International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Joint Data Integrity and Loss Recovery Mechanism for Secure Storage in Cloud Computing

[1*] K. K. Baseer, [2] Dr. M. Jahir Pasha, [3] Subhashini R., [4] S. Srinivasan, [5] Dr. Jyoti Prasad Patra and [6]V. Jagannatha Reddy

Submitted: 18/09/2023 **Revised**: 19/11/2023 **Accepted**: 28/11/2023

Abstract: Data outsourcing lowers the cost of maintenance and storage, but the user is unaware of the location of their data. Since cloud data is uncontrollable, new security issues must be addressed. Despite much research in the literature, there are still significant problems with secure storage and data integrity for shared dynamic data. Intending to solve these issues, this research develops a data security and integrity methodology that also allows for data loss recovery in the cloud using the efficiency of Machine Learning (ML) algorithms. When a cloud fails due to a disaster, an attack, or data loss and corruption, the data loss recovery process helps to recover the lost data and restore the cloud backup.

Keywords*: Data Integrity, Loss Recovery, Secure Storage, Cloud Computing, Machine Learning*

1. Introduction

Cloud computing is a promising next-generation computing paradigm that primarily relies on technologies such as virtualization, and utility computing. The number of people accessing the internet has dramatically increased recently. To meet the needs of the users, a system with broad access and storage is necessary. Users may more efficiently store and share information online thanks to cloud technologies. People of cloud systems can access and distribute digital data across numerous users all over the world [1][2]. Additionally, users of cloud technology are able to import and export data through web-based object storage, which may contain sensitive data like employee profiles and company information. Even while the cloud offers a number of services and vast storage capacities, the security of data shared online is still susceptible to a number of threats, including brute force attacks and occlusion attacks [3][4].

The main threats to storage security are (i) modifying the stored data to compromise the data integrity (ii) compromising cloud users' privacy by leaking user information, (iii) unauthorized access of stored data (iv) Losing part of data or whole data. (v) data update and consistency maintenance [5]. The traditional cryptographic technology could not be applied as the users lose their control of data storage. Moreover, it is a very tough job to verify the actual data using a verification strategy. Existing solutions should address these threats for efficient storage security [6][7]. Hence an efficient solution is required with the following issues such as data integrity should provide data loss recovery and ensure data consistency, and data confidentiality should protect the privacy of users and unauthorized access with the least storage overhead [8][9].

In order to use cloud computing, Cloud Service Providers (CSPs)are given control over physical data and equipment. The CSPs, however, are typically not reliable. To their advantage, they might hide data loss or inaccuracy from the users [10]. Besides, data from the CSP hosting client must be accessible and cannot be read or changed by unauthorised users [11].Numerous studies have been conducted to improve secure data sharing using key generation methods in order to address the problems with cloud computing. However, it has problems with authentication and trust between the data transmitter and receiver. The key cryptosystem occasionally takes a long time and may result in mistakes [12]. Later, an access control-based secure storage system is presented for safe cloud data sharing [13]. However, the problems with efficacy in terms of computing complexity and lengthy execution times persist [14].

Researchers created a variety of ML techniques to improve cloud secure storage during transmission in an effort to resolve the problems. With the evolution of ML algorithms, plenty of research has been made to eradicate the cloud secure storage issues. Yet, the challenges remain as it needs

[1]*Associate Professor of CSE, GITAM School of Technology,GITAM (Deemed to be University),Bengaluru, Karnataka, India, *baseerphd@yahoo.com*

[2]*Associate professor, Department of Computer Science and Engineering, G Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India, jahir4444@gmail.com*

[3]*Assistant professor, Department of Information Technology, Sona college of Technology, Salem, subhashini.it@sonatech.ac.in*

[4]*Professor, Department of Advanced Computing Sciences, AMET University, Chennai, Tamilnadu, India, srinikcgmca@gmail.com*

[5]*Professor Head EE and EEE, Krupajal Engineering College, KEC Pubasasan Prasanthi Vihar Kausalyaganga Near CIFA District Puri Bhubaneswar, Odisha, India, jpp42003@yahoo.co.in*

[6]*Assistant Professor, Department of CSE, Gandhi Institute of Technology and Management (GITAM Deemed to be University) Bengaluru Campus, Doddaballapura, Karnataka, India, mvjagannathareddy@gmail.com*

to be investigated more [15][16]. With this aim, this paper intends to present a novel ML algorithm to ensure data integrity via encryption and decryption of data with an optimized key selection procedure. The main contribution of this context is given as follows.

- In general, cloud systems lack security for online data sharing. A novel technique is developed to ensure data integrity and loss recovery to increase security while utilising effective computational power.
- To speed up calculation and increase security, an optimum key is first generated using Improved Harris Hawks Optimization (IHHO).
- In order to ensure integrity, it is also necessary to guarantee the accuracy of data shared among the data.
- Besides, the data loss tolerance is accomplished by applying to replacement recovery strategy via the Greedy approach.
- Finally, the significance of the proposed model is verified through a comparative analysis with existing models.

The rest of this article is structured as follows. Section II presents a brief description of the recent cloud secure storage systems and data integrity. The proposed architecture of the novel cloud secure storage systems with the optimization concepts is explained in Section III. Section IV dealsgive the obtained results and their experimentation details. Finally, Section V terminates the paper.

2. Literature

In 2021, Amr M. Sauber[17] developed a cloud data security scheme to prevent data from unauthorized user access. Besides, the unauthorized user identity was achieved by encryption and authentication processes. To safeguard users and data owners against any fictitious illegal access to the cloud, it was designed with the One-Time Password (OTP) as a logging approach and uploading technique.

In 2018, Senthil Kumar, and Latha Parthiban[18] proposeddata integrity as well as data recovery approaches in the cloud. Here, the secure storage was accomplished by the layered architecture of the cloud with various blocks. Furthermore, a replacement recovery strategy was implemented to retrieve the data after failure or attack.

In 2019, Paul R Rejin*et al.* [19] established a secure cloud storage model for data integrity as well as data recovery. The Cloud Data Owner (CDO)divided a ciphertext file into several cipher blocks and distributed them to randomly chosen CSPs. Moreover, a Cloud Data User (CDU) downloaded the corresponding ciphertext file after reconstructing it from the blocks in order to access any file. If the user's attribute set and the application's access policy were compatible, the file was decrypted.

In 2017, Rongzhi Wang[20] developed asecure storage approach for data integrity as well as data recovery using encryption. In order to solve issues and recover lost data, the program used the boot password forthe data encryption problem. It also corrected the Tornado data redundancy code and used a hash keyed to the Tornado code with an error correction function to address the issues.

In 2022, Rose Adee and HaralambosMouratidis[21] introduced a four-step data security scheme using encryption. Initially, the cloud data were protected via encryption processes and then followed by steganography, data recovery, and sharing. Eventually, it attained confidentiality and integrity for data in the cloud.

A. Review

From the literature, it is clear that plenty of research has been made on the secure cloud storage model. Some of the features and challenges of the existing methods are briefly described in this section. The various approaches used for secure cloud data storage attained a certain level of integrity and ensures loss recovery. The OPT strategy was used in [17] to confirm protection from unauthorized access yet, the experimentation was limited to prove the performance of the proposed system. Similarly, the approaches in [18] exposed minimized performance over the Advanced Encryption Standard (AES) model. The CDO and CDU model [19] exposed better data integrity performance but the loss recovery strategy results were limited in accuracy. The Tornado code encryption [20] model exposed improved results however, the process was highly time-consuming and the experimentation was difficult to implement. The four-step data security scheme [21] accomplished confidentiality and integrity still, the loss recovery scheme was not included. From this assessment, it clearly states that there is a need to implement effective secure cloud data storage systems with ML ideas and optimization concepts so as to assure optimal results and performance.

3. A Novel Cloud Secure Storage Model

B. Proposed Architecture

Cloud platforms generally don't provide enough security for online data sharing. In order to strengthen security and recover from data loss, a novel technique is created that makes efficient use of computer capacity. At first, user IDs for each user from the access historyare created. When a DataOwner(DO),stores his data $\mathcal{D}$in the database, the data $\mathcal{D}$are ready to be encrypted. Using IHHO, an ideal key is first created to expedite calculation and boost security.The most topical swarm-based optimization technique, called HHO, mimics Harris' hawks' surprise pounce nature when pursuing prey. HHO usually displays various exploitation and exploration methods. It has a straightforward structure that varies with time, which helps the flow among the main stages.Besides, the correctness of data shared among the data must also be guaranteed in order to maintain integrity. Additionally, the replacement recovery strategy using the greedy approach is used to achieve data loss tolerance. In general, any algorithm that makes the locally optimal decision at each stage when solving a problem is said to be greedy.Finally, a comparison study with other models is used to demonstrate the significance of the suggested model.Fig. 1 represents the systematic depiction of the proposed work.

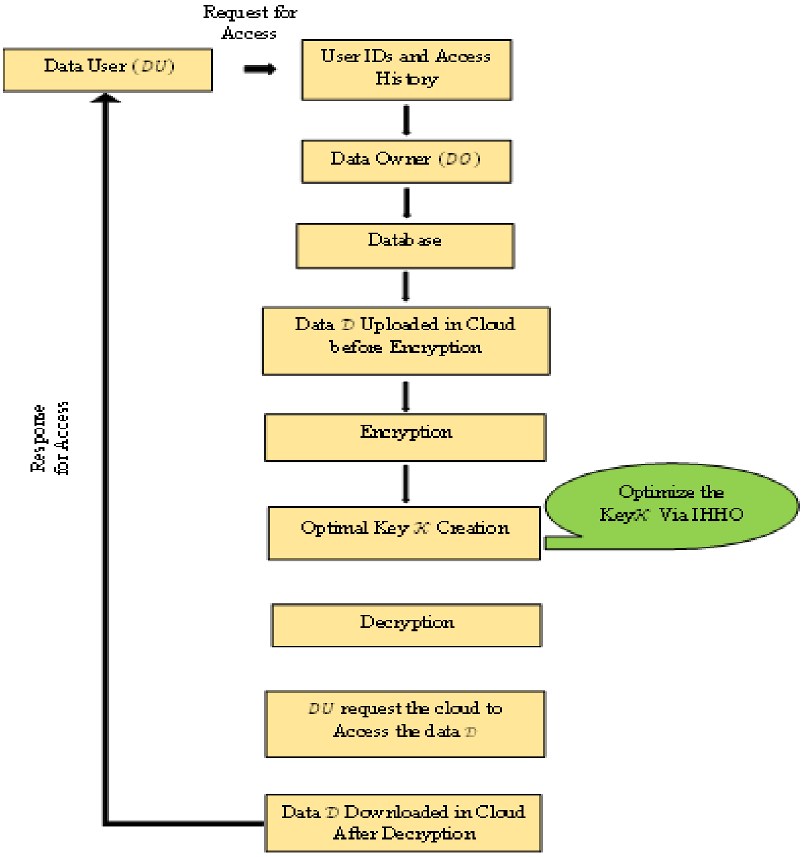
Request for Access
Data User (DU)
User IDs and Access History
Data Owner (DO)
Database
Data $\mathcal{D}$ Uploaded in Cloud before Encryption
Encryption
Optimal Key $\mathcal{K}$ Creation
Optimize the Key$\mathcal{K}$ Via IHHO
Decryption
DU request the cloud to Access the data $\mathcal{D}$
Data $\mathcal{D}$ Downloaded in Cloud After Decryption
Response for Access

Fig. 1. Systematic Depiction of Proposed Model

Initially, the data is split into a secret sequence as $\mathbb{S}$, and the datais to be encrypted $\mathcal{D}$, and the converted sequence $\overline{\mathbb{S}}$make up the three main processes of this model. A secret $\mathbb{S}$ is initially selected from the accessible sequence. Only the sender A and recipient B of the sensitive data$\mathcal{D}$ are aware of this secret $\mathbb{S}$. Additionally, the secret $\mathbb{S}$ is included in the data $\mathcal{D}$,

which turns into $\overline{\mathbb{S}}$. The altered sequence $\overline{\mathbb{S}}$ is now transmitted to the receiver B by sender A together with a collection of multiple codes. Together with $\overline{\mathbb{S}}$, the receiver B gets all the codes and processes them all. Once Bhas identified the precise $\overline{\mathbb{S}}$, it can decrypt the $\overline{\mathbb{S}}$ and obtain the data $\mathcal{D}$. Additionally, the secure storage is constructed using three main stages, similar to traditional systems, including key creation, encryption of DO, and decryption of DataUser(DU). However, the IHHO technique is used to optimize the key created for encryption and decryption. Here, the DOuploads the data using the suggested structure for data sharing. Besides, the uploaded data are secured by an optimum key $\mathcal{K}$ and encrypted using the suggested IHHO encryption. Thesuggested IHHOalgorithm generates the best key for encrypting the message converted into $\overline{\mathbb{S}}$. The framework authenticates the $DU$when the end user requests the data retrieval by checking the $\mathbb{S}$ from the owner level. The DUcan use his best key$\mathcal{K}$ to decrypt the secret data$\overline{\mathbb{S}}$ after the data-sharing framework has retrieved the relevant data from the cloud. The DUcan get the unencrypted version of data $\mathcal{D}$ from this decrypted data. The major three key stages are listed as shown in Fig. 2.

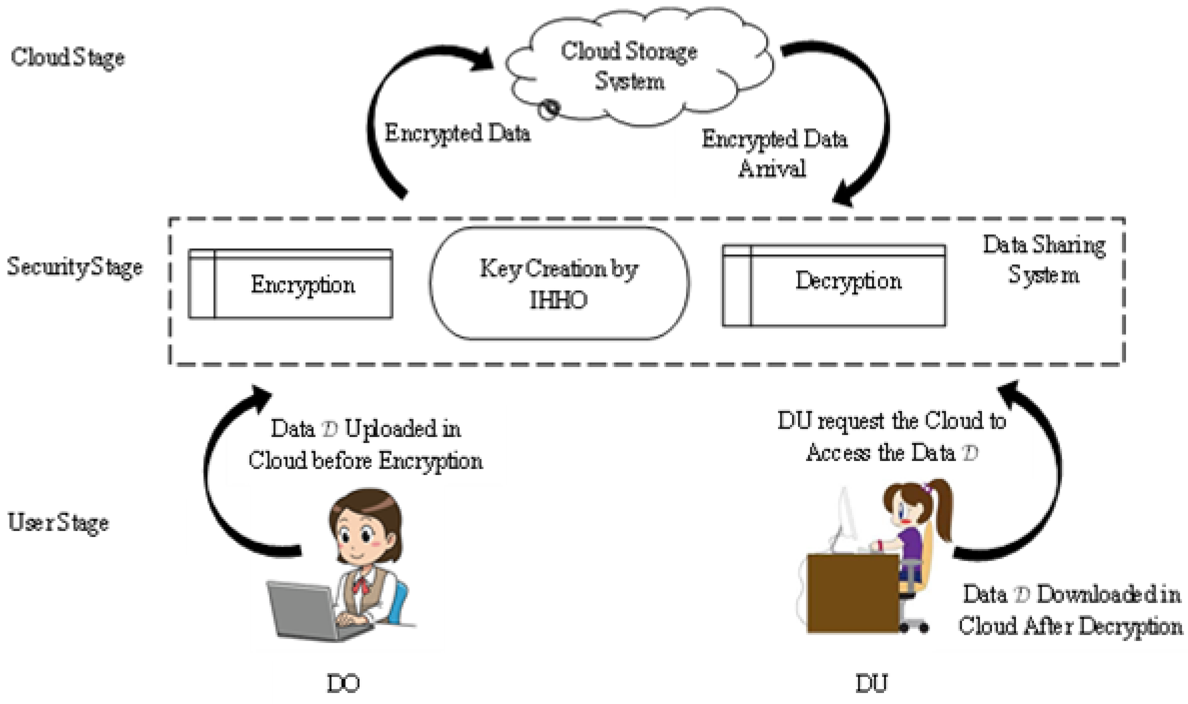

Fig. 2. The Major Three Stages of the Proposed Model

Algorithm 1 shows the pseudocode of proposed encryption model.

```
Algorithm 1: Encryption of data$\mathcal{D}$ using proposed data integrity model
Encrypt the data $\mathcal{D}$   // the data to be uploaded in the cloud
                                 // the conversion of original data $\mathcal{D}$ into secret sequence $\mathbb{S}$
Data is split into a secret sequence as $\mathbb{S}$
Encrypt the secret sequence as $\mathbb{S}$ with the traditional AES algorithm
For $C \leftarrow 1$ to $n(\mathbb{S})$   // $n$ points to number of blocks $C$ in secret sequence as $\mathbb{S}$
{
$C' = En\_AES(C, \mathcal{K})$   //AES encryption via optimal key $\mathcal{K}$ using IHHO model
}
Send $\overline{\mathbb{S}}$ to cloud   // Encrypted data (altered sequence)
End
```

Algorithm 2 demonstrates the pseudocode of proposed decryption model.

```
Algorithm 1: Decryption of altered sequence$\overline{\mathbb{S}}$ using proposed data integrity model
Decrypt the altered sequence $\overline{\mathbb{S}}$   // the data to be downloaded in the cloud
 // the conversion of altered sequence $\overline{\mathbb{S}}$ into secret sequence $\mathbb{S}$
Convert $\overline{\mathbb{S}}$ into $\mathbb{S}$ sequence
```

```
Decrypt the secret sequence $\mathbb{S}$with the traditional AES algorithm
For $C' \leftarrow 1$ to $n(\mathbb{S})$    // $n$ points to number of blocks $C$ in data $\mathcal{D}$
{
    $C = De\_AES(C', \mathcal{K})$   //AES encryption via optimal key $\mathcal{K}$ using IHHO model
}
Send $\mathcal{D}$ to cloud   // Encrypted data
End
```

C. Optimal Key Selection via IHHO

Here, the optimal key is created using the IHHO algorithm. Generally, HHO [22] is used in the suggested procedure because of its effectiveness and ease of use. In HHO, various hawks work together to collaboratively pounce the prey in various directions as they investigate the chasing and fleeing behaviours of prey.The "seven kills" paradigm is the name given to the prey-capturing strategy. The Harris Hawks H hunt the prey via 2 tactics like perch arbitrarily and wait $(r < 0.5)$ or perch on tall trees $(r \geq 0.5)$. Population initialization of the HHO algorithm is given in Eq. (1), where $H(\tau + 1)$ indicates the location vector of H in succeeding iterationτ, $H_{rab}(\tau)$specifies the rabbit location, $H(\tau)$ signifies the present location vector of H, $q_1 . q_2, q_3, q_4$ and r points to an arbitrary number in $(0,1)$, lb and ub mean lower and upper limits, $H_{rand}(\tau)$represents arbitrarily chosen H, H_c portrays the average location of H.

$$H(\tau + 1) = \begin{cases} H_{rand}(\tau) - q_1 |H_{rand}(\tau) - 2q_2 H(\tau)| & (r < 0.5) \\ \left(H_{rab}(\tau) - H_c(\tau)\right) - t_3\left(lb + q_4(ub - lb)\right) & (r \geq 0.5) \end{cases} \tag{1}$$

The average location of H is accomplished using Eq. (2), in which $H_i(\tau)$ specifies the position of every H in τ, and n refers to the overall H count.

$$H_a(\tau) = \frac{1}{n} \sum_{i=1}^{n} H_i(\tau) \tag{2}$$

Normally, a prey's energy minimizes while escaping and this effect is expressed in Eq. (3), in which ε signifies the prey's escaping energy, T portrays iteration count, and ε_0 represents the initial energy between $(-1,1)$. Besides, if $|\varepsilon| < 1$, then the exploitation stage begins whereas if $|\varepsilon| \geq 1$, then the exploration stage begins.

$$\varepsilon = 2\varepsilon_0 \left(1 - \frac{\tau}{\mathrm{T}}\right) \tag{3}$$

If $(q \geq 0.5) \& (|\varepsilon| \geq 0.5)$, then it is soft besiege of H to implement the surprise pounce as given in Eq. (4), and (5), in which $\Delta\, H(\tau)$ points to a difference between rabbit location and present location vector, L indicates the rabbit's arbitrary jump strength and is explained in Eq. (6), where q_5 is an arbitrary number in $(0,1)$.

$$H(\tau + 1) = \Delta\, H(\tau) - \varepsilon |L H_{rab}(\tau) - H(\tau)| \tag{4}$$

$$\Delta\, H(\tau) = H_{rab}(\tau) - H(\tau) \tag{5}$$

$$L = 2(1 - q_5) \tag{6}$$

If $(q \geq 0.5) \& (|\varepsilon| < 0.5)$, then it is hard besiege of H to implement the surprise pounce while the rabbit has low ε as stated in Eq. (7).

$$H(\tau + 1) = H_{rab}(\tau) - \ \varepsilon |\Delta\, H(\tau)| \tag{7}$$

If $(q < 0.5) \& (|\varepsilon| \geq 0.5)$, the rabbit possesses adequate ε, and a soft besiege is performed as shown in Eq. (8).

$$Z = H_{rab}(\tau) - \varepsilon |L H_{rab}(\tau) - H(\tau)| \tag{8}$$

Asymmetrical, fast, and abrupt dives of H are performed to hunt the prey based on Levy Flight (LF) pattern when H notices the prey implement deceptive moves as portrayed in Eq. (9), in which G denotes dimension, and R represents an arbitrary vector of $(1 \times G)$.

$$V = Z + R \times LF(G) \tag{9}$$

Eq. (10) shows the LF expression, in which a, b, h indicate the arbitrary numbers and α represents a constant of 1.5.

$$LF(h) = 0.01 \times \frac{a \times \sigma}{|b|^{\frac{1}{\alpha}}},\ \sigma = \left(\frac{\Gamma(1+\alpha) \times \sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{1+\alpha}{2}\right) \times \alpha \times 2^{\left(\frac{\alpha-1}{2}\right)}} \right)^{\frac{1}{\alpha}} \tag{10}$$

The last soft besiege position update is applied as defined in Eq. (11), in which the Z and V are obtained from Eq. (8) and (9).

$$H(\tau+1) = \begin{cases} Z & if\ f(Z) < f\big(H(\tau)\big) \\ V & if\ f(V) < f\big(H(\tau)\big) \end{cases}$$
(11)

The last hard besiege position update is applied as given in Eq. (12), in which the Z and V are obtained from Eq. (13) and (14).

$$H(\tau+1) = \begin{cases} Z & if\ f(Z) < f\big(H(\tau)\big) \\ V & if\ f(V) < f\big(H(\tau)\big) \end{cases}$$
(12)

$$Z = H_{rab}(\tau) - \varepsilon|LH_{rab}(\tau) - H_c(\tau)|$$
(13)

$$V = Z + R \times LF(G)$$
(14)

At this point, the traditional HHO is improved by replacing the last hard besiege position update in Eq. (12) witha new position update in Eq. (15).

$$H(\tau+1) = \begin{cases} Z & if\ f(Z) < f\big(H(\tau)\big) \\ V & if\ f(V) < f\big(H(\tau)\big) \end{cases}$$
(15)

Hither, the V is attained based on Euclidean distance among H of every iteration using Eq. (16), where $X = x_1, x_2, \dots x_n)$, and $Y = y_1, y_2, \dots y_n)$.

$$V = \sum_{i=1}^{n}(x_i - y_i)^2 \times H_i(\tau)$$
(16)

Here, H is the candidate solution, which is the optimal key $\mathcal{K}$ attained for the data encryption and decryption processes. Using IHHO, the optimal key $\mathcal{K}$ is created.

D. Replace Recovery Strategy via Greedy Approach

The recovery process [23] substitutes an algorithm to recover lost data in the cloud due to disaster, attack, or system crash. All servers store the directory information.In order to replace the files from servers that were destroyed, the proposed system uses directory information. According to this system's streamlined recovery model, there is a recovery solution that has precise parity symbols needed to regenerate lost data symbols for every server crash. Besides, a replacement recovery strategy with high computational efficiency aims to recover from a single disc failure with the least amount of read symbols.The loss of a cloud server's data is recovered using the replace recovery procedure. The data that is kept on the failing server is restored via this approach. This replacement recovery method has a number of goals, including efficient operation in search and recovery and adaptability to varied network systems. As a result, this method discovers a polynomial-complex recovery strategy. The primary server in this operation can house each server's directory querying data. The data are received from servers in accordance with the replace recovery technique method.

This method offers the best cloud efficiency. A greedy approach is a mathematical technique that determines which subsequent step will yield the most glaring benefit in order to find straightforward, straightforward solutions to complex, multi-step problems.Greedy techniques function by building a set of objects iteratively from the fewest feasible component pieces. Recursion is a method of problem-solving where the answer to one problem relies on the answers to other, smaller versions of the same problem. The benefit of adopting a greedy approach is that simpler and easier-to-understand solutions are found for smaller instances of the problem. While the recovery solution is immediately identified, the main goal is to reduce the amount of data read from the remaining discs for recovery and, consequently, the overall length of the recovery effort. In order to maximise the recovery solution, the replacement recovery technique proposed in this research employs a hill-climbing (greedy) strategy.It begins with a workable recovery solution and gradually replaces it with a minimum data-intensive alternative.

4. Simulation Results

E. Experimentation Setup

The proposed secure cloud data storage systemusing the IHHO approach was implemented in MATLAB on Intel core® core i3 processor 7020U@2.3 GHz, 8 GB RAM, 64-bit operating system. Here, the efficiency and novelty of the implemented model were recorded via the simulation results.With varying data sizes between 1 MB and 5 MB, plain text was arbitrarily chosen for data transmission. Additionally, the size of the generated cipher text for each input plaintext was assessed in order to determine the total execution time. The effectiveness of the suggested model was evaluated using different encryption approaches, including, Advanced Encryption Standard (AES) [24], Data Encryption Standard (DES) [25], Grey Wolf Optimizer (GWO) [26],and Particle Swarm Optimization (PSO) [27] models. Here, the evaluation was implemented through various performance parameters such as execution time and throughput.

F. Algorithmic Analysis

The proposed secure cloud data storage systemusing the IHHO approach attained better performance in terms of accuracy and throughput over other encryption techniques. Fig. 3 depicts the accuracy performance of the proposed IHHO encryption model for 1 MB to 5 MB of data $\mathcal{D}$. The proposed framework is a collection of measurements that is a proximate measurement of the exact values used to test the accuracy of the model's dependability. Additionally, it is the attribute of being accurate or exact. Accuracy refers to the extent to which an achieved specification complies with a reference value. Now, accuracy Acc of the encrypted and decrypted data is measured by the formula given in Eq.

(17), in which CD refers to the correct data transmitted and TD specifies the total data transmitted.

$$Acc = \frac{CD}{TD} \tag{17}$$

Here, the proposed framework accomplished better accuracy performance for all data sizes, which is 5.86% better than DES, 4.73% better than AES, 3.09% better than PSO, and 1.44% improved than GWO for 1 MB of data $\mathcal{D}$.

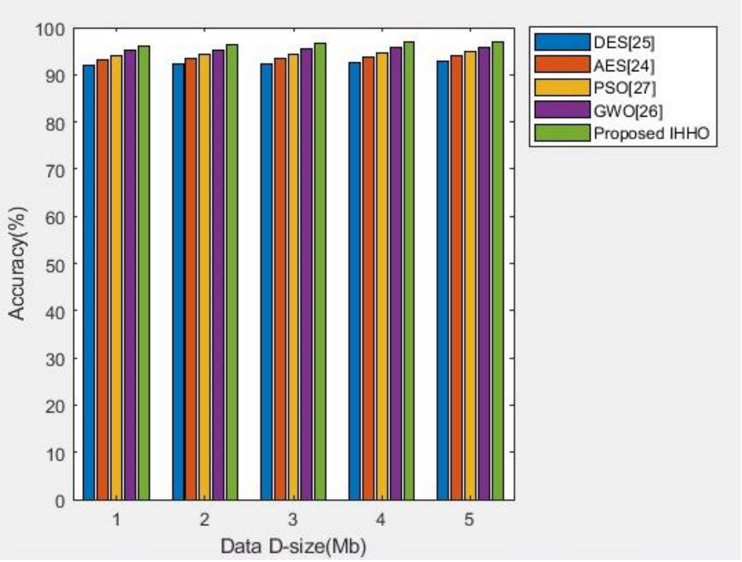

Fig. 3. Accuracy Performance of Implemented IHHO Encryption Model vs other Existing Models

Execution time is usually the fraction of the time necessary to accomplish a task. Besides, it is computed based on the total computation time of cloud data transmitted on the cloud to the overall time needed to encrypt and decrypt the data $\mathcal{D}$ in the cloud. Fig. 4 represents the execution time of implemented IHHO encryption model over other existing models for 1MB to 5 MB of data. Hither, the proposed IHHO model attained better results which is better than the other conventional models. The execution time of the proposed IHHo model is less than one second for all 1 MB to 5 MB of data GWO performed well next to the IHHO model however, the IHHO model reached minimized computation time.

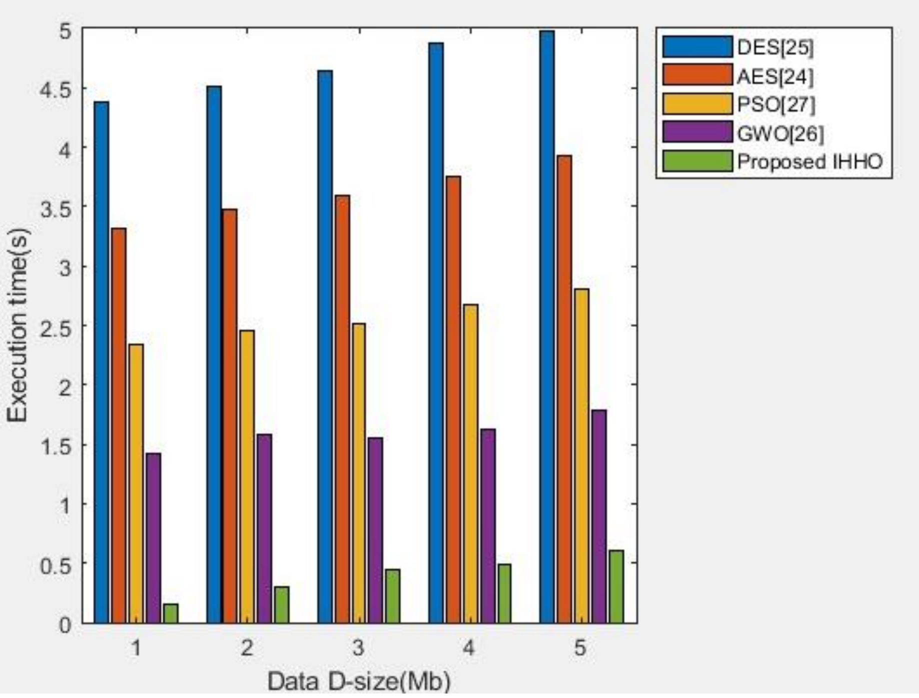

Fig. 4. Execution Time of Implemented IHHO Encryption Model vs other Existing Models

The encryption process takes time because the data is transformed into a secret code that conceals its true intent. Encryption time refers to the amount of time needed to transform secret codes.The throughput of the encryption operation can also be calculated using encryption time. Total plaintext encryption divided by encryption time is used to calculate encryption time. Fig. 5 portrays the encryption time taken by the proposed IHHO model over the other models. Here, the proposed IHHO model achieved improved performance which is far better than other methods since it takes less than 0.5 seconds for data of 1MB to 4MB and 0.6seconds to 5MB of data whereas GWO comes next to IHHO still required 3 times more encryption time than IHHO.

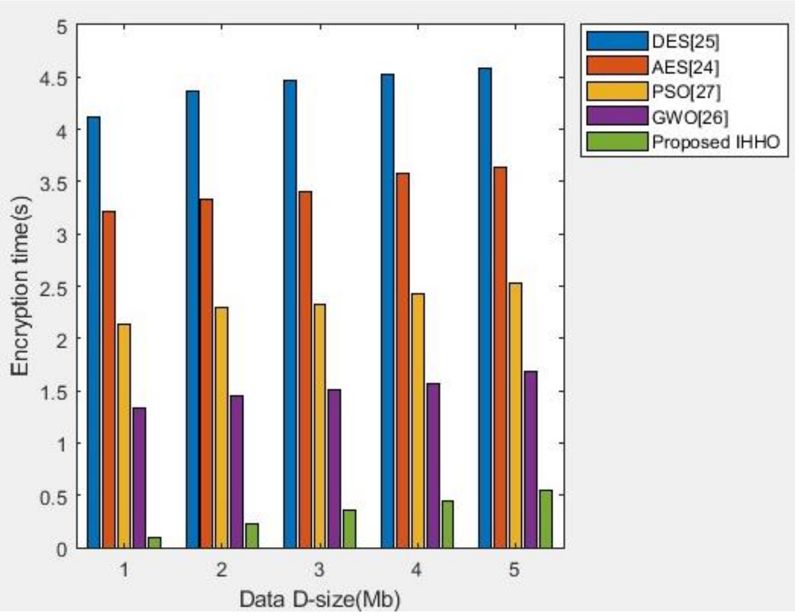

Fig. 5. Encryption Time of Implemented IHHO Encryption Model vs other Existing Models

Decryption, which is primarily a reverse demonstration of encryption, is the process of restoring encrypted material to its original state.Additionally, as decryption requires a secret key, only users with permission can decrypt encrypted data. Decryption time is also the typical amount of time required to open encrypted files or data. Fig. 6 illustrates the decryption time required to decode the encrypted data for the proposed IHHO model and other models. Herein, the proposed IHHO model required less than 0.5 seconds for all data ranges such as 1MB to 5 MB which is far better than other conventional models.

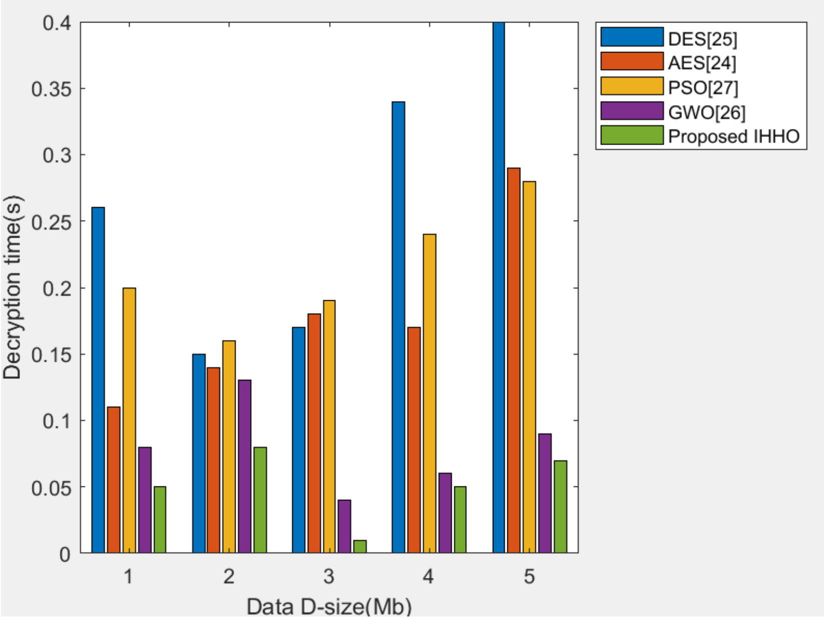

Fig. 6. Decryption Time of Implemented IHHO Encryption Model vs other Existing Models

The higher throughput of the suggested paradigm is what determines its effectiveness. Performance is improved by throughput levels that are higher. The original data size $D_s$and the execution time $E_t$are used to assess the throughput T for the encryption approach as demonstrated in Eq. (18).

$$T = \frac{D_s}{E_t} \quad (18)$$

Fig. 7 shows the throughput performance of the proposed IHHO model over other traditional models. The proposed model attained improved throughput which is 9.23%, 7.18%, 5.12%, and 3.56% improved than DES, AES, PSO, and GWO respectively for 1 MB of data. Similarly, the proposed IHHO outruns all other methods for other data sizes. From this investigation, it is obvious that the proposed IHHO key selection process helps to accomplish improved encryption and decryption processes.

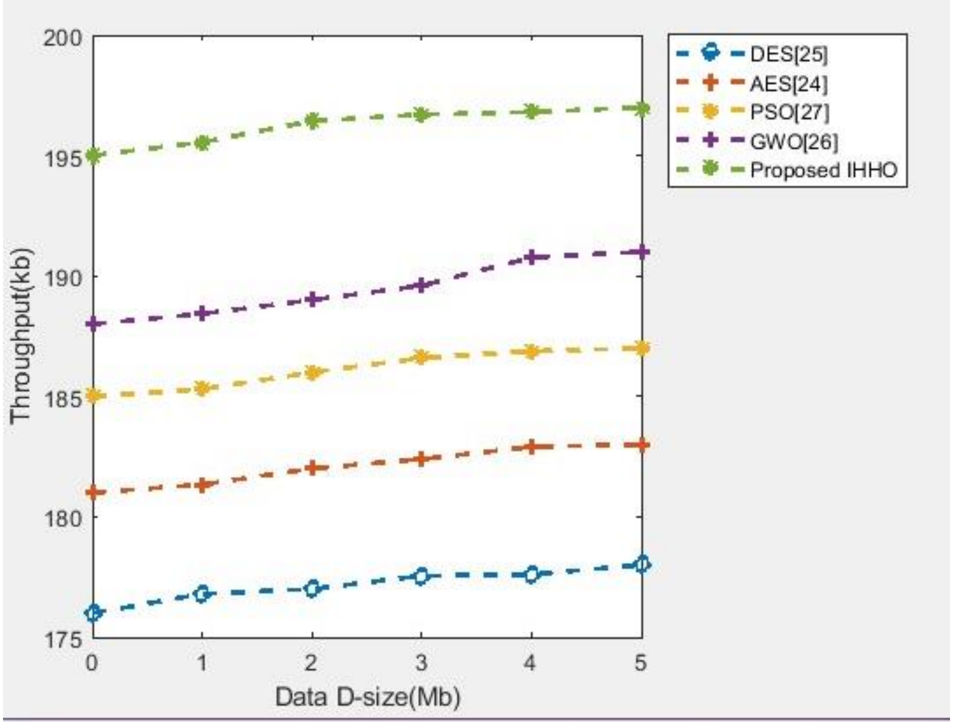

Fig. 7. Throughput Performance of Implemented IHHO Encryption Model vs other Existing Models

5. Conclusion

This paper introduced a revolutionary method that effectively utilized computer capacity to bolster security and recoup data loss. Each user's user ID from the access history was initially formed. The data $\mathcal{D}$were prepared to be encrypted when a DO stores them in the database. An ideal key was initially produced using IHHO to speed up calculations and increase security. The most popular swarm-based optimization method, known as HHO was employed to attain an optimal key selection process.Additionally, in order to maintain integrity, the accuracy of data shared among the data must also be assured. To achieve data loss tolerance, the replacementrecovery strategy utilising the greedy approach was also applied. The importance of the proposed model was then shown through a comparison study with other models concerning accuracy and throughput. The proposed framework accomplished better accuracy performance for all data sizes, which was 5.86% better than DES, 4.73% better than AES, 3.09% better than PSO, and 1.44% improved than GWO for 1 MB of data $\mathcal{D}$. Similarly, the proposed model attained improved throughput which is 9.23%, 7.18%, 5.12%, and 3.56% improved than DES, AES, PSO, and GWO respectively for 1 MB of data. Thus, the proposed model proved its efficiency and significance in secure cloud data storage systems.

References

[1] G. B. Tarekegn, G. Abadi Maru, and H. Zelalem Liyew, “Privacy and security issues IN cloud computing,” International Journal Of Current Research, vol. 8, no. 7, pp. 34894–34898, 2016.

[2] A. Venkatesh and . M. S. Eastaff, "A Study of Data Storage Security Issues in Cloud Computing," International Journal of Scientific Research in Computer Science, Engineering and Information Technology , IJSRCSEIT , vol. Vol. 3, No. 1, pp. 2456-3307, 2018.

[3] O. O. Olakanmi and A. Dada, “An efficient privacy-preserving approach for secure verifiable outsourced computing on untrusted platforms,” International Journal of Cloud Applications and Computing, vol. 9, no. 2, pp. 79–98, 2019.

[4] S. Basu, A. Bardhan , K. Gupta , P. Saha, M. Pal and M. Bose , "Cloud computing security challenges & solutions - A survey," in IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018.

[5] E. Alsolami, “Security threats and legal issues related to cloud based solutions,” IJCSNS International Journal Of Computer Science And Network Security, vol. 18, no. 5, 2018.

[6] F. M. Awaysheh, M. N. Aladwan, M. Alazab, S. Alawadi, J. C. Cabaleiro, and T. F. Pena, “Security by design for big data frameworks over cloud computing,” IEEE Transactions On Engineering Management, vol. 99, 2021.

[7] H. Yusuf and S. Selvan, "Confidentiality Issues in Cloud Computing and Countermeasures: A Survey," in National Conference On Emerging Computer Paradigms 2016, NMAMIT, Nitte, 2016.

[8] Sadavarte, Rajesh and Kurundkar, Gajanan, "Data security and integrity in cloud computing : Threats and Solutions", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 356-363, 2020.

[9] Abdullatif Ghallab Saif, Mohammed and Mohsen Abdulqader, "Data Integrity and Security in Distributed Cloud Computing-A Review", Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, pp.767-784, 2021.

[10] Ait, Almokhtar, Ammari Najim, Abou Anas, Ait Abdellah, and De Mina, "New mechanism for Cloud Computing Storage Security", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.

[11] 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, 2022.

[12] N.K. Sehgal, P.C.P. Bhatt, and J.M. Acken, "Future trends in cloud computing", Cloud Computing with Security, pp. 235-259, 2020.

[13] J. Hong, K. Xue, Y. Xue, W. Chen, D. S. L. Wei, N. Yu, and P. Hong, ‘‘TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud,’’ IEEE Trans. Services Comput., vol. 13, No. 1, pp. 158–171, Jan. 2020.

[14] P. Yang, N. Xiong,and J. Ren, "Data security and privacy protection for cloud storage: A survey", IEEE Access, Vol. 8, pp. 131723-131740, 2020.

[15] E. Hesamifard, H. Takabi, M. Ghasemi, and N. W. Rebecca, ‘‘Privacy preserving machine learning as a service,’’ Proc. Privacy Enhancing Technol., vol. 2018, no. 3, pp. 123–142, Jun. 2018.

[16] P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, ‘‘Privacy-preserving machine learning with multiple data providers,’’ Future Gener. Comput. Syst., vol. 87, pp. 341–350, Oct. 2018.

[17] Amr M. Sauber, Passent M. El-Kafrawy, Amr F. Shawish, Mohamed A. Amin, and Ismail M. Hagag, "A New Secure Model for Data Protection over Cloud Computing", Computational Intelligence and Neuroscience, vol. 2021, 2021.

[18] Senthil Kumar, and Latha Parthiban, "Secure Data Storage In Cloud With Enhanced Public Verifiability And Data Loss Recovery", International Journal of Pure and Applied Mathematics, Vol. 119, No. 16, pp. 1979-1987, 2018.

[19] Paul R Rejin, Raj D Paul, Amir H. Alavi, "Verification of data integrity and co-operative loss recovery for secure data storage in cloud computing", Cogent Engineering, Vol. 6, No. 1, 2019.

[20] Rongzhi Wang, "Research on data security technology based on cloud storage", Procedia Engineering, Vol. 174, pp. 1340 – 1355, 2017.

[21] Rose Adee and Haralambos Mouratidis, "A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography", Sensors, Vol. 22, 2022.

[22] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen, "Harris hawks optimization: Algorithm and applications", Future Generation Computer Systems, Vol. 97, pp. 849-872, 2019.

[23] Dr.V.Dheepa, and Ms.G.Prathima, "Replace Recovery Algorithm Approach In Recovering A Failure In Cloud Environment", Hindco journals, Available Online.

[24] Rashmi Ramesh Rachh, Ananda Mohan P.V,and Anami B.S, “Efficient Implementations for AES Encryption and Decryption”, Circuits, Systems, and Signal Processing, Vol.31, pp.1765-1785, 2012.

[25] Indumathi Saikumar, “DES- Data Encryption Standard”, International Research Journal of Engineering and Technology,Vol. 4 No.3, 2017.

[26] Seyedali Mirjalili, Seyed Mohammad Mirjalili and Andrew Lewisa, "Grey Wolf Optimizer", Advances in Engineering Software, Vol. 69, pp 46-61, March 2014.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, vol.4, pp. 1942-1948, 1995.