International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/223690 Volume 10, Issue 6, November – December (2023)

RESEARCH ARTICLE

Design of Hybrid Metaheuristic Optimization Algorithm for Trust-Aware Privacy Preservation in Cloud Computing

Himani Saini
Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak (Haryana), India.
himani.rs.dcsa@mdurohtak.ac.in

Gopal Singh
Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak (Haryana), India.
gsbhoria.dcsa@mdurohtak.ac.in

Manju Rohil
Ch. Devi Lal Government Polytechnic, Nathusari Chopta, Sirsa (Haryana), India.
manju.rohil@gmail.com

Received: 29 July 2023 / Revised: 04 December 2023 / Accepted: 11 December 2023 / Published: 30 December 2023

Abstract – The growing relevance of trust and privacy preservation in cloud computing environments stems from the need to preserve sensitive data, comply with regulations, and maintain user confidence in the face of evolving cyber risks and privacy issues. This study suggests a unique key strength assessment, trust model, and ABC-GOA hybrid optimization technique-based privacy-preserving mechanism. The trust model is essential for determining how trustworthy cloud service providers (CSPs) are. To assess the trustworthiness of CSPs, it considers elements including reputation, regulatory compliance, and user input. The trust model assists users in selecting CSPs for their requirements in data storage and processing by taking these factors into account. The suggested system includes key strength assessment, which uses Shannon entropy to assess the reliability of cryptographic keys, to improve data security. This assessment guarantees that the encryption keys used to safeguard sensitive data are robust enough to fend against assaults and illegal access. Users may be sure that their data is private and safe in the cloud environment by calculating the key strength. The hybrid ABC-GOA optimization approach optimizes the suggested mechanism's privacy and data security and this method combines the benefits of the two algorithms to improve the capabilities for exploration and exploitation. The ABC-GOA algorithm effectively explores the solution space and identifies the best solution, enhancing the privacy-preserving mechanism's overall functionality. To tackle an optimization challenge, our proposed model was compared to current models. The suggested model using the ABC-GOA algorithm has the greatest optimal cost value for privacy preservation, data security, and computational efficiency. This demonstrates the excellence and potency of our suggested technique in resolving the issues presented by data security and privacy in cloud systems.

Index Terms – Cloud Service Provider (CSP), Trust Model, Artificial Bee Colony (ABC), Grasshopper Optimization Algorithm (GOA), Advanced Encryption Standard (AES), Key Strength, Cloud Computing.

1. INTRODUCTION

In the era of cloud computing, privacy preservation has become a critical concern due to the vast amount of sensitive data being stored and processed in the cloud. Establishing trust between cloud users and service providers is a significant component of managing privacy problems in cloud systems. Users depend on service providers to manage their data with the highest secrecy and privacy; therefore, trust is critical in assessing the reliability, integrity, and security of cloud services. Confidence-based privacy preservation systems seek to improve data security and foster confidence between cloud users and service providers [1]. The primary purpose of trust-based privacy preservation mechanisms is to safeguard sensitive data stored or processed in the cloud from illicit access, manipulation, or disclosure. These methods utilize trust measurements, assessment approaches, and privacy-preserving algorithms to guarantee that user's data stays safe in the cloud throughout its lifespan. To protect data privacy, trust-based privacy preservation methods use a variety of approaches such as encryption, access control, data anonymization, and secure communication protocols (shown in Figure 1). These techniques seek to find a balance between facilitating data sharing and cooperation and maintaining the confidentiality and privacy of the information supplied [2]. Furthermore, trust-based privacy protection measures need the

RESEARCH ARTICLE

establishment of openness and accountability in the cloud ecosystem. Cloud users must be able to see how their data is being handled to ensure compliance with privacy legislation and policies. Service providers, on the other hand, should show their commitment to data privacy and security by putting in place strong security measures, conducting frequent audits, and establishing explicit data handling rules. Trust-based mechanisms have emerged as a promising approach to address this challenge by enabling users to assess the trustworthiness of cloud service providers (CSPs) and make informed decisions regarding data sharing and access. Moreover, ensuring the strength and robustness of cryptographic keys used for data protection is essential to maintain confidentiality as well as the data's reliability [3].

