Secure Privacy-Preserving using Enhanced Advanced Encryption Standard and Heuristic algorithm in Cloud Computing

Sulthana ASR, Dr. K. Juliana Gnanaselvi
Research Scholar, Rathinam college of arts and science
Head of IT Department Rathinam college of arts and science

Abstract: **With the rapid development of cloud computing technology, the data outsourcing service model in cloud environment is becoming increasingly popular. However, while cloud storage technology provides users with efficient services, its complex architecture also brings significant challenges to data privacy security. Because cloud service providers are not completely trusted, there is a risk of tampering or leaking user data, and third-party audits or malicious attacks by unauthorized users may also lead to data damage or loss. Therefore, o overcome the abovementioned problem, the privacy protection and integrity audit of data in the cloud environment has become an important research topic. In this work, Improved Cat Swarm Optimization (ICSO) and Enhanced Advanced Encryption Standard (EAES) algorithm is proposed for security and privacy preservation on cloud. The cloud data backup and recovery is done by using the ICSO algorithm. It generates best fitness values for data protection in cloud. Then, the secured data transmission is carried out to verify data integrity and privacy with EAES algorithm. The encryption and decryption processes are done using the public and private keys. Only the private key is employed to quickly decrypt messages that have been encrypted with the public key. Finally, dual authentication is done to enhance the system's security. From the result, it concluded that the proposed ICSO-EAES algorithm provides better performance in terms of computational cost, communication cost and execution time rather than the existing algorithms.**

Key words**: Cloud computing, privacy preserving, security, Improved Cat Swarm Optimization (ICSO) and Enhanced Advanced Encryption Standard (EAES)**

1. INTRODUCTION

With the rapid development of information technology, cloud computing technology has gradually become an important support for data processing and storage in various industries. However, with the outsourcing of large amounts of data to the cloud, the issue of data privacy protection has become increasingly prominent. It is very important to build an efficient and secure audit framework for data privacy protection to ensure the data security of tenants. Cloud services benefit countless users worldwide due to notable features, such as on-demand self-service, scalability, easy maintenance, etc. Secure storage and access to data in the cloud is critical. Cloud Identity and Access Management (IAM) service, which acts in a centralized way to provide access requests to the authenticated users [1]. Controlled access sometimes fails to preserve the privacy of the sensitive information stored in the cloud due to several reasons, such as insider attacks, breaches of data security, or any other types of unauthorized access.

Cloud servers with greater storage and computing capabilities supplied by Cloud Service Provider (CSP) are favoured by more and more users. However, once data file is outsourced, the Data Owners (DOs) will lose direct control of personal file. The software error, hardware failure, such as power outages in the server room, internal or external malicious attacks will cause data corruption, and DOs do not learn that the data corruption incident has occurred. Some CSP even hide data corruption or data loss from DOs to maintain their interests and reputations [2] [3]. To save storage space on specific servers, some CSP advertently remove files which aren't accessed lately by some DO from storage block. Besides, hacker attacks and intentional sabotage from third parties may also cause integrity corruption even data loss. To address these issues, the DOs need to check periodically whether outsourced data file has been completely stored in the storage servers provided by CSP. Therefore, various cloud storage data integrity auditing protocols with distinctive features gradually emerge.

Authentication is to check the identity of users, and access control is to constrain what a user can do. Generally, the two procedures are separate, i.e.,

verifying the identity first and then assigning permissions. If the two procedures could be integrated reasonably, it would greatly improve the user experience of cloud services. To achieve this goal, several access schemes integrated with authentication and hierarchical access control (hereafter called unified access scheme) [4] have been used, which provide new thought and reliable technical support for access control. The schemes mainly divided into two types of modes: the centralized mode and the distributed mode. The centralized mode has smaller storage resource consumption and is easier to manage, which makes it more suitable for an organization with several different services. Obviously, the centralized mode is suitable for this application scenario.

Public auditing introduces a TPA to complete the data integrity auditing task on behalf of the users, which greatly reduces the computation burden of the users. The user pays the TPA based on the workload of the auditing services provided by the TPA [5]. The more files the TPA audits, the higher fee the user has to pay. A recent survey shows that by 2025, each user will hold 15 terabytes of data (volume of data/information created). When such massive data is migrated to the cloud for storage, the users need to pay expensive fees to the TPA if the TPA verifies the integrity of all cloud data in each auditing task. In addition, it will lead to a waste of computational resources. As a result, the user expects the TPA to check the integrity of high-value data more frequently than the integrity of low-value data. Achieving different auditing frequencies in public auditing is very important.

A potential method of solving the above problem is to tell the TPA which files are of high value. The TPA can frequently execute auditing tasks for the valuable files, while performing less for low ones. However, using this method, the user's privacy is leaked to the TPA. Furthermore, during the phase of auditing, the TPA may obtain the user's real data content by challenging the same data block multiple times [6] [7]. Once the TPA knows which files are valuable, it can repeatedly challenge these files to obtain the user's private data. Therefore, it is essential and valuable to research how to complete public auditing supporting different auditing frequencies on the condition that the user's privacy is protected.

The aim of this research is to improve the privacy preserving and data recovery in cloud environment using ICSO-EAES algorithm. Existing approaches suffer from drawbacks related to communication cost and the attacks for cloud data. To address these challenges, this research introduces the ICSO-EAES algorithm which is aimed at enhancing the overall performance of cloud by means of security as well as integrity. The proposed ICSO-EAES method is focused to improve the communicational cost, execution time and computational cost. Thus, it is used for increasing the security as well as data protection effectively over cloud computing.

The paper is organized as follows in the parts that follow A summary of the literature on public auditing, secured data transmission and privacy preservation on cloud data is provided over Section 2. Section 3 elaborates on the proposed methodology, namely the ICSO-EAES method. Section 4 presents the simulation results and engages in a discussion on performance analysis. Lastly, Section 5 summarizes the conclusions drawn from the study.

