edge methodologies like Bitwise to fragment and repartition the information and constructing transforming blocks to effectively fetch the information to implement the recommended system safeguards the secrecy, truthfulness, and accessibility of the user's data. Data integrity is checked using hashing as it is downloaded from the cloud server.

2.5. Data Fragmentation Scheme

Khedkar S.V et al. [2], Here, Automatic partitioning occurs when data is fed for cloud storage. If the original file needs to be accessed, it is also rebuilt Pardeshi Ali M et al. [5]; in this study, the author proposes the (DROPS), which takes a combined approach to performance and security challenges. To avoid an intruder from predicting where the pieces are stored, graph T-coloring segregates the nodes containing the fragments by a predefined proportion. Sanjeevi P et al. [8], Here, in order to increase online security by 14.4%, the study extends the DROP approach with the AES algorithm. Edwin E.B et al. [15], Given that it is the most straightforward and well-known sort of text-based password employed here, one-time secret keys can be used to secure account access. Abdel Raouf A.E et al. [18], This study offers an improved allocation and replication technique that can be used in the early stages of a distributed database architecture when there is no knowledge of how a query will be executed. Additionally, many clustering strategies are presented. Ali K.F et al. [20], In this work, the author offers Attribute-Based Encryption (ABE) for data partition and replication in the cloud, which concurrently tackles security and performance issues. Nashat D et al. [23], This article attempts to complete this stage by presenting a thorough taxonomy of the fragmentation and allocation mechanisms accessible in distributed database design. For a fuller understanding of these strategies' successes and limits, the article also analyses a few case stories.

Lentini S et al. [24], This study focuses on data fragmentation techniques and illustrates how they can affect a cloud service's overall performance. Sugumar R et al. [30], The RRNS encoding and CTNA, which uses the centralization measurement as just a point of allocation and recovery, are employed to segment the data. Najmi M et al. [33], For more efficiency, the author here offers load balancing using cutting-edge load balancing algorithms. There are no known security concerns with data encryption and fragmentation in cloud computing. Santos N et al. [34], This article looked at several data fragmentation techniques to prevent direct unauthorized entry to the information stored in the cloud. It evaluates how well they operate on a cloud instance while considering all processing time, including data upload and download time in the implementation. Ubaidillah et al. [37], This survey article examines and evaluates several fragmentation techniques in terms of the performance of various databases. Khobragade et al. [39], Implementation of the HDRS Method (Hierarchical Data Replication Strategy). In this study report, they used the idea of exponential decay/growth to concentrate on popular files. Finally, HDRS modifies the copies according to their relevancy.

Jose P.P et al. [40], The optimal node for replication is selected using a method inspired by the characteristics of the bee colony Optimization method. Consequently, fewer copies are needed to accomplish the same load-balancing effect. Prajapati P et al. [42], Data deduplication can reduce storage server computing, bandwidth, and time is made

available to small businesses within private cloud storage. Combining the ideas of identity-based encryption and attribute-based encryption can result in secure data deduplication with more security and less overhead. Kallel S et al. [43], Dynamic data replication is advised by the Multi-Objective Particle Swarm Optimization (MO-PSO) and Ant Colony Optimization techniques (MO-ACO).

The proposed method was also put through a simulation using CloudSim. Every Data Centre (DC) has hosts that are home to a number of virtual machines (VMs). Ge Y. F et al. [50], A perturbation-based method with adaptive modification strategic decision-making is created to convey adaptive material in the outcomes appropriately. Despite its great precision, faster resolution, and flexibility, the proposed method outperforms the competitors.

3. Intelligent Data Fragmentation Model (IDFM)

File fragmentation generally occurs when the file is not required to be stored in the correct sequence in successive blocks on the disk. It is proposed in this methodology to develop a model to store file fragments on different nodes using this methodology. Security in large-scale systems depends on both the system as a whole and each individual node. A hacker will have only one point of failure if the file is compromised.

Data division or fragmentation technology is thus employed to prevent such errors. An attacker's effort may be increased through fragmentation. A file 'f' is divided into 'n' fragments in a fragmentation scheme; each fragment is signed and sent to 'n' remote servers, one fragment per server. An existing fragmentation method is done, which is based on a user's preference for the threshold value.

It is the user's responsibility to determine the fragmentation threshold for a data file. The user also defines the fragmentation threshold in terms of either proportion or the quantity and size of various pieces. The summarized drawbacks of all the existing fragmentation models are listed in Table 1.

This methodology uses an Intelligent Data Fragmentation Model (IDFM) to find the fragmentation threshold value by which a file should be fragmented into small chunks. Customers can choose their Cloud Service Provider (CSP) for offsite data storage based upon the Service Level Agreement (SLA), which details their features and capabilities.

Once a CSP is decided, the user can submit their data file for storage by entrusting it to the CSP for storage space, security, integrity, reliability, and availability.

Table 1. Drawbacks of the Existing Data Fragmentation Model

Existing Model	Drawbacks
AES-based file Encryption System	Fail to provide data integrity and availability.
Data Fragmentation and Distribution Model	Fragment threshold is not an optimized threshold value.
Cloud Storage Encryption (CSE)	This architecture is not tested to ensure the integrity of data.
Information-Centric Approach	Stands theoretical and depends on encryption methods.
Attribute-Based Encryption	Requires more effort to handle the communication between the separated parts. Also, it has failed to consider the integrity of the data file.
Obfuscrypt	It does not involve any integrity verification process.
Redundant Array of Independent Net-storages	Eliminate the trust parameter that is required. No trial or practical prototypes have been done to validate it.
Normalization of relational database	Does not guarantee data integrity.
High-Performance Anonymization Engine	Data is processed at the cloud end by employing its own control methods at the organization end.
Anonymization approach	Fails to provide data integrity.
Division and Replication of Data in the Cloud for Optimal Performance and Security	Does not concentrate on finding the sensitive information in the entire file.

The input file from the user is taken as input by the cloud controller node, and the number of words occurring in the data file is calculated using the function,

$$\text{Count}(F) = \text{number of words in } F \quad (1)$$

where F is the data file in KB.

Now, generate random values T_i where $T_i = 1$ to n and n is assumed to be 50 in this model. The number of random values is fixed to ω where ω = 3 is considered in this model.

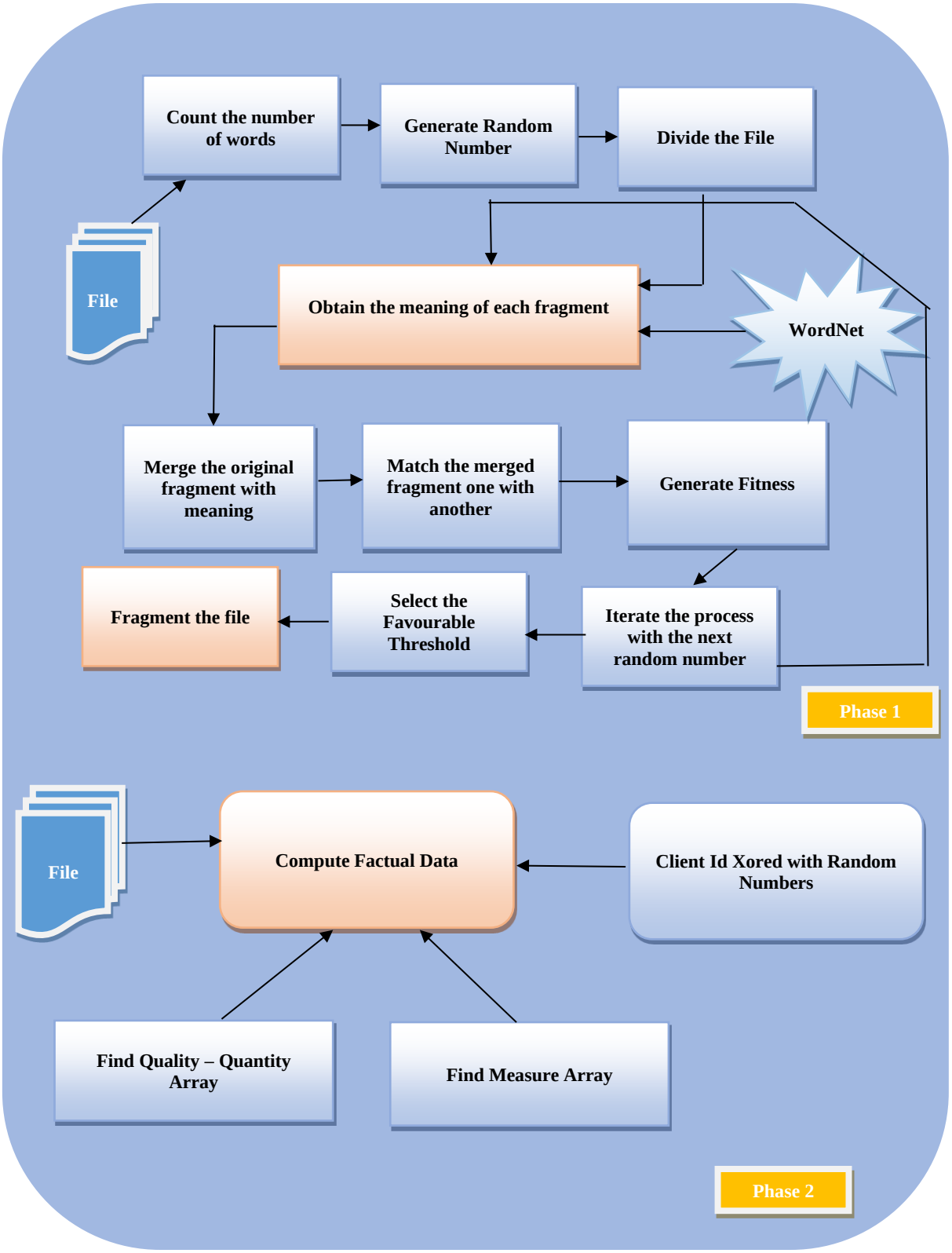

Fig. 1 Data Flow Diagram for Intelligent Data Fragmentation Model (IDFM)

3.1. Threshold Selection

Upon generating the random numbers, the file F is divided into each T_i using the,

$$Frag\ (F)\ =\ Count\ (F)\ /\ \{Ti\ Where\ i\ =\ [1,n] \qquad (2)$$

If the first random number generated is 10 then the file is divided into 10 fragments.

Using WordNet ontology, find the meaning of each fragmented segment using,

$$\sum T_i meaning\ (Frag\ [F])\ for\ each\ word\ in \sum T_i\ (Frag\ [F])$$

Now merge and match the meaning of each fragment $\sum T_i$ meaning (Frag [F]) with the original fragment $\sum T_i$ (Frag [F]) that results in $Frag(F)_{new}$.

With the newly arrived fragment, find the fitness value using,

$$Fit(T_i^{New}) = Total\ (Match(Frag(F)_{New} / T_i \forall\ T_i\ where\ i = 1\ to\ n$$

This will result in a fitness value, and the steps are again repeated with other random numbers generated. The fitness value obtained under each random number is compared, and the least similarity measure is taken as the favourable threshold $Threshold_{Fav}$ for dividing the file into fragments which are given by,

$$Threshold_{Fav}\ \cong\ measure_{least}\left(fit(T_i^{New})\right) \forall T_i \qquad (3)$$

The step-wise algorithm for threshold selection using IDFM is shown below.

Phase 1 Pseudocode:
Input : Data File in KB.
Output : File Fragmentation Threshold

Step 1 : Input File F
Step 2 : Add up using Count(F) = Number of Words in F
Step 3 : Generate random values T_i Where i= any number between 1 and n
Step 4 : For File "F," execute Steps 5 through 8 for each Ti.
Step 5 : Divide the File 'F' into each T_i using

$$Frag\ (F)\ =\ Count\ (F)\ /\ \{Ti\ |\ i\ =\ [1,n]\}$$

Step 6 : Using Word Ontology, Obtain the meaning using,
$\sum T_i$meaning (Frag [F]) for each word in $\sum$ Ti (Frag [F])
Step 7 : Merge and Match the

$$\sum Ti\ meaning\ (Frag\ [F])\ with \sum Ti\ (Frag\ [F])$$

That results in frag (F) $_{New}$
Step 8 : Calculate the fitness value,

$$Fit(T_i^{New}) =\ Total\ (Match\big(Frag(F)\big)_{New}\ /\ T_i\ \forall\ T_i\ Where\ i = 1\ to\ n.$$

Step 9 : Locate the favorable threshold, Threshold $_{fav}$
Step 10 : Fragment the data file.

3.2. Generation of Fact file for Data Recovery

Online data storage is used by many users simultaneously who are in need of data access in parallel. Data stored in the cloud may seriously threaten integrity and confidentiality due to human intrusion, network failure, equipment faults, or spoofing intention by any other third party.

To overcome such issues prevailing in cloud data storage, a novel methodology that takes the client identifier, C_{id} and a random number, $rand_{num,}$ which is XORed together and attached to Quality-Quantity [array] that holds the data in bytes with the possible occurrence of each byte in the file and a Measure[array] that represents the position in which each byte is present in the file. This factual data helps in data recovery in case of any catastrophic event.

The factual data itself is confidential as it is separated into two arrays with less probability of leaking out the data when the diversity of bytes increases.

For instance, if the number of diverse bytes is equal to 1, then the probability of reassembling the data is given by, $P = P(x) * \frac{1}{256}$ where P(x) is the possibility of finding the XORed parameter and whereas for a file with 10 diverse bytes, the probability of generating and reordering the original data is assumed to be $P = P(x) * \frac{1}{10^{256}}$. When the diversity of bytes increases, the probability of reassembling the data will be zero.

The algorithm for recovery from the fact file is shown below,
Phase 2 Pseudocode:
Input: Data File in KB.
Output: Fact file for Data Recovery

Construct the Fact $_{data}$ for each file as per the steps below.
Step 1: Construct the $Quality - Quantity\ [array]$
Step 2: Construct the $Measure[array]$
Step 3: Store the $Quality - Quantity\ [array]$ and
$Measure[array]\ by\ appending\ Cid \oplus rand\ num$

4. Results and Discussion

As part of the proposed work, cloudSim is used to implement the work in Java. In addition to improving server computation time, memory storage, and the time it takes to upload and download data, the results have also shown improvement in data upload and download times. The system has undergone various tests related to the above performance metrics and has shown enhanced output, yielding better results. Examining the proposed work with the current work reveals the numerous features and qualities that have proven the suggested work superior.

The amount of time it takes the server to find the appropriate threshold value is known as the server computing

time. The estimated and average value is compared with the current method. The figure appearing below illustrates the server computing time for a number of iterations.

The time to complete the fragmentation process with 25 iterations is 18452 ms; the novel approach finishes the 20th iteration in 16895 ms. The proposed approach finishes the fragmentation process in the 15th iteration by taking a server computing time of 14658 ms. To complete the fragmentation process in the 10th iteration, the proposed work takes 12142 ms. The overall average server computing time for the proposed work is 15536.75 ms.

The graphical illustration is demonstrated in Figure 2. The Server Computing Time (SCT) is given by:

$$SCT = \sum iteration_{Time}\{(Threshold\ fav) \mid Time = 0\ to\ n\ Seconds\} \quad (4)$$

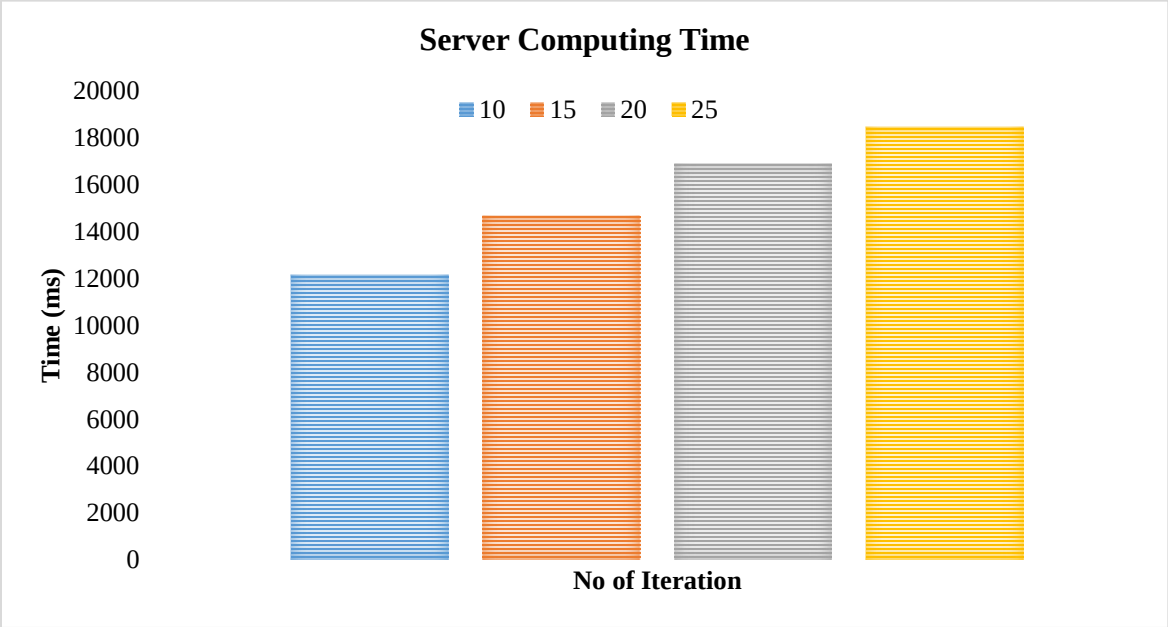

Fig. 2 Server Computing Time

4.1. Data File Uploading and Downloading Time

Receiving information from a remote system, usually a server like a web server, and downloading files from that system. In contrast, uploading involves sending data to a remote server. This system focuses on finding file fragments for downloading in order to carry out any modification. After it is done, it has to be uploaded to the same location. Table 2 below illustrates the uploading and downloading times for various file sizes. The time taken for uploading and downloading a 10 KB file size is 6539 ms and 2363 ms. The file upload and download time are given by:

$$Time(File_{upload \mid download} = \prod (File\ size, Overhead, Speed\ in\ Mbps) \quad (5)$$

Table 2. Uploading and Downloading time for various files

File Size	Uploading Time(ms)	Downloading time (ms)
10 kb	6539	2363
20 kb	8453	4650
30 kb	10112	5260
40 kb	12252	6470

Table 3. Memory occupied by various-sized files

File size (Kb)	Fact file size (Kb)	Total size (Kb)	Memory Occupied (Bytes)
10	8	18	294910
20	16	36	589821
30	24	54	884733
40	32	72	1179645
50	40	90	1474554

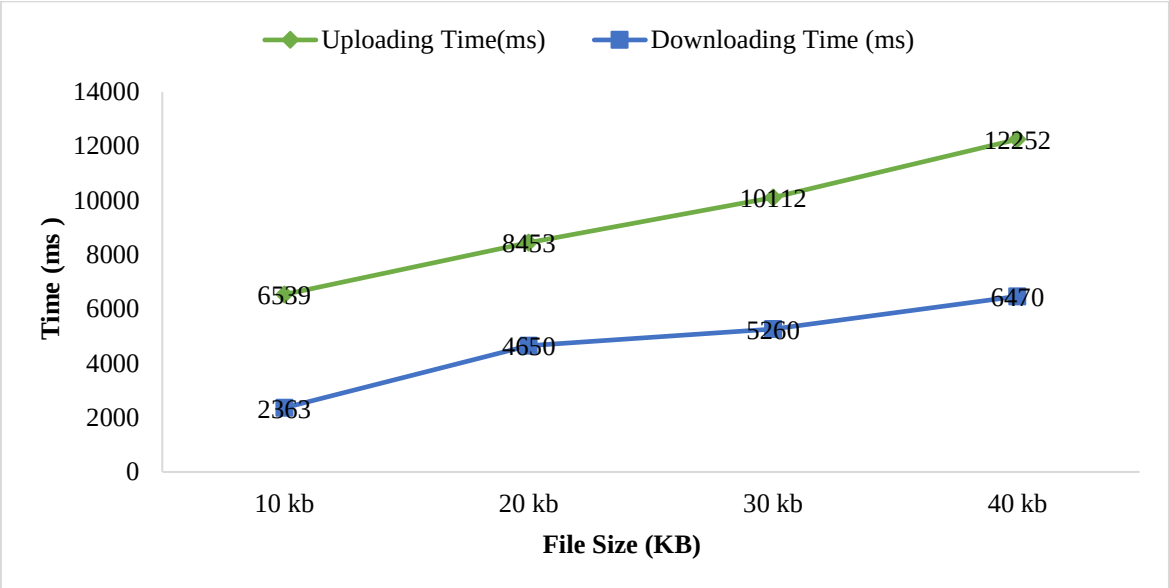

Fig. 3 File Uploading and Downloading time

The proposed approach finishes uploading and downloading a 20 kb file size in 8453 ms and 4650 ms. The time taken for completing uploading and downloading of a 30 kb file size is 10112 ms and 5260 ms, respectively. Uploading and downloading time of 12252 ms and 6470 ms is taken to finish the 40 kb file size. The corresponding graphical representation is shown in Figure 3.

4.2. Memory Value with Respect to File Size

Memory value is the number of bytes occupied by each file after the fragmentation process and creating the fact file. Table 3 reveals the memory value in bytes for various-sized files and a fact file.

Memory Value (MV) is given by:

$$MV = \prod (File\ fragment_{Size} + Fact\ File_{size} 4096\ bytes) \quad (6)$$

The memory space utilized by the suggested novel technique is presented in Table 3 to complete the fragmentation process and impose data protection in the cloud computing environment. The greater the file size, the greater the memory value occupied.

The various factors and characteristics that make the proposed work superior can be determined through a comparative analysis of the existing and current work. By comparing IDFM and DROPS approaches, which are both based on user-defined thresholds, experiments were conducted to determine their effectiveness.

Table 4 reveals the result of the closeness between fragments obtained using the existing DROPS methodology and the proposed IDFM.

It is observed that for a file size of 10Kb,20Kb,30Kb, and 40Kb, the IDFM phase I is iterated. The proposed model outperforms the DROPS with a larger difference in terms of closeness between data fragments that keeps up confidentiality.

For a file size of 40Kb, the confidentiality or privacy percentage of IDFM is found to be 16.4% greater than the existing DROPS model due to getting the fragment threshold from the user.

The findings of the proposed and current models are presented in the below figure with the file size versus their privacy (%).

Parameters Considered:

F(Size) – File Size in Kb

$\sum T_i\ Frag(F)$ – Number of Fragments

$Fit\ (Ti^{New})$ – Fitness Value

The privacy (%) is calculated using:

$$Privacy\ (\%) = F\ (Size) / Fit\ (T_i^{New}) \quad (7)$$

Comparing the existing work with the proposed work proves that uploading and downloading time works better.

Table 5 illustrates the uploading time and downloading time excluding fact file comparison with the existing Homomorphic Token and Cryptographic Encryption (HTCE) method, where there is high overhead in the encryption of data fragments and less overhead in dividing the file into fixed-sized blocks.

Table 4. Confidentiality level of DROPS (Existing model) and IDFM (Proposed model)

File size (Kb)	No. of fragments by DROPS	No. of fragments by IDFM	No. of information matched in DROPS	No. of information matched in IDFM	DROPS Privacy (%)	IDFM Privacy (%)
10	4	12	15	6	0.7	1.6
20	5	16	15	4	1.3	5.0
30	6	16	11	4	2.7	7.4
40	7	22	11	2	3.6	20

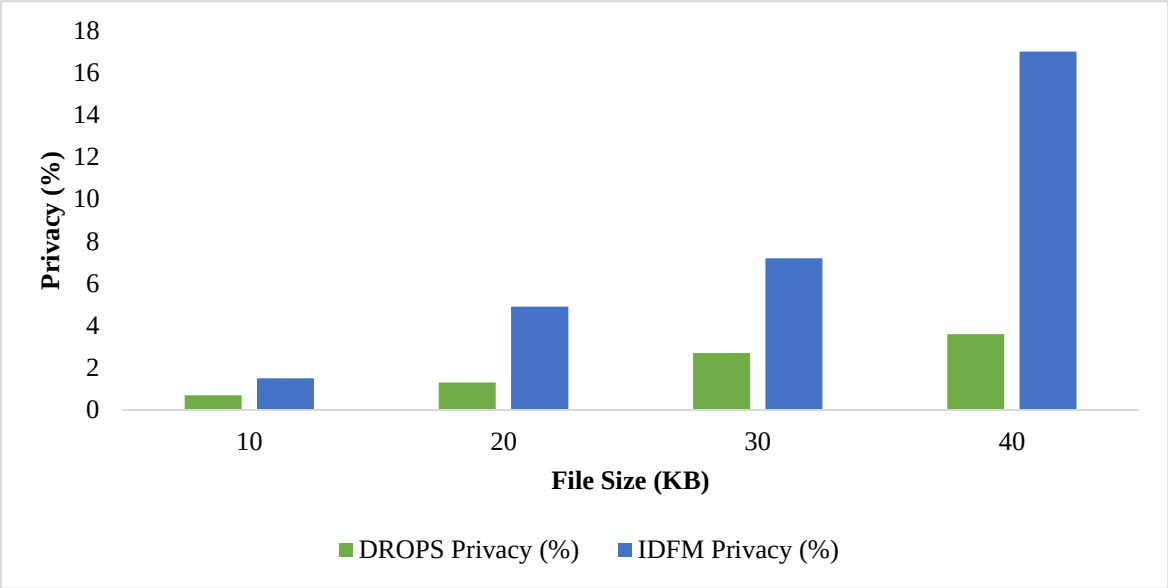

Fig. 4 Different-sized files with their Privacy Percentage

Table 5. Comparative analysis of uploading and downloading time

Method	Uploading Time (S)	Downloading time (S)
IDFM Model	9.30	4.61
Homomorphic Token and Cryptographic Encryption (HTCE) Method	30	15

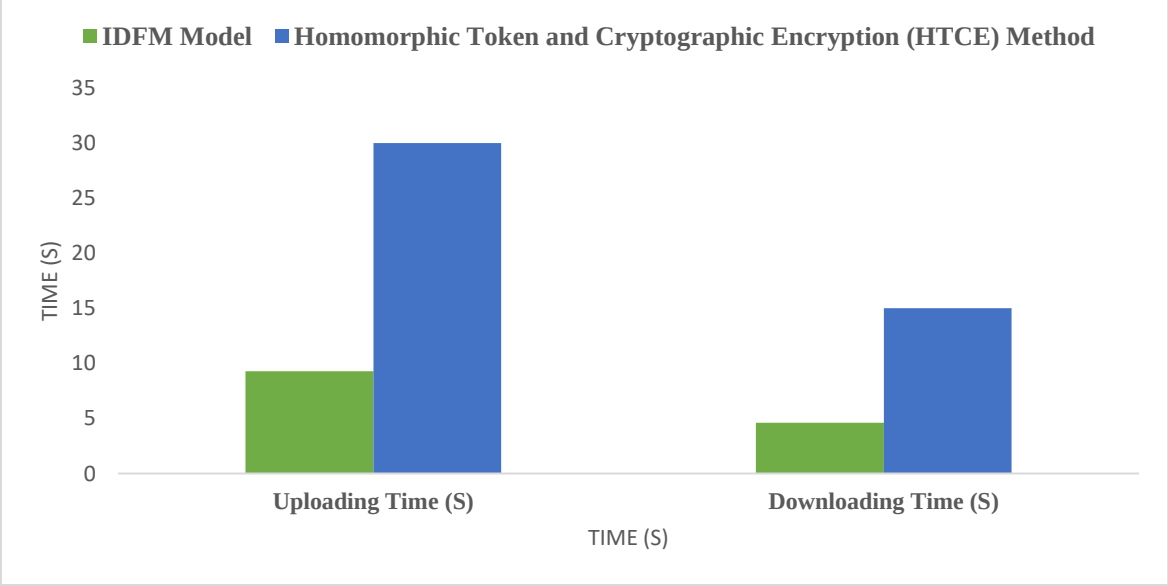

Fig. 5 Upload and Download Time of the IDFM and the HTCE Method

Table 5 shows that the suggested technique takes minimum time to upload and download the file in the cloud server. Here the proposed IDFM model takes 9.30 seconds to upload the file to the cloud server, but the existing method takes nearly 30 seconds to upload the file. The proposed approach uploads the file with less delay than the current methods. The proposed method takes 4.61 seconds less time to download the file from the cloud server than the current method, which takes 15 seconds.

The graphical representation of the uploading and downloading time of the proposed technique with the existing method is plotted in Figure 5.

5. Conclusion

Intruders on the internet are constantly working to take full advantage of the gaps and flaws present in the cloud environment so that the facilities slip back, rendering the service unavailable or the customer experience unsatisfactory, earning a bad reputation for the CSP as people prioritize their applications and storage services over people and every component that makes up information components. This makes offering features like security, valid access control, triviality, and traceability possible. The allocation of replicas and fragments inside the multi-cloud framework emphasizes the effectiveness of dynamic replication for both management and fragments. Every organization values its data, which must be protected and kept secret. Data should be accurate, legitimate, and dependable regardless of its format or nature, which is a critical concern for any business. Cloud data security issues have drawn much attention recently since they are an essential part of cloud computing and because it is required to guarantee the security and safety of data and its related operations. By utilizing the Intelligent Data Fragmentation Model (IDFM), which securely separates the file into many independent pieces and creates the fact file for recovering the data in case of major disasters, this research study focuses on enhancing data confidentiality in a cloud environment. IDFM model is used to generate the favorable threshold that is used for dividing the file into fragments. This model also ensures that even if one node is compromised by intruders, it will not disclose the entire data. Also, most central nodes are selected using degree centrality for fragment storage to reduce access time. In order to withstand any disastrous situation, a recovery technique called fact file is generated from the original file that can assist in recovering the original file. Thus a medium level of confidentiality can be obtained through this model, and the efficiency of IDFM is O (n) for any number of the data file. An intelligent Data Fragmentation Model is proposed that helps the users to fragment the data based on the threshold obtained and depending less on cryptographic mechanisms to protect the data at the remote site. WordNet Ontology is used to find the fitness between fragments so that if one fragment is compromised, it does not reveal any information about data in the next fragment or its continuity. The fragmented data are stored at various nodes obtained through centrality measures and node stability factors, making the data access very efficient as the center-most stable nodes are selected for primary fragment placement. The findings of this study encourage the future enhancement that the performance of cloud computing environment can be further maximized if dependencies between tasks are modelled using workflows at certain levels.

References

[1] Zhao Jining et al., "Identity-Based Public Verification with Privacy-Preserving for Data Storage Security in Cloud Computing," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96, no. 12, pp. 2709-2716, 2013. [CrossRef] [Publisher link] [Google Scholar]

[2] Swapnil V.Khedkar, and A.D.Gawande "Data Partitioning Technique to Improve Cloud Data Storage Security," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3347-3350, 2014. [Publisher link] [Google Scholar]

[3] Nesrine Kaaniche, "Cloud Data Storage Security Based on Cryptographic Mechanisms," *(Doctoral dissertation, Institute National des Télécommunications)*, 2014.

[4] Poonam M. Pardeshi, and Deepali R. Borade, "Improving Data Integrity for Data Storage Security in Cloud Computing," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 15, no. 7, pp. 61-67, 2015. [Publisher link] [Google Scholar]

[5] Mazhar Ali et al., "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security," *IEEE Transactions on Cloud computing*, vol. 6, no. 2, pp. 303-315, 2015. [CrossRef] [Publisher link] [Google Scholar]

[6] Mohammed M. Alani, *Elements of Cloud Computing Security: A Survey of Key Practicalities*, Springer International Publishing, 2016. [Publisher link] [Google Scholar]

[7] Mrs. Radhika Chavan, and S.Y.Raut "Cloud Security Solution: Fragmentation and Replication," *International Journal of Advance Research and Innovative Ideas in Education*, vol. 2, no. 4, 2016. [Google Scholar]

[8] P. Sanjeevi P, G. Balamurugan, and P. Viswanathan, "The Improved DROP Security Based on Hard AI Problem in Cloud," *International Journal of Internet Protocol Technology*, vol. 9, no. 4, pp. 207-217, 2016. [CrossRef] [Publisher link] [Google Scholar]

[9] P.D. Patni, and S.N. Kakarwal, "Security Enhancement of Data in Cloud Using Fragmentation and Replication," *International Journal of Engineering and Management Research*, vol. 6, no. 5, pp. 492-497, 2016. [Publisher link] [Google Scholar]

[10] S.Suganya, and R.Kalaiselvan, "An Optimization and Security of Data Replication in Cloud Using Advanced Encryption Algorithm." *International Journal of Engineering and Computer Science*, vol. 5, no. 6, pp. 2319-7242, 2016. [Publisher link] [Google Scholar]

[11] S. Jegadeeswari, P. Dinadayalan, and N. Gnanambigai, "Enhanced Data Security Using Neural Network in Cloud Environment," *International Journal of Applied Engineering Research,* vol. 11, no. 1, pp. 278-285, 2016. [Publisher link] [Google Scholar]

[12] Mohamed Al Morsy, John Grundy, and Ingo Müller, "An Analysis of the Cloud Computing Security Problem," *ArXiv preprint,* 2016. [Publisher link] [Google Scholar]

[13] Nidal Hassan Hussein, and Ahmed Khalid "A Survey of Cloud Computing Security Challenges and Solutions." *International Journal of Computer Science and Information Security*, vol. 14, no. 1, pp. 52, 2016. [Google Scholar]

[14] S. Geetha et al., "Data Leakage Detection and Security Using Cloud Computing," *International Journal of Engineering Research and Applications*, vol. 6, no. 3, 2016. [Google Scholar]

[15] Edwin E.Bijolin, P. Umamaheswari, and M. Thanka Roshni, "Fragmentation and Dynamic Replication Model in Multicloud by Data Hosting with Secured Data Sharing," *Asian Journal of Research in Social Sciences and Humanities*, vol. 7, no. 2, pp. 459-474, 2017. [CrossRef] [Publisher link] [Google Scholar]

[16] Jayshri Singh, and Keshav Kori, "Optimal Performance and Security in Cloud Using Division and Replication of Data," *Journal of Network Communications and Emerging Technologies,* vol. 7, no. 5, 2017. [Publisher link] [Google Scholar]

[17] Sahar Abdalla Elmubarak, Adil Yousif, and Mohammed Bakri Bashir, "Performance Based Ranking Model for Cloud SaaS Services," *International Journal of Information Technology and Computer Science*, vol. 9, no. 1, pp. 65-71, 2017. [CrossRef] [Publisher link] [Google Scholar]

[18] Ahmed E. Abdel Raouf, Nagwa L. Badr, and Mohamed Fahmy Tolba, "Distributed Database System (DSS) Design over a Cloud Environment," *Multimedia Forensics and security, Intelligent Systems Reference Library*, vol. 115, pp. 97-116, 2017. [CrossRef] [Publisher link] [Google Scholar]

[19] N. Mansouri, M. Kuchaki Rafsanjani, and M.M. Javidi, "DPRS: A Dynamic Popularity Aware Replication Strategy with Parallel Download Scheme in Cloud Environments," *Simulation Modelling Practice and Theory*, vol. 77, pp. 177-196, 2017. [CrossRef] [Publisher link] [Google Scholar]

[20] Khan Faisal Ali, and Kanade Kalpana, "A Review On-Reducing The Data Attacks on the Cloud by Dividing and Replicating the Data Over Multiple Cloud Nodes," *International Journal of Advance Scientific Research and Engineering Trends*, vol. 2, no. 4, 2017. [Publisher link] [Google Scholar]

[21] M U Omana, S. Nandhini, and R. Priyadharshini, "Improving Security and Performance by Implementing FS Algorithm in Distributed Cloud," *International Journal for Advance Research and Development*, vol. 3, no. 4, pp. 96-99, 2018. [Publisher link] [Google Scholar]

[22] Thulasi Mohan et al., "Divisioning and Replicating Data in Cloud for Optimal Performance and Security," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 8, pp.271-274, 2018. [Publisher link] [Google Scholar]

[23] Dalia Nashat, and Ali A. Amer, "A Comprehensive Taxonomy of Fragmentation and Allocation Techniques in Distributed Database Design," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1-25, 2018. [CrossRef] [Publisher link] [Google Scholar]

[24] Salvatore Lentini, Enrico Grosso, and Giovanni L. Masala "A Comparison of Data Fragmentation Techniques in Cloud Servers," *International Conference on Emerging Internetworking, Data & Web Technologies*, vol. 17, pp. 560-571, 2018. [CrossRef] [Publisher link] [Google Scholar]

[25] S. V. Divya, R. S. Shaji, P enkadesh, "An Efficient Data Storage and Forwarding Mechanism Using Fragmentation-Replication and DADR Protocol for Enhancing the Security in Cloud," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 1, pp. 111-120, 2018. [CrossRef] [Publisher link] [Google Scholar]

[26] P. Ravi Kumar, P. Herbert Raj, and P. Jelciana "Exploring Data Security Issues and Solutions in Cloud Computing," *Procedia Computer Science*, vol. 125, pp. 691-697, 2018. [CrossRef] [Publisher link] [Google Scholar]

[27] Moin Hasan, and Major Singh Goraya, "Fault Tolerance in Cloud Computing Environment: A Systematic Survey," *Computers in Industry*, vol. 99, pp. 156-172, 2018. [CrossRef] [Publisher link] [Google Scholar]

[28] Sandeep Kumar Polu, "Security Enhancement for Data Objects in Cloud Computing," *International Journal for Innovative Research in Science & Technology*, vol. 5, no. 6, pp. 18-21, 2018. [Publisher link] [Google Scholar]

[29] Mostefa Kara et al., "A Homomorphic Digit Fragmentation Encryption Scheme Based on the Polynomial Reconstruction Problem," *4th International Conference on Future Networks and Distributed Systems*, pp. 1-6, 2020. [CrossRef] [Publisher link] [Google Scholar]

[30] R. Sugumar, A. Rajesh, and R. Manivannan, "Performance Analysis of Fragmentation and Replicating Data Over Multi-clouds with Security," *International Conference on Computer Networks and Communication Technologies*, vol. 15, pp. 1031-1040, 2019. [CrossRef] [Publisher link] [Google Scholar]

[31] K. R. Sajay, Suvanam Sasidhar Babu, and Yellepeddi Vijayalakshmi, "Enhancing the Security of Cloud Data Using Hybrid Encryption Algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp.1-10, 2019. [CrossRef] [Publisher link] [Google Scholar]

[32] Olly Beckham et al., "Techniques used to Formulate Confidential Data by Means of Fragmentation and Hybrid Encryption," *International Research Journal of Management, IT and Social Sciences*, vol. 6, no. 6, pp.68-86, 2019. [CrossRef] [Publisher link] [Google Scholar]

[33] Mahboobeh Najmi, "Investigation of Improving Load Balancing and Data Fragmentation in Cloud Computing Performance," *Indian Journal of Science and Technology*, vol. 12, no. 7, pp. 1-9, 2019. [CrossRef] [Publisher link] [Google Scholar]

[34] Nelson Santos et al., "Performance Analysis of Data Fragmentation Techniques on a Cloud Server," *International Journal of Grid and Utility Computing*, vol. 10, no. 4, pp.393-401, 2019. [CrossRef] [Publisher link] [Google Scholar]

[35] L. M. Nithya, and P. Anu Priya, "Data Fragmentation and Duplication in Cloud for Secure Performance," *Indian Journal of Science and Technology*, vol. 12, no. 20, pp.1-6, 2019. [CrossRef] [Publisher link] [Google Scholar]

[36] S. Sangeetha, and P. Suresh Babu, "An Investigation Analysis on Secured Communication with Big Data," *International Journal of Computer Trends and Technology*, vol. 70, no. 8, pp. 15-20, 2022. [CrossRef] [Publisher link]

[37] Sharifah Hafizah Sy Ahmad Ubaidillah, Noraziah Ahmad, and Julius Beneoluchi Odili, "Fragmentation Techniques for Ideal Performance in Distributed Database–A Review," *International Journal of Software Engineering and Computer Systems*, vol. 6, no. 1, pp. 18-24, 2020. [CrossRef] [Publisher link] [Google Scholar]

[38] Riad Mokadem, and Abdelkader Hameurlain, "A Data Replication Strategy with Tenant Performance and Provider Economic Profit Guarantees in Cloud Data Centers," *Journal of Systems and Software*, vol. 159, pp. 110447, 2020. [CrossRef] [Publisher link] [Google Scholar]

[39] Shrutika Khobragade, Rohini Bhosale, and Rahul Jiwane, "High Security Mechanism: Fragmentation and Replication in the Cloud with Auto Update in the System," *Computer Science and Information Technologies*, vol. 1, no. 2, pp. 78-83, 2020. [CrossRef] [Publisher link] [Google Scholar]

[40] P. Peter Jose, and S.P. Victor, "Cloud Storage Security Scheme using Dynamic Data Division and Replication," *Solid state Technology*, vol. 63, no. 6, pp. 9599-9607, 2020. [Publisher link] [Google Scholar]

[41] S. Periyanatchi, and K. Chitra "A Lion Optimization Algorithm for an Efficient Cloud Computing with High Security," *Journal of Scientific Research*, vol. 64, no. 1, pp. 278-284, 2020. [CrossRef] [Publisher link] [Google Scholar]

[42] Priteshkumar Prajapati, and Parth Shah "A Review on Secure Data Deduplication: Cloud Storage Security Issue," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3996-4007, 2020. [CrossRef] [Publisher link] [Google Scholar]

[43] Slim Kallel et al., "Restriction-Based Fragmentation of Business Processes over the Cloud," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, pp. 1-1, 2021. [CrossRef] [Publisher link] [Google Scholar]

[44] Ali Shakarami et al., "Data Replication Schemes In Cloud Computing: A Survey," *Cluster Computing*, vol. 24, no. 3, pp. 2545-2579, 2021. [CrossRef] [Publisher link] [Google Scholar]

[45] Uros Tos et al., "Achieving Query Performance in the Cloud via a Cost-Effective Data Replication Strategy," *Soft Computing*, vol. 25, no. 7, pp. 5437-5454, 2021. [CrossRef] [Publisher link] [Google Scholar]

[46] Quadri Waseem, et al., "Quantitative Analysis and Performance Evaluation of Target-Oriented Replication Strategies in Cloud Computing," *Electronics*, vol. 10, no. 6, pp. 672, 2021. [CrossRef] [Publisher link] [Google Scholar]

[47] Tawfiq S. Barhoom, and Mahmoud Y. Abu Shawish, "Text File Privacy on the Cloud Based on Diagonal Fragmentation and Encryption," *Journal of Engineering Research and Technology*, vol. 8, no. 1, 2021. [CrossRef] [Publisher link] [Google Scholar]

[48] Annette Shajan, and Shanta Rangaswamy, "Survey of Security Threats and Countermeasures in Cloud Computing," *International Journal for Research & Technology*, vol. 2, no. 7, 2021. [Publisher link] [Google Scholar]

[49] Sridhar Reddy Vulapula, and Hima Bindu Valiveti, "Secure and Efficient Data Storage Scheme for Unstructured Data in Hybrid Cloud Environment," *Soft Computing*, pp. 1-8, 2022. [CrossRef] [Publisher link] [Google Scholar]

[50] Yong-Feng Ge et al., "MDDE: Multitasking Distributed Differential Evolution for Privacy-Preserving Database Fragmentation," *The VLDB Journal*, pp. 1-19, 2022. [CrossRef] [Publisher link] [Google Scholar]

[51] Bijeta Seth et al., "Integrating Encryption Techniques for Secure Data Storage in the Cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. 4108, 2022. [CrossRef] [Publisher link] [Google Scholar]

[52] S. Yuvaraman, and D. Saveetha, "Secure Cloud Data Storage and Retrieval System Using Regenerating Code," *In Proceedings of International Conference on Deep Learning, Computing and Intelligence*, pp. 313-321, 2022. [CrossRef] [Publisher link] [Google Scholar]

[53] K. Rajalakshmi, K. Ramesh, and P.N. Renjith, "Comparative Study of Cryptographic Algorithms in Cloud Storage Data Security," *International Conference on Intelligent Technologies*, pp. 1-7, 2022. [CrossRef] [Publisher link] [Google Scholar]

[54] Deepak Kumar Sharma et al., "Modified Minimum Spanning Tree Based Vertical Fragmentation, Allocation and Replication Approach in Distributed Multimedia Databases," *Multimedia Tools and Applications*, vol. 81, pp. 37101-37118, 2022. [CrossRef] [Publisher link] [Google Scholar]