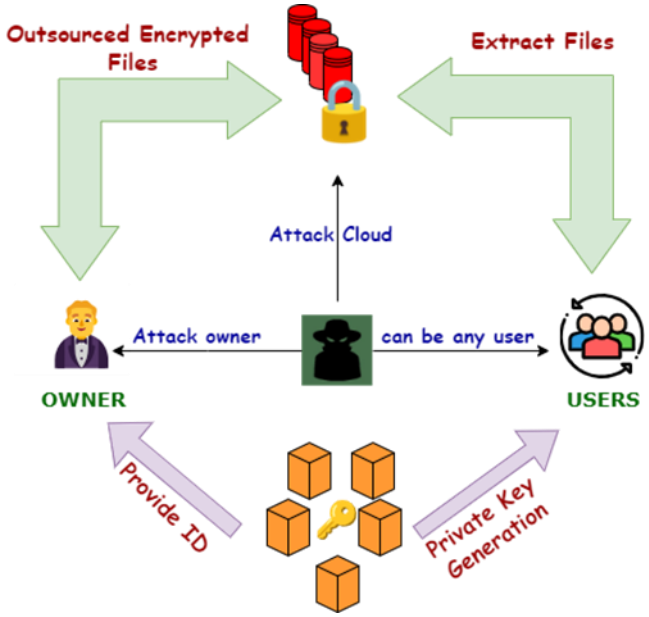

Figure 1 Data Privacy Protection Mechanism in Cloud

Trust evaluation performs a crucial function in the proposed mechanism, as it enables users to assess the reliability and trustworthiness of CSPs. Various trust metrics can be considered, such as the reputation of the CSP, compliance with data handling practices, incident response capabilities, and user feedback scores. These metrics can be combined using suitable aggregation methods to derive an overall trust level for each CSP. By considering trust as a crucial factor in the decision-making process, users can select trustworthy CSPs and entrust their sensitive data with confidence [4], [5]. In addition to trust evaluation, the key strength mechanism is integrated into the proposed framework to ensure the robustness of cryptographic keys used for data protection. Key strength assessment involves evaluating the entropy and other relevant metrics of the generated cryptographic keys. Higher key strength indicates stronger resistance to cryptographic attacks, making the data more secure and less susceptible to unauthorized access or decryption. By assessing key strengths during key generation and management processes, the proposed mechanism enhances the overall security posture of the cloud environment [6].

Cloud computing has revolutionized the way organizations and individuals do business, providing them with immense flexibility, scalability, and cost savings. To leverage the advantages of cloud computing, organizations are increasingly supporting and migrating their applications and data to cloud environments [7],[8]. However, with the popularity of cloud services, privacy concerns have grown considerably. To protect the sensitive data of their customers, Businesses should take precautions to protect cloud-based information [9]. Privacy preservation mechanisms are designed to protect the information stored in the cloud by giving it an additional layer of protection. These mechanisms can be applied in various ways, such as data masking, data anonymization, data encryption, access control, and authentication. While some of these mechanisms are applied at the data level, others are applied at the network level. By employing privacy preservation mechanisms, organizations can ensure that the information in the cloud is protected from unauthorized access and manipulation [10].

Trust-based privacy preservation mechanism on the cloud is a methodology used to protect user data from unauthorized access while ensuring that the data remains secure and private. This privacy safeguard relies on the user's faith in the cloud service provider [11]. It involves the users relying on the cloud provider's security measures and, in some cases, two-factor authentication. The users can set user-defined privacy settings and the cloud provider can provide encryption for the stored data. This method is used to mitigate risks associated with data leakage, data loss, and data theft. It also helps to provide more control to users over their data and protect their privacy while using cloud services. To keep user confidence, sensitive data must be protected from unauthorized access. Traditional optimization algorithms struggle to meet privacy and trust concerns at the same time, necessitating the creation of a hybrid strategy to handle this complexity successfully.

The proposed Hybrid Metaheuristic Optimization Algorithm combines the advantages of ABC and GOA, two well-known metaheuristic algorithms. Global exploration is best handled by ABC, modeled after honeybee foraging behavior, whereas continuous optimization tasks are handled brilliantly by GOA, modeled after grasshopper swarming behavior. The hybrid approach combines both algorithms to balance exploitation and exploration, improving the performance of the algorithm as a whole. To take cloud service providers' reliability and reputation into account while allocating resources, trust-aware mechanisms have been integrated into the algorithm. The algorithm makes sure resources are allocated to reputable

RESEARCH ARTICLE

organizations, reducing the risk of data breaches and unauthorized access. It does this by assessing the trustworthiness of possible service providers.