2. RELATED WORK

In [8], Gan et al (2018) suggested an efficient auditing scheme for outsourced big data based on algebraic signatures and an XOR-homomorphic function, that can achieve numerous advantages, such as fewer challenges and proofs, non-block verification, data privacy preservation, and lower computational and communication costs. The proposed scheme enables a trusted third-party auditor, on behalf of DOs, to audit the outsourced data in a cloud. Thus, reducing the computational burden on the DOs. Subsequently, we construct a new data structure called a Record Table (RTable) and extend the basic auditing scheme to support the data dynamic operations. As our extended scheme does not use public key encryption, the entire process of updating the data incurs only a small computational and communication overhead with regard to the auditor, the DOs, and the cloud server.

In [9], Hou et al (2018) introduced identity-protected secure auditing and deduplicating data scheme in this work. In the scheme, the cloud cannot learn any useful information on the relationship of data owners. Different from existing schemes, the cloud does not need to store the file-owner link for supporting valid data downloading. Instead, when the user downloads the file, he only needs to anonymously submit a credential to the cloud, and can download the file only if this credential is valid. Except this main contribution, our scheme has the following advantages over existing schemes. First, the proposed scheme achieves the constant storage, that is, the

storage space is fully independent of the number of the data owners possessing the same file. Second, the proposed scheme achieves the constant computation. Only the first uploader needs to generate the authenticator for each file block, while subsequent owners do not need to generate it any longer. As a result, our scheme greatly reduces the storage overhead of the cloud and the computation overhead of data owners. The security analysis and experimental results show that our scheme is secure and efficient.

In [10], Zhang et al (2019) discussed auditing scheme supporting publicly integrity checking and dynamic data sharing with multi-user modification, which aims at allowing multiple cloud users to modify data while ensuring the cloud data's integrity. Also recently Yuan et al. proposed a public Proofs Of Retrievability (POR) in cloud with constant cost, they showed their scheme is the first POR scheme which can simultaneously achieve public verifiability, constant communication and computational costs on users, and prove the security of their scheme. However, in this work, it shows their schemes are not secure, concretely, the tags in their schemes can be easily forged. It also gives an improved fuzzy cloud auditing scheme for the data owners.

In [11], Pawar et al (2023) develop authentication and data security model in cloud computing. This method includes six various units, such as cloud server, data owner, cloud user, inspection authority, attribute authority, and central certified authority. The developed privacy preservation method includes several stages, namely setup phase, key generation phase, authentication phase and data sharing phase. Initially, the setup phase is performed through the owner, where the input is security attributes, whereas the system master key and the public parameter are produced in the key generation stage. After that, the authentication process is performed to identify the security controls of the information system. Finally, the data is decrypted in the data sharing phase for sharing data and for achieving data privacy for confidential data. Additionally, dynamic splicing is utilized, and the security functions, such as hashing, Elliptic Curve Cryptography (ECC), Data Encryption Standard-3 (3DES), interpolation, polynomial kernel, and XOR are employed for providing security to sensitive data.

In [12], Shen et al (2021) employing the proxy re-encryption algorithm and Oblivious Random Access Memory (ORAM), a privacy-preserving and untraceable scheme is proposed to support multiple users in sharing data in cloud computing. On the one hand, group members and a proxy use the key exchange phase to obtain keys and resist multiparty collusion if necessary. The ciphertext obtained according to the proxy re-encryption phase enables group members to implement access control and store data, thereby completing secure data sharing. On the other hand, this article realizes data untraceability and a hidden data access pattern through a one-way circular linked table in a binary tree (OCLT) and obfuscation operation. Additionally, based on the designed structure and pointer tuple, malicious users are identified and data tampering is prevented. The security analysis shows that the protocol designed in this article can meet the security requirements of proxy re-encryption and ORAM. Both theoretical and experimental analyses demonstrate that the proposed scheme is secure and efficient for group data sharing in cloud computing.

In [13], Bingu et al (2024) suggested cloud auditing with secure file maintenance to retain the duplicate copy of the encrypted file. During the process of public auditing, integrity is maintained for each copy with reduced storage. Secure Authentication with User Anonymity (SAUS) is established with cloud auditing and confidentiality during the process of duplication, and auditing integrity is maintained with encryption approaches. The duplication process is invoked with Multi-Failure Resilient Replication (MRR) procedure in which the data loss is reduced. In order to deal with the multiple auditing strategy for outsourced data, data integrity is maintained with each data owner and delegating Third Party Auditing (TPA) independently. The performance of the work is compared with the existing approaches, which shows a better trade-off in contrast to prevailing approaches.

3. PROPOSED METHODOLOGY

Cloud storage is an essential service of cloud computing, which allows Data Owners (DOs) to move data from their local computing systems to the cloud. Moreover, in cloud storage system, none of the sides, namely DO and cloud storage provider, is guaranteed to provide unbiased auditing result. Therefore, TPA is an intuitive choice for cloud storage auditing to save computation and communication cost. TPA provides a clear yet cost effective method for setting up the trust between DO and cloud server. In order to execute continuous integrity verification without the DOs local copies of

the data and an efficient and novel auditing method is introduced for securing data in cloud storage. This method allows the auditor to check for the data availability and correctness in cloud storage, with minimum computational cost on the auditor and server when compared with the state of art data auditing methods. Furthermore, this work extends the auditing method by utilizing a data structure that permits the auditor to execute dynamic operation like insert, delete, append and modify efficiently with fewer computational cost on DO and cloud storage server

ICSO-EAES Model

The proposed system model Fig 1 consists of entities such as DOs, Cloud Storage Providers (CSP), TPA, and Authorized Users (AUs)

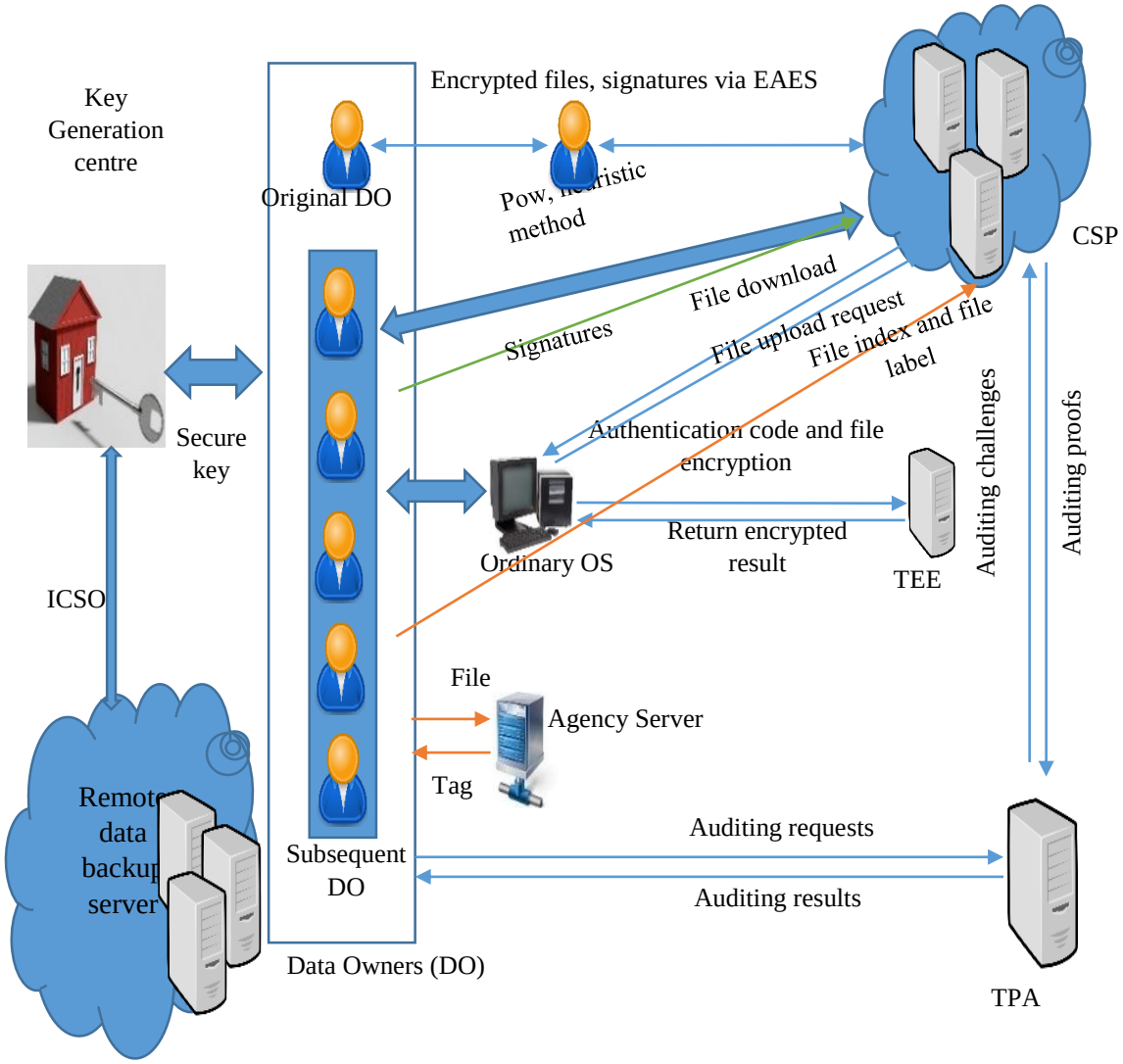

Fig 1 Overall block diagram of the proposed system

3.1 System model

Data Owner (DO): It is a resource bounded entity which generates gargantuan amount of data to be outsourced to remote cloud storage. They depend on cloud storage providers for data maintenance and computation thus minimizing the storage cost and utilizing the esources of the service providers. The DOs are allowed to insert, modify, append or delete their data if required. DOs can be either individual customer or commercial organization

Cloud Service Provider (CSP): This service provider, which has unlimited storage space and highly capable computational resources. It provides the storage space to DOs and is responsible for managing and maintaining the servers and data. CSP is an untrusted entity of hardware and software resource cluster.

Third party auditor (TPA): It is a trusted and reliable entity, with the expertise, capabilities and resources to perform data integrity auditing service on outsourced data under the deputation of DO. It is

trusted by both the DO and CSP. TPA eases the burden of DO for security upkeep. Metadata manager of TPA is responsible for collection of metadata from the DO and indexes the basic information about the data in the index table. This enables the navigation and search process of a specific data to be at ease

Authorized user (AU) users: AU possibly customers or co-workers of DO, who have been authorized to access stored data. They can be either inside or outside the cloud.

Cloud controller (CC): the CC checks for deduplication entries and moves the non-duplicated data from the DO to the cloud storage servers. The index for the deduplication data is updated with modification in the address pointers and the storage index table is generated for the data stored by the storage manager. The query generator in CC is responsible for generating query based on the scheduling policy in Table 1 and the proof as the response for the request is generated and send to the requester. The verifier in TPA checks if the received proof is valid or not and if any misbehaviour found will raise the audit report to the DO for further action. Initially the client uploads the data in the required file format and using random key generator the client generates the key to verify the data.

The key generation centre: It generates the keys using pseudo random generator function PRG(.). The file to upload is split into smaller units as blocks $b_1, b_2, \ldots, b_n$. This EDICE encryption enables privacy of the user's data blocks and also enables efficient deduplication. After the entire data blocks are encrypted the tag for each data block is generated which will be used for integrity verification. The client sends the data to the CSP to store it in their server premises. The CSP controller receives the data and checks for the deduplication data entries if any. If there exists any duplicate entry of the blocks, then discard the duplicated blocks and stores the unique data blocks. The duplicated blocks will have a pointer to the original data blocks storage address

Remote data backup: The main objective of this proposed optimization model is using HPSOABC for efficient Disaster Recovery in a cloud environment, that shorten the data recovery time and data backup cost as much as possible.

Threat model: The security goal in the proposed scheme is to fully resist the brute-force dictionary attacks and provide strong privacy protection for the users' files. Here consider two types of adversaries: internal adversary and external adversary. The malicious cloud or the AS plays the role of the internal adversary, which is honest-but-curious. That is to say, the cloud performs the deduplication protocol honestly but tries to cheat the user about the data corrupt event or derive the file's content from the encrypted file. The AS faithfully performs the assigned operations, but intents to guess the contents of users' files and do not consider the collusion of the cloud and the AS. The user, who somehow knows the file's content or the file's hash value, acts as the external adversary. He intends to obtain a link of a file but does not keep the corresponding file

3.2 Data Recovery Process using ICSO algorithm in cloud

The cloud data backup and recovery is done by using the ICSO algorithm. The data are considered as solutions and those solutions are taken as an initial particle. Fitness value is calculated by using the following equation (1)

The position (P_i^c) and Velocity (V_i^c) ith cat in the C-dimensional space can be defined as $P_i^c = \{P_i^1, P_i^2, \ldots . P_i^C\}; V_i^c = \{V_i^1, V_i^2, \ldots . V_i^C\}$

Where the best position of the cat is represented using $P_{best}^c = P_{best}^1, P_{best}^2, \ldots . P_{best}^C$

The velocity and position of the jth cat are computed using (1) and (2).

$$V_{i\,new}^C = w * V_i^c + a * r * (P_{best}^c - P_i^c) \qquad (1)$$

where, $V_{i\,new}^C$ represents the updated velocity of ith cat in the cth dimension, w denotes a weight factor in the range of 0 and 1, V_i^c represents the old velocity of the jth cat, a is a user defined constant, r denotes a random number in the range of 0 and 1, P_{best}^c represents the best position achieved by jth cat in dth dimension, and P_i^c denotes the current position of the ith cat in cth dimension where $c = 1, 2, \ldots C$.

$$P_{i\,new}^C = P_i^c + V_i^c \qquad (2)$$

where, $P_{i\,new}^C$ denotes the updated position of the ith cat in cth dimension, P_i^c denotes the current position of the ith cat in cth dimension and V_i^c represents the velocity of the ith cat. In cloud, data are stored continuously on a Virtual Machine (VM) location. Based on the fitness value of the data, the retaining of data is decided. First tasks are scheduled and related resources are allocated to the data which is stored in the cloud. So that the data from the cloud can be accessed at any time. During the data recovery procedure, the following steps describe the execution flow

1. A data recovery request is the initial step from the user to the data disaster recovery service provider CSP. This request is transmitted via cloud interface.

2. The second step is an immediate authentication process once the request reaches the CSP. It authenticates the user's details like privilege and account; and neglect the request once it is found to be not the legal one.
3. In order to find the backup locations of the data, every CSP has its own recovery manager that found the data in the Meta database.
4. If the CSP find the data in the database in three different locations, then it compares all the location and selects the one with highest recovery bandwidth.
5. Consider the selected location be CSP_x, and the recovery manager of CSP_x alerts the CSP to made the temporary access authorization which can be used by the recovery proxy of the user who send the request in begin to read the replica. Once the authorization started, the CSP_x informs CSP.
6. The recovery proxy of the user who gave request receives the authorization information and data location from the CSP and the data from the CSP_x is taken from its recovery proxy and intimates CSP.
7. Once the data is retrieved from the CSP_x, it abolish the temporary access authorization which was crested before according to the instruction of CSP_x. Then it records the information regarding the user request for later use.
8. In parallel, the CSP_x analyze and monitor the traffic consumption of the request to charge the CSP when needed

In this work, the optimal data are recovered through the ICSO algorithm to improve the privacy protection over cloud. The CSO algorithm is developed based on the common behavior of cats. It has been found that cats spend most of their time resting and observing their environment rather that running after things as this leads to excessive use of energy resources. To reflect these two important behavioural characteristics of the cats, the algorithm is divided into two sub-modes and CSO refers to these behavioural characteristics as ―seeking mode‖ and tracing mode‖, which represent two different procedures in the algorithm. Tracing mode models the behavior of the cats when running after a target while the seeking mode models the behavior of the cats when resting and observing their environment [14]

Each cat represents a solution set, which has its own position, a fitness value and a flag. The position is made up of M dimensions in the search space and each dimension has its own velocity; the fitness value depicts how well the solution set (cat) is; and finally, the flag is to classify the cats into either seeking or tracing mode. Thus, we should first specify how many cats should be engaged in the iteration and run them through the algorithm. The best cat in each iteration is saved into memory and the one at the final iteration will represent the final solution.

3.2.1. Seeking Mode: Resting and Observing

The seeking mode describes the resting skill of cats. In seeking mode, a cat moves to different positions in the search space, but remains alert. It can be interpreted as local search for the solutions. The following notations are used in this mode.

- Seeking Memory Pool (SMP): This parameter describes the number of copies of a cat to be replicated.
- Seeking Range of selected Dimension (SRD): It denotes the difference between new and old dimensions of cat selected for mutation.
- Counts of Dimension to Change (CDC): It represents the number of dimensions a cat position undergone for mutation.

The steps of seeking mode of CSO algorithm are given as follows.

1. Define the number of copies $(T)\ of\ i^{th}$ cat
2. According to CDC parameter, do the following

a. Randomly add or subtract SRD values from current position of cats
b. Replace the old values for all copies

3. Compute the fitness for all copies
4. Choose the best candidate solution and deploy at the position of i^{th} cat.

3.2.2 Tracing mode

This mode reflects the hunting skill of cats. When a cat hunts the prey, the position and velocity of cat are updated. So, a large difference occurs between new and old positions of cats. Mixture Ratio (MR) is used to combine the seeking and tracing modes of the CSO algorithm. MR is designed to determine the number of cats in seeking and tracing modes. The steps of the CSO algorithm are as follows.

1. Initialize the population of cats.
2. Define the user-defined parameters and numbers of cats in seeking mode and tracing mode according to MR parameter value.
3. Compute the fitness function for each cat and memorize the best position.
4. According to flag:

• If cat is in seeking mode, apply the seeking mode process.
• Otherwise, apply tracing mode process.

5. Again set the number of cats in tracing and seeking modes according to the MR parameter.
6. Repeat steps 3–5 until the termination condition is satisfied.

The CSO algorithm suffers from premature convergence due to its weak diversity hence to overcome the issue the concept of inertia weight is introduced [15]. Moreover, it is also stated that in CSO algorithm, positions of cats are updated by using the current positions and velocities of cats. Sometime, the algorithm fails to explore optimal solutions due to the lack of information regarding global best position of cat. Hence, to deal with these issues, the following modifications are proposed in the CSO algorithm.

The ICSO algorithm

To explore more promising solution and enhance the convergence rate, the global best position of cat is used to guide the positions of cats in tracing mode. Hence, a new modified search equation is proposed for tracing mode of CSO algorithm which includes the global best position of cat.

$$P_{i\,new}^{C+1} = (1-\alpha) * P_i^c + \alpha * X_g + V_i^c \qquad (3)$$

The CSO algorithm uses a velocity vector and previous position of cat to update the position of a cat in tracing mode. The updated position of a cat is only influenced by velocity vector. Hence, to improve the diversity of CSO algorithm, especially in tracing mode, a new velocity updated equation is proposed, which is inspired from

$$V_{i\,new}^{C+1} = V_i^c + \alpha\,(X_g - P_i^c) + \beta * \delta \quad (4)$$

where, δ is a random vector uniformly distributed in the range [0, 1]; α and β are acceleration parameters used to direct the position of a cat toward local and global best positions and X_g presents the global best position of a cat. To make the balance between the exploration and exploitation processes, both of acceleration parameters α and β act as control parameters. The β parameter acts as decreasing function, whereas α parameter serves as an increasing function. In this work, the values of both the parameters are adaptive and computed using the following equations.

$$\beta(T) = \beta_{max} - \left\{\frac{\beta_{max}-\beta_{min}}{T_{max}}\right\} * T \qquad (5)$$

In (5), β_{max} and β_{min} present the upper and lower limits, T_{max} denotes the maximum number of iterations and T denotes the current iteration number. Hence, $\beta(T)$ is a step function whose value ranges between upper and lower limits. The larger value of α supports exploration whereas small values support exploitation. The aim of $\beta(T)$parameter is to control the exploration process of cats in search space.

$$\alpha(T) = \alpha_{min} + (\alpha_{max} - \alpha_{min}) sin\left\{\frac{\rho T}{T_{max}}\right\} \qquad (6)$$

In (6), α_{min} and α_{max} denote the minimum and maximum values of first and last iterations respectively. T_{max} presents the maximum number of iterations and t denotes the current iteration number. The reason behind the incorporation of $\alpha(T)$ parameter is to influence the global exploration ability of the proposed algorithm. A large value of parameter strengthens the global best position of cat and also tends to the solution refinement.

3.2.3 Average-Inertia Weighted CSO

In the pure CSO, a condition on the velocity equation should be put in order to control the velocities of the cats for every dimension and check whether the velocities are in the range of maximum or not. For modifying this part, a parameter as an inertia weight to handle this problem will be used. Here the value of inertia weight (w) will be chosen randomly and experimental results indicate that it is better to choose w in the range of $[0.4, 0.9]$. So selecting the largest value for w in the first iteration $(w = 0.9)$ and then it will be reduced to 0.4 in the next iterations.CSO with inertia weight can converge under certain conditions even without using vmax.

For $w > 1$, velocities increase over time, causing cats to diverge eventually beyond the boundaries of the search space. For w<1, velocities decrease over time, eventually reaching 0, resulting convergence behavior. So the new global best position update equation can be written as

$$V_{K,c} = WV_{k,c} + r_1 d_1 (P_{best} - P_{k,c}) \qquad (7)$$

Where d_1 is acceleration coefficient and usually is equal to 2.05 and r_1 is a random value uniformly generated in the range of $[0, 1]$ and w is inertia weight (ICSO).

A new form of the position update equation composing two terms will be used. In the first term,

the average information of current and previous position and in the second, the average of current and previous velocity information will be used(ICSO). So new global best position equation is described below:

$$P_{i+1} = \frac{P_{i+1}+P_i}{2} + \frac{V_{i+1}+V_i}{2} \quad (8)$$

To improve the performance and achieve better convergence in less iteration ICSO is utilized in this work. By adding a new parameter to the position equation as inertia weight that will be chosen randomly, then by making a new form of the velocity equation to improve searching ability in the vicinity of the best cats. By using this parameter, a balance between global and local search ability can be made. A large inertia weight facilitates a global search while a small inertia weigh facilitates a local search. First a large value will be used and it will be reduced gradually to the least value. So the maximum inertia weight happens in the first dimension of the each iteration and it will be updated decreasingly in every dimension, the velocity update equation for each cat to a new form can be changed. Also the proposed fitness calculation strategy, seen from an optimization perspective, favours the exploitation and exploration in the search process.

Algorithm 1 : ICSO for data recovery
Step 1: Initialize the different parameters of the proposed algorithm like number of cats (n), SMP, SRD, neighborhood structure, $\beta, \alpha \; and \; A$ and randomly placed n number of cats in random space search for cloud data.
Step 2: Initialize position and velocity of each cat (cloud data) into the C-dimensional search space.
Step 3: Compute the fitness function of cats and store the best position of cat into memory
Step 4: While(i < maximum iteration)
Step 5: According to the value of Flag, randomly distribute cats into tracing and seeking modes
Step 6: If (Flag==1); Cat in seeking mode
Step 7: For each cloud data privacy, apply seeking mode process
Make j copies of each cat.
Compute shifting bit value for each cat using SRD.
Add or subtract each cat to shifting value
Compute the fitness function for each new position of cats.
Each time, choose the average inertia weight (w) randomly in range of [0.4, 0.9] in order to controlling excessive roaming of cats outside the searching window
Compute new global best position updation using (7)
Obtain current global best position using (8)
Compare the value of fitness function and keep the best position of cat into memory
End for
Step 8: Else, Cat in tracing mode (recover the deleted cloud data)
Step 9: For each cat, apply tracing mode process
Update the velocity of each sensitive cloud data using (3).
Update the position of each sensitive cloud data using (4).
Compute the fitness function for newly generated position of cat
Compare the fitness function value and keep the best position of recovered cloud data in memory
End for
Step 10: Update the position of cloud data and also determine the best position of cloud data
Step 11: End while
Step 12: Obtain the final data recovery files.
The novelty of the ICSO algorithm is increasing the local and global best solutions using average inertia weight values. Hence it is used to improve the privacy protection and data recovery results

3.3 Secured data transmission via EAES algorithm
In this study, the secured data transmission is carried out to verify data integrity and privacy with EAES algorithm. AES rest on the principle of design called replacement network [16]. The blocks and keys can be selected from sets of 128, 160, 192, 224, or 256 bits. AES can only handle keys with a block size of 128 bits and one of three different key lengths: 128, 192, or 256 bits, according to the AES standard. Depending on whatever version is in use, the standard's name is modified to AES-128, AES-192, or AES-256. The data block's two halves are first utilized to alter the other half, and then they are switched. Permutations and replacements are utilized in this illustration to process the entire data block concurrently during each cycle. To improve the key size and shift operations, EAES algorithm is proposed

Keyword Generation
For this task, generating an AES encryption key is a significant process for encryption purposes. The key size for encryption is the same as the size of the entry. Here, the 16-byte encryption key is used in the proposed system for the purpose of encryption. The 16-byte keys are arranged in the format of the 4x4 matrix displayed below.

Key=
$k0\ k4\ k8\ k12\ k1\ k5\ k9\ k13\ k2\ k6\ k10\ k14\ k3\ k7\ k11\ k15$
(9)

Generally speaking, there are several stages to the AES algorithm's process of encrypting the input data file. Additionally, this is an iterative block cypher with changeable key lengths and 128 block sizes. States are intermediate products that are impacted by various transformations. The state resembles a rectangle collection of bytes in essence. The following Table 1 shows that if the block is 16 bytes in size, the rectangular array is 4 by 4.

Table 1 Block size of the AES algorithm

b0	b1	b2	b3	b4	b5	b6	… … … … …	b15

The arrangement of the input files is in 4 x 4 as showed below. The first four-byte of the input is
$b0\ b4\ b8\ b12\ b1\ b5\ b9\ b13\ b2\ b6\ b10\ b14\ b3\ b7\ b11\ b15$
(10)

The state only performs small sets of operations in rounds where tasks include:

- Sub-bytes
- Shift row
- Mix column
- Add round key

Sub-bytes operation

Subbyte operations are non-linear byte substitutions that work independently for every byte of report. The S-Box is reversible and is built using two transformations.

Round-based Shift operation

The following actions are carried out during this typical shift row operation,

- 1st row to 0 positions to the left
- 2nd row to 1 position to the left
- 3rd row to 2 positions to the left
- 4th row to 3 positions to the left

Normally, for 128 bits, the AES algorithm performs the operation in 10 turns. Here, the new system has been ameliorated by altering the shift operation. Each turn, the sub-byte operation is performed first. The next step is a round-based shift operation. The suggested approach determines if the number of turns is odd or even for each turn. The row shift transforms operate on the rows in the state table if the number of rounds is seven. The row shift transform travels to two different locations in the matrix if the number of rounds is even. Operation of the gear shift is depicted in Fig 5.4 and 5.5.

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

b0	b4	b8	b12
b5	b9	b13	b1
b10	b14	b2	b6
b15	b3	b7	b11

Fig 2 Shift Operation for odd

b0	b4	b8	b12
b5	b9	b13	b1
b10	b14	b2	b6
b15	b3	b7	b11

b0	b4	b8	b12
b13	b1	b5	b9
b2	b6	b10	b14
b7	b11	b15	b3

Fig 3 Shift Operation for even

Mix-column operation

The latest mathematical calculations are used by mix column operations.

Add round key

With the addition of the round key, the group key is used for reporting by XOR at the bit level. The circular solution can be attained from the encryption key with a key program

EAES algorithm is applied to encrypt all the incoming data from the cloud data, the proposed work uses the cryptography only for the IDs

The public key is chosen randomly from the two huge prime integers in this key generation procedure, and the private key is computed using the public key. The key generation procedure is slowed down because the randomly chosen public key requires more time to execute. Additionally, it compromises system security while encrypting and decrypting data.

Therefore, the ICSO is used to tune key generation process optimally. Key generation, encryption, and decryption are the three primary phases

a) Key generation

The encryption and decryption processes are done using the public and private keys. Only the private key is employed to quickly decrypt messages that have been encrypted with the public key. Compute the fitness of the individual by using equation (11).

$$Fitness = Max(Throughput) \quad (11)$$

b) Encryption

Next, the encryption takes place after generate the keys for encryption and decryption. It transforms the information into a code that is only known to a select few, hence concealing the information's actual meaning. To encrypt the patient data $\left(\vec{I}_{PD}\right)$ using a public key $\left(\tilde{P}_U\right)$ that has been generated at key generation process to generate the cipher. It is defined as follows:

$$\vec{C}_T = \left(\vec{I}_{PD}\right)^{\tilde{P}_U} \bmod \vec{E} \quad (12)$$

In this equation, $\vec{I}_{PD}$ indicates the input patient data to be transmitted to the cloud and $\vec{C}_T$ denotes the ciphertext for the corresponding input. These encrypted ciphertexts are securely stored in the CS for further processing.

c) Decryption

Decryption is converting data that has been encrypted back into its original form. In most cases, decryption is simply the reverse procedure of encryption. It decodes the encrypted information so that only a user with the appropriate authorization decrypts the data. The decryption process is mathematically expressed as follows:

$$\vec{I}_{PD} = \left(\vec{C}_T\right)^{\tilde{P}_R} \bmod \vec{E} \quad (13)$$

d) Authentication

When the user wants to download their encrypted healthcare data from the CS, authentication first occurs on the receiver side. On this receiver side, two levels of authentication are done to enhance the system's security.

- First level authentication

Initially, if the user wants to read the files from the hospital CS, the first level of authentication is performed. At this level, the user sends a read request along with the ID and password to TC. TC verifies it with the stored cipher value in the cloud to check whether it is an authenticated user. The TC permits the users to read the files from the CS if they are authorized.

- Second level authentication

After successful completion of the first level of authentication, if the user wants both read and download the file from the CS, the user sends a read and download request along with their user's name, user ID, and timestamp to TC. Next, the TC takes hash values for the user providing details and is checked with the stored hash values. If matched, the user can read and download the files from the CS. Finally, the receiver decrypts the downloaded file using equation (12) for early diagnosis

4. SIMULATION RESULT

4.1 Experimental Results and Discussions

Numerically analyses and simulation experiments on the communication, computation and storage costs of our scheme are conducted in this section. Then, give the comprehensive performance comparisons between our ICSO-EAES scheme with the existing schemes PCAD [17], P-PCAD [18], CTAPDA and SPRPLDA all of which realize with the public cloud auditing. For the sake of fairness of comparison, assume that the size of original file F in all schemes is equal, the communication costs of uploading F and its identifier and the storage costs of storing F are not considered for all schemes in the numerical and experimental analyses. The criteria considered for the evaluation of the LDAP system are communication cost, computation cost and storage cost. For calculating the computation cost, the time taken for creating the hash tables and time for verifying the metadata are considered. For communication cost, the time taken to transfer the data between CSP and TPA and also for the TPA to send the result to the DO will be measured. Data storage auditing is extensively a resource demanding service in terms of communication cost, computational resource, and storage space. In this subsection briefly analyse these performance metrics. In this experiments, set the base field size to be 512 bits, the size of an element to be |p| = 160 bits, the size of a data file to be 20MB composed by 1,000,000 blocks.

4.2 Communication Cost Comparison

This section compare ICSO-EAES scheme with existing scheme PCAD, P-PCAD, CTAPDA and SPRPLDA about the communication costs in storage phase and auditing phase, and the results are shown

in Fig. 4. In the PCAD scheme, each subsequent owner does not need to upload the authentication tags, because this scheme audits the integrity of file F by using the authentication tags uploaded by the original owner. However, the original owner needs heavy communication costs in storage phase and it is not able to ensure that each owner can audit the integrity of file F separately. The P-PCAD scheme can ensure that each owner audits the integrity of file F separately, the CSP needs to store all authentication tags from all owners of file F, which results in heavy storage overheads for the CSP. In ICSO-EAES, the communication costs during the auditing phase is minimal, the ICSO-EAES scheme requires a smaller proof with constant size than the PCAD, P-PCAD, CTAPDA and SPRPLDA schemes. From the results it is obvious that, the ICSO-EAES scheme is comprehensively more efficient in communication costs during storage and auditing phases.

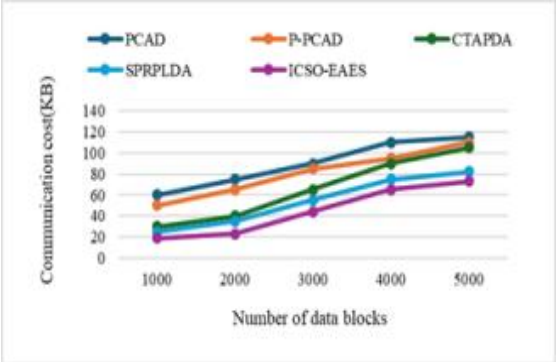

Fig 4 Comparisons of communication costs

4.3 Computation Cost

Now, the ICSO-EAES scheme with the existing schemes such as PCAD, P-PCAD, CTAPDA and SPRPLDA show them in Fig. 5. Without loss generality, the computation complexity only considers the multiplication operation, exponentiation operation and bilinear pairing operation on multiplicative group to describe computation costs simply and effectively. However, the PCAD scheme needs more computation costs in storage phase and generating the proof than the P-PCAD. In many other auditing schemes such as PCAD, P-PCAD and CTAPDA the dynamic auditing is not supported due to computation cost and difficulty in performing audit operation. This proposed ICSO-EAES method will solve these two problems for better performance of the cloud system and to improve users trust. This ICSO-EAES method reduces the computation cost at the user side by simply outsourcing the audit process to a TPA. To sum up, can get a conclusion that the ICSO-EAES scheme is relatively computationally efficient during the storage phase and auditing phase.

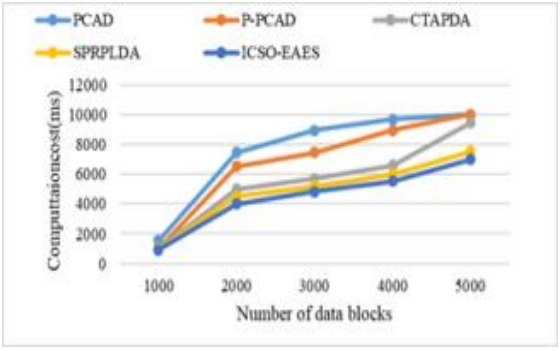

Fig 5 Computation Cost Comparison

4.4 Execution Time

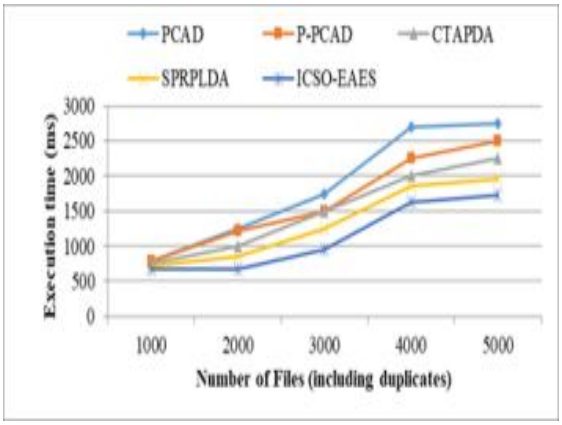

Fig 6 Execution time

Fig.6 illustrates the relationship between the execution time on communications and the number of files. It can be said that the proposed ICSO-EAES approach consumes less time when compared with the other PCAD, P-PCAD, CTAPDA and SPRPLDA approaches. ICSO-EAES takes lower execution time value compared with PCAD, P-PCAD, CTPDA, SPRPLDA since it secures against the poison attack and brute force. From the result, it concluded that the proposed ICSO-EAES algorithm provides better performance for privacy protection and data recovery over cloud data.

5. CONCLUSION

In this work, ICSO-EAES algorithm is proposed to solve the problem of user's privacy leakage on cloud storage auditing. In the proposed scheme, the privacy of user can be well preserved against the cloud and other parties. In this work, ICSO and EAES algorithm is proposed for security and privacy preservation on cloud. ICSO algorithm is focused for data backup and recovery. It generates best fitness values for data protection in terms of security and

privacy in cloud. Then, the secured data transmission is done and it ensured data integrity as well as privacy via EAES algorithm. The encryption and decryption processes are done using the public and private keys. Finally, dual authentication is done to enhance the system's security. From the result, it concluded that the proposed ICSO-EAES algorithm provides better performance in terms of computational cost, communication cost and execution time rather than the existing algorithms.

REFERENCES

[1]. Das, Swatisipra, et al. "A secure, privacy-preserving, and cost-efficient decentralized cloud storage framework using blockchain." *Journal of King Saud University-Computer and Information Sciences* 36.10 (2024): 102260.

[2]. Guan, Xuening, Jinyong Chang, and Wei Zhang. "Secure data sharing scheme with privacy-preserving and certificateless integrity auditing in cloud storage." *Computer Communications* 224 (2024): 285-301.

[3]. Anbuchelian, S., C. M. Sowmya, and C. Ramesh. "Efficient and secure auditing scheme for privacy preserving data storage in cloud." *Cluster Computing* 22 (2019): 9767-9775.

[4]. Agarkhed, Jayashree, and R. Ashalatha. "An efficient auditing scheme for data storage security in cloud." *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 2017.

[5]. Li, Jiaxing, et al. "Blockchain-based public auditing for big data in cloud storage." *Information Processing & Management* 57.6 (2020): 102382.

[6]. Huang, Yinghui, et al. "Privacy-preserving certificateless public auditing supporting different auditing frequencies." *Computers & Security* 128 (2023): 103181.

[7]. Shu, Jiangang, et al. "Blockchain-based decentralized public auditing for cloud storage." *IEEE Transactions on Cloud Computing* 10.4 (2021): 2366-2380.

[8]. Gan, Qingqing, Xiaoming Wang, and Xuefeng Fang. "Efficient and secure auditing scheme for outsourced big data with dynamicity in cloud." *Science China Information Sciences* 61.12 (2018): 122104.

[9]. Hou, Huiying, et al. "Enabling secure auditing and deduplicating data without owner-relationship exposure in cloud storage." *Cluster Computing* 21 (2018): 1849-1863.

[10]. Zhang, Jindan, et al. "Improved secure fuzzy auditing protocol for cloud data storage." *Soft Computing* 23 (2019): 3411-3422.

[11]. Pawar, Ankush Balaram, Shashikant U. Ghumbre, and Rashmi M. Jogdand. "Privacy preserving model-based authentication and data security in cloud computing." *International Journal of Pervasive Computing and Communications* 19.2 (2023): 173-190.

[12]. Shen, Jian, et al. "A privacy-preserving and untraceable group data sharing scheme in cloud computing." *IEEE Transactions on Dependable and Secure Computing* 19.4 (2021): 2198-2210.

[13]. Bingu, Rajesh, and S. Jothilakshmi. "Cloud auditing and authentication scheme for establishing privacy preservation." *Multimedia Tools and Applications* 83.15 (2024): 43849-43870.

[14]. Sharma, Deepak Kumar, Arushi Garg, and Aparna Jha. "Assorted Cat Swarm optimisation for Efficient Resource Allocation in Cloud Computing." *2018 Fourteenth International Conference on Information Processing (ICINPRO)*. IEEE, 2018.

[15]. Zhang, Haiyu, and Runliang Jia. "Application of chaotic cat swarm optimization in cloud computing multi objective task scheduling." *IEEE Access* 11 (2023): 95443-95454.

[16]. Abikoye, Oluwakemi Christiana, et al. "Modified advanced encryption standard algorithm for information security." *Symmetry* 11.12 (2019): 1484.

[17]. J. Yuan and S. Yu, ''Secure and constant cost public cloud storage auditing with deduplication,'' in Proc. IEEE CNS, National Harbor, MD, USA, Oct. 2013, pp. 145–153,

[18]. C. Li and Z. Liu, ''A secure privacy-preserving cloud auditing scheme with data deduplication,'' Int. J. Netw. Secur., vol. 21, no. 2, pp. 199–210, Mar. 2019