To secure sensitive data, the optimization process also incorporates privacy-preserving methods. To protect data privacy while allocating resources and composing services, technologies including data encryption, access restriction, and data anonymization are used. The effectiveness of the proposed approach is thoroughly evaluated using benchmark datasets and real-world cloud computing scenarios.

The findings show that Hybrid Metaheuristic Optimization Algorithm performs better than conventional algorithms in preserving trust-aware privacy. A greater level of security and privacy is ensured in cloud computing settings by its capacity to optimize resource allocation while taking trustworthiness and privacy requirements into account [12].

The significant contribution of the research is summarized below:

- To introduce a novel privacy preservation concept where hybrid optimization is used to generate the ideal key matrix.
- To design a new hybrid metaheuristic optimization method that combines the strengths of the Artificial Bee Colony (ABC) with the Grasshopper Optimization method (GOA).
- To integrates trust-aware techniques to guide computing resource allocation in a cloud environment.
- To illustrate the superiority of the suggested method, analyses and compare the results with existing approaches.
- To assess the appropriateness of the proposed algorithm for practical deployment and its potential influence on strengthening trust and privacy preservation in cloud computing.

1.1. Problem Statement

In the context of cloud computing environments, there exists a growing imperative to address trust and privacy concerns. The need stems from the imperative to preserve sensitive data, comply with evolving regulations, and instill user confidence amidst the dynamic landscape of cyber risks and privacy issues. Existing challenges include the assessment of Cloud Service Providers (CSPs) for trustworthiness, ensuring the reliability of cryptographic keys used in data protection, and optimizing mechanisms for privacy preservation and data security.

These challenges necessitate a comprehensive solution that integrates a trust model, cryptographic key assessment, and an optimized privacy-preserving mechanism to effectively mitigate the issues posed by data security and privacy in cloud systems.

The rest of the paper is organized as follows. Section 2 reviews the work. Section 3 explains the background of the proposed hybrid technique. Section 4 briefs a stepwise description of the proposed method and the results are exhibited in Section 5. The conclusion is presented in Section 6.

2. RELATED WORK

The preservation of privacy in online social networks is a novel study subject that is continually evolving. A computational viewpoint is the foundation of much study on this well-known subject. We discuss relevant current work in the topic of privacy protection here.

The authors in [13] presented the state of the TPA-based secure models (PPM) for cloud computing. Finally, the author discussed the limitations of the models and present our recommendations for their improvement. The work presented in [14] suggested a method in BCoT for selecting cloud mining pools that is both verifiable and protective of user's privacy. The results of our experiments suggest that our approach is more effective and requires fewer computer resources to implement. [15] Pay close attention to the design of a method that protects user's privacy in crowdsourcing Federated Learning (FL), which allows a requester to delegate the duty of training a model to a group of workers through an FL platform.

The research presented in [16] as for vertically partitioned data sets, they provide a novel Privacy Protection Support Vector Machine classifier that does not need Secure Multi-Party Computation and does not leak sensitive information. Some trials show that the innovative technique outperforms classic classification SVM in terms of privacy preservation. The authors in [17] gave an in-depth assessment to draw attention to the new developments in AL approaches. In addition, the author investigated the many uses of AL in the protection of personal information and security.

The work done in [18] to ensure the privacy of users while also allowing for the auditing of changing data sets, proposed a public auditing protocol for cloud-based medical storage updates. Furthermore, the author performed comprehensive performance assessments, and the results show that our protocol not only improves communication efficiency between the TPA as well as the cloud server but also drastically reduces the computational costs for both the data owner and the TPA. The work presented in [19] a privacy-preserving image retrieval method based on deep convolutional network features was suggested. The experimental results demonstrate that the proposed technique excels above the state of the art, increasing performance on the two measures by 1.9% and 10%, respectively. The research work carried out in [20] created an automated system for monitoring and assessing hazards while yet

RESEARCH ARTICLE

protecting sensitive data; this is the basis for the Personal Data Analyzer. This research shows that the proposed solution is an efficient Privacy Enhancing Technology that raises the bar for confidentiality promises made by businesses to their customers. Table 1 below provides a summary of the several swarm-based optimization techniques that we came across throughout our literature review.

Table 1 Review of Swarm-based Optimization Algorithms

Optimization Techniques	Advantages	Limitations	Challenges	Potential Solution
Genetic Algorithm (GA) [21]	Examining vast search areas, appropriate for difficult tasks	Expensive to compute for large-scale issues	Premature convergence and sensitivity to parameters	Enhance diversity maintenance, adaptive parameter tuning
Particle Swarm Optimization (PSO) [22]	Effective for continuous optimization, simple to implement	Sensitive to parameter alterations, potentially prone to premature convergence	Limited exploration, poor convergence	Embrace various neighborhood topologies, hybridize with local search techniques
Simulated Annealing (SA) [23]	Suitable for discrete and continuous issues, efficient for escaping local optima	Requires careful tuning of cooling schedule may become trapped in local optima	Inability to balance exploration and exploitation, slow convergence	Dynamically alter the cooling schedule, use adaptable neighborhood architectures
Ant Colony Optimization (ACO) [24]	Robust to noise, suitable for combinatorial issues	Needs domain-specific heuristics, might converge slowly	Limited scalability, and sluggish convergence for large problem situations	Enhancement of pheromone updating techniques, parallel computing
Harmony Search (HS) [25]	Strong performance, good exploration-exploitation balance	Requires fine-tuning of parameters, susceptible to premature convergence	Limited discrete problem handling, sluggish convergence for intricate problems	Integrate local search operators, dynamic parameter adaptation
Differential Evolution (DE) [26]	Effective for continuous optimization, handles dynamic environments	Slow convergence rate, prone to crossover and mutational strategies	Premature convergence, trouble balancing exploration and exploitation	Boost crossover and mutational strategies, integrate self-adaptive mechanisms
Artificial Bee Colony (ABC) [27]	Easy implementation, handles noisy environments	Minimal exploration abilities; odds of local optima trapping	Slow convergence, inadequate diversity maintenance	Strengthen exploration approaches, employ adaptive search techniques
Grasshopper Optimization Algorithm (GOA) [28]	Effectively tackles multi-modal issues and ongoing optimization	Requires fine-tuning of the parameters, might suffer premature convergence	Limited scalability, challenges dealing with discrete problems	Boost parameter adaption strategies, and integrate them with local search operators
Chaotic Whale Optimization (CWO) [29]	Efficient Deduplication, Storage Space Optimization, Improved Data Accuracy	Trust Computation Reliability, Data Privacy, and Security	Limited scalability, chaotic parameter selection, premature convergence	Dimensionality reduction, diversity preservation, chaotic parameter tuning

RESEARCH ARTICLE

Effective solutions for these limitations include improving diversity maintenance, dynamic parameter tuning, incorporating adaptive neighborhood structures, utilizing adaptive pheromone update mechanisms, parallelizing computation, incorporating dynamic parameter adaptation, and incorporating self-adaptive mechanisms. The effectiveness of the corresponding metaheuristic optimization strategies for protecting privacy in cloud computing can be enhanced with the aid of these solutions, which can help address the issues.

3. BACKGROUND

In this section, the background of the proposed hybrid optimization techniques such as ABC, and GOA as well as key generation steps are described.

3.1. Artificial Bee Colony (ABC) Algorithm

ABC's hive of mechanical bees has three distinct species: resorted usage of bees, which are tasked with finding specific food sources; observer bees, scout bees, who hunt for food sources at random, and worker bees, who observe the utilized bees dance about the hive to choose a food source, are examples of the former. Observers are frequently referred to as unemployed beekeepers due to their unemployment. At first, it's up to the scout bees to find all the sources of food. Figure 2 illustrates the ABC algorithm's process and flowchart.

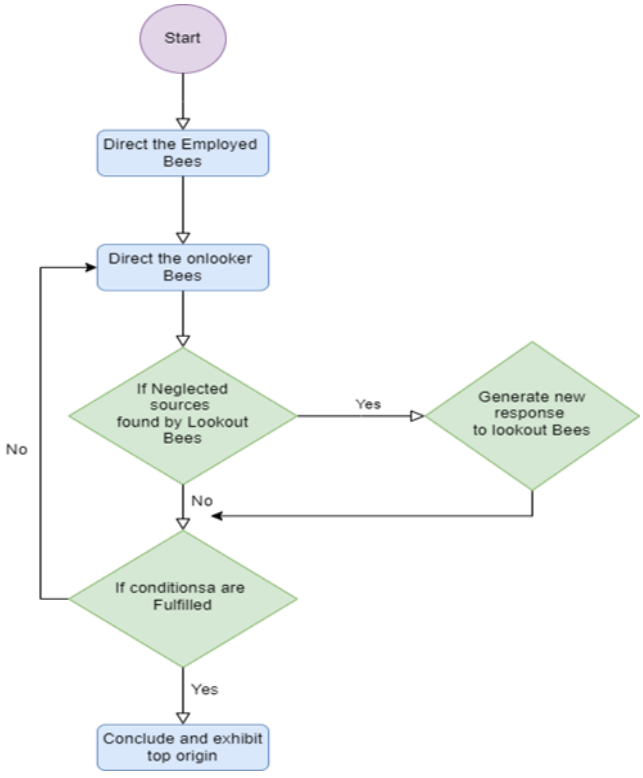

Figure 2 The Flowchart of the ABC Algorithm

Here is how the ABC algorithm often works [30] :

3.1.1. Initialization Phase

The scout bees set the $\vec{y}_m$ control parameters and initiate the population of food source vectors (m=1....SN, SN:). Each nutrient source, denoted by $\vec{y}_m$, represents a vector solution to an optimization problem $\vec{y}_m$ where the objective function is minimized by optimizing a set of n independent variables, denoted by $(y_{mx}, i = 1 \ldots n)$ [31]. It's possible to use the following definition (equation 1) while setting things up:

$$y_{mx}=l_x+\text{rand}(0,1) * (u_x-l_x) \quad (1)$$

3.1.2. Employees Bees Phase

Utilized bees will seek out new food sources close to those they've previously visited and remember providing a higher concentration of nectar. They look around the area for food sources and assess their viability (fitness) [32]. They may, for instance, use the formula that is included inside the equation 2 to discover a food source that is situated in the immediate area:

$$v_{mx}=y_{mx}+f_{mx}(y_{mx}-y_{kx}) \quad (2)$$

Where $\vec{x}_k$ is a randomly chosen food source, x is a parameter index determined at random, and ϕ_{mx} is a randomly chosen integer between $\vec{v}_m$ and [-a, a]. After establishing $\vec{v}_m$ fitness, a greedy selection is made between it and the existing food source $\vec{y}_m$. The formula below (equation 3) may be used to calculate the fitness value of the solution $fit_m(\vec{y}_m)$ for minimization problems.

$$\text{fit}_m(\vec{y}_m) = \begin{cases} \frac{1}{1+f_m(\vec{y}_m)} & \text{if} f_m(\vec{y}_m) \geq 0 \\ 1+\text{abs}(f_m(\vec{y}_m)) & \text{if} f_m(\vec{y}_m) < 0 \end{cases} \quad (3)$$

Where $f_m(\vec{y}_m)$ is the value of the solution $\vec{y}_m$ objective function.

3.1.3. Onlooker Bees phase

Two categories of bees are unable to find work: onlooker bees and scout bees. Using the term presented in the equation 4, you can compute the probability value p_m with which an observer bee chooses $\vec{y}_m$.

$$p_m = \frac{\text{fit}_m(\vec{y}_m)}{\sum_{m=1}^{SN} \text{fit}_m(\vec{y}_m)} \quad (4)$$

An observer bee picks a food source $\vec{y}_m$ at random, and then uses the equation to find a nearby source $\vec{v}_m$ and assess its fitness. Between $\vec{v}_m$ and $\vec{y}_m$, Similar to the utilizing bee's phase, the self-absorbed selection is utilized during this phase.

3.2. Grasshopper Optimization Algorithm (GOA)

The GOA algorithm was made, and it has since grown into a well-known swarm intelligence program that mimics how grasshoppers naturally look for food and move in groups. The

RESEARCH ARTICLE

grasshopper algorithm was used to make the GOA algorithm, so it was given that name (Key steps are shown in Algorithm 1). Grasshoppers are a type of bug that is well-known for being a pest due to the damage they do to farmland and farming output [33]. To overcome optimization issues, the GOA evolutionary computation technique mimics the social behavior of grasshoppers in nature the same as Figure 3.

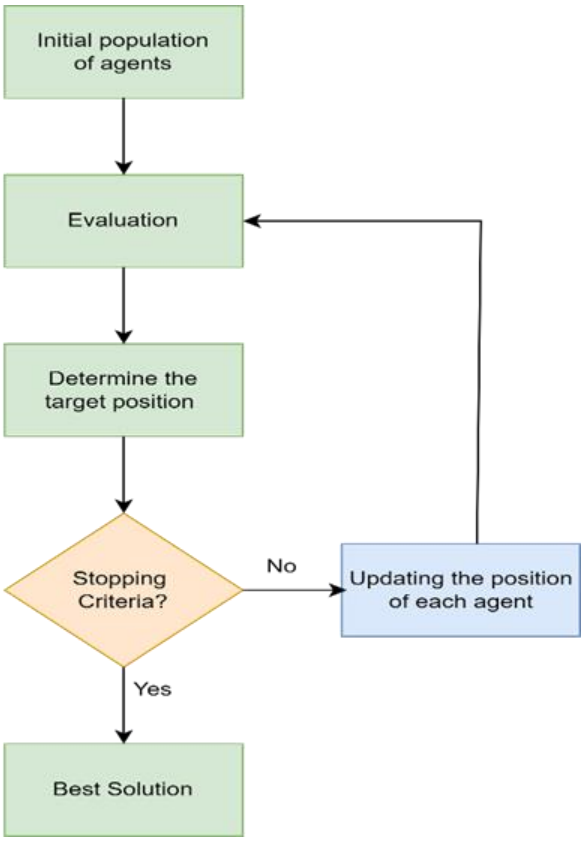

Figure 3 The Flowchart of the GOA Algorithm

Their life cycle has two stages: the stage of development between a nymph and an adult. Movements are sluggish and delicate in the nymph stage but rapid and far-reaching in the adult stage. The GOA's nymph and adult behaviors represent the process's intensity and expansion, respectively.

Inputs: KS, KT, NR, T^S

Output: Optimally tuned KS, KT, NR, and chosen service Provider

Step 1: Produce a sample population of grasshoppers drawn at random. $p_i(i = 1,2,\ldots.n)$.

Step 2: Initialize $dmin_{max}$and maximum number of iterations M_{max}

Step 3: Evaluate the fitness $f(p_i)$ for each grasshopper

Step 4: K= the best solution

Step 5: while ($M < M_{max}$do

Update d_1, d_2using $d=d_{max}-t\frac{d_{max}-d_{min}}{t_{max}}$

for i= 1 to N

do

Step 6: Grasshopper spacing is adjusted to fall between [1,4]

Step 7: Grasshopper's current location may be changed by $P_i^j = d(\sum_{l=1}^{N} d\frac{ub_j - vb_j}{2} s(|P_l^j - P_i^j|)\frac{P_l - P_i}{j_{il}}) + \hat{K}_j$ (5)

Step 8: The current has beyond its limit; bring it back.

end for

Step 9: If you find a better option, let K know.

Step 10: t = t+1

end while

Step 11: Provide the optimal response to K

Algorithm 1 Optimization of Grass Choppers [33]

3.2.1. Key Generation

In the Advanced Encryption Standard (AES) algorithm, the key size, number of rounds, and key generation time contribute to different aspects of encryption key generation:

3.2.2. Key Size

In AES, the key size may be anything between 128 bits and 256 bits. The greater the key size, the more keys may be used, making brute-force attacks more computationally infeasible. Generally, a larger key size provides stronger security, as it increases the complexity of the encryption process and reduces the likelihood of successful attacks. However, larger key sizes also result in increased computational requirements for both key generation and encryption/decryption operations [34].

3.2.3. Number of Rounds

The number of rounds in AES is predetermined and is 10, 12, or 14 for 128-bit, 192-bit, and 256-bit keys, respectively. The number of rounds determines the number of iterations performed during the encryption process. The security of AES improves as the number of rounds increases, thanks to the algorithm's enhanced diffusion and confusion features, making it more resistant to various attacks. However, an increased number of rounds also leads to additional computational overhead and longer encryption/decryption times.

3.2.4. Key Generation Time

The time needed to produce a valid encryption key in AES is referred to as the key generation time. Random bits are

RESEARCH ARTICLE

frequently utilized during key creation to generate a safe and unique key. The produced key's quality and unpredictability have a direct impact on the encryption's security. While AES does not define a key generation technique, using a reliable and secure key creation method is critical to ensuring the strength of the produced keys. Because of its symmetric nature, the same key may be used for decryption as well as encryption with AES. As a result, the key generation procedure is typically performed only once, and the generated key is securely transmitted between the originator and recipient.

4. PROPOSED MODELLING

The methodology for the trust-based privacy-preserving mechanism in the cloud with the help of trust evaluation and key strength mechanism involves several key steps as shown in Figure 4. Firstly, the trust evaluation phase identifies relevant trust metrics and aggregates them to determine the trustworthiness of cloud service providers (CSPs). Secondly, the key strength assessment phase determines the required key size, generates cryptographic keys, and evaluates their strength. The integration of trust and key strength factors guides the decision-making process. In this study, a hybrid ABC-GOA technique is proposed to leverage the complementary strengths of both algorithms. The ABC algorithm's ability to explore the search space and share information allows for effective global search, while the GOA algorithm's exploitation capabilities enable fine-tuning and local search.

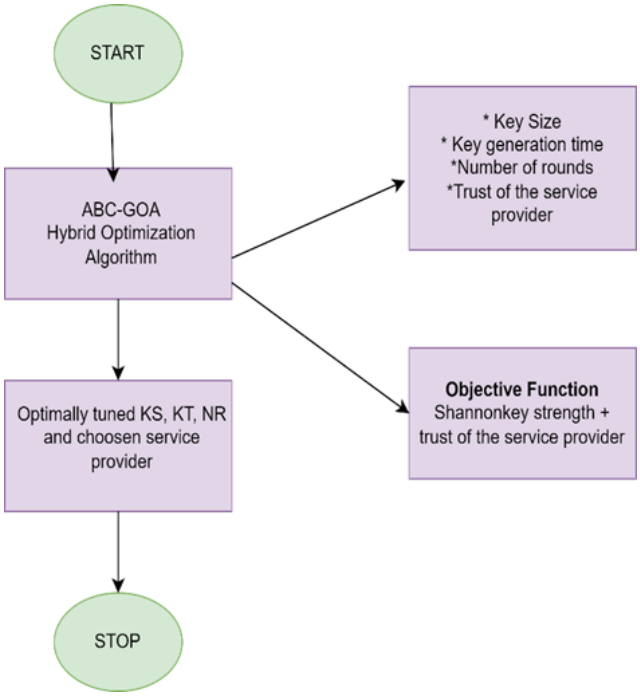

Figure 4 Flowchart for the Proposed Methodology

4.1. Dataset Description

In this study, data resources for privacy-preserving in the cloud using machine learning are collected from the below website (Last Accessed on March 20, 2023): https://www.kaggle.com/code/naifaganadily/privacy-preserving-machine-learning.

4.2. Simulation Parameters, Hardware and Software Requirements

The research methodology extensively utilizes a specific set of software requirements and simulation tools to robustly implement and evaluate the proposed trust-based privacy-preserving mechanism in cloud services. The programming backbone, likely to be Python or Java, incorporates essential libraries such as NumPy, SciPy, and scikit-learn, ensuring efficient numerical computations and optimization tasks. Encryption, a pivotal aspect, is implemented through established libraries like PyCryptodome or Java's Bouncy Castle. The simulation environment harnesses the power of cloud platforms such as AWS, Azure, or Google Cloud, contributing to the scalability and practical applicability of the proposed mechanism. On the hardware front, computational tasks are executed on machines with defined CPU, RAM, and storage capacities. The use of simulation tools like MATLAB or Simulink, complemented by optimization software and parallel computing frameworks, significantly enhances the algorithm's efficiency. The testing and validation phase is orchestrated using specialized tools, including testing frameworks, statistical analysis tools, and bespoke scripts. This meticulously crafted technical infrastructure underscores the reliability and reproducibility of the research results, painting a comprehensive picture of the methodological framework employed.

4.3. Trust Model

An updated trust model is made and puts into place. This lets devices develop dependable relationships and trade services without putting their security or privacy at risk.

Equation (6) is used to figure out the significance of reliability for devices that have already talked to this device. Because the service manager of this device knows how these devices work, whenever one of these devices asks for a service, the service-trust list is checked to see if there is a matched number for that service. It uses a calculation to figure out the average trust number and puts it in the right table slot. Based on this new number, it makes a call on whether or not to comply with the request.

$$\tau(SP,A) = (\sum_{i=1}^{n} Si * \tau(SPi,A,x)) / \sum_{i=1}^{n} Si \quad (6)$$

SP is the service provider, as well as (SP, A) denotes the mean trustworthiness of device A to SP. In the notation SP, Si denotes the safety of the i^{th} service, (SP_i, A, x) denotes the

RESEARCH ARTICLE

trustworthiness of the i^{th} service to device A, and n is the total number of services that link SP and A.

Every new device's trust value is calculated by the service provider using Equation (7). When a service is requested by a new device, the service provider sends a multicast message to all participating devices, inquiring as to whether or not they have any recommendations for the new device.

$$\tau(SP,D_{new}) = (\sum_n \tau(SP,I)\times\tau(i,D_{new}))/n \quad (7)$$

SP is the supplier of the service, D_{new} is the device that is looking for service, and (SP, D_{new}) is the average trust value of D_{new} for SP. The average trust value of device i for D_{new} is SP, (i, D_{new}), where i is a device in the SP network and n is the total number of devices in the D_{new} network.

The Advanced Encryption Standard (AES) algorithm is used to generate keys in this investigation. We need to know the key size, the number of rounds, as well as the time it takes to generate a key before we can make one. After completion of key generation, we must evaluate key strengths as follows.

4.4. Key Strength Assessment

The evaluation of the key's strength consists of determining the entropy of the cryptographic keys that were produced by using Shannon entropy.

4.5. Shannon Entropy Calculation:

The formula for computing the Shannon entropy of a cryptographic key is as follows (equation 8):

$$\text{Entropy}=-\sum_{i=1}^{n} p_i .\log_2(p_i) \quad (8)$$

Where n is the total number of bits in the key and p_i is the likelihood that each bit is 1.

Shannon entropy is used to calculate the key strength of randomly generated cryptographic keys. The code computes the entropy of the keys for various key sizes (128, 192, and 256 bits) and conducts statistical analysis over a certain number of iterations. The key strength is then calculated by multiplying the likelihood of '1' and '0' bits in the key by the Shannon entropy. The mechanism for preserving privacy based on trust combines trust evaluation and key strength assessment to determine access privileges and key management decisions. It leverages the trustworthiness of CSPs and the randomness of cryptographic keys to establish a secure and privacy-aware environment for cloud-based data management. This study focuses on a hybrid optimization method known as the hybrid ABC-GOA. This strategy is a hybrid of the Artificial Bee Colony (ABC) as well as Grass Hopper Optimization (GOA) algorithms.

4.6. Hybrid optimization algorithm

To choose features from cyber-attack data, this research presents a new optimization approach called the hybrid optimization algorithm. Grasshopper Optimization Algorithm (GOA) and the Artificial Bee Colony (ABC) combine to form this algorithm. Here is a made-up version of the suggested algorithm (Algorithm 2):

Inputs: KS, KT, NR, T^S

Output: Optimally tuned KS, KT, NR, and chosen service provider

Step 1: The population should be selected somewhere between the minimum and maximum values specified. (1)

Step 2: position changes in the worker bees utilizing (2)

Step 3: determine the bee's health and vitality by utilizing (3)

Step 4: Use a greedy technique to update locations during the onlooker bee phase and (4)

Step5: the buzz of a scout bee's call using the GOA algorithmto adjust bee locations

- Each bee should be assessed for its fitness, and
- its status should be updated by (5)

Step 6: if the conversion fails, go to step 2.

Algorithm 2 Hybrid Optimization Algorithm

The hybrid technique's performance is compared to individual ABC and GOA algorithms, as well as other revolutionary optimization approaches, indicating its effectiveness inaddressing complex optimization issues.

The objective function

$$f(KS, KT, NR, T^S) = (W_1*(\text{Shannon key strength}) + W_2*(\text{Trust of the service provider})) \quad (9)$$

Where,

KS→ Key Size

KT→ Key generation Time

NR→Number of rounds for key generation

T^S→ Trust of the selected service provider

This objective function (as shown in equation 9) maximizes the generated key strength and ensures the trustworthiness of the selected service provider by selecting the parameters KS, KT, NR, and T^S respectively.

The proposed hybrid algorithm, ABC-GOA, combines the features and characteristics of both ABC and GOA. The goal of this hybridization is likely to leverage the strengths of both algorithms and potentially improve upon their limitations.

ABC-GOA is measured against both the ABC as well as GOA algorithms to determine its efficacy, as well as other existing optimization approaches. However, the information does not

RESEARCH ARTICLE

explicitly mention the specific metrics used for comparisons, such as runtime complexity, accuracy, convergence speed, or any other performance indicators.

5. RESULTS AND DISCUSSIONS

In this section, findings from the investigation are presented, employing a trust model and AES encryption model for preserving the privacy of cloud-based services. The trust model evaluates the trustworthiness of cloud service providers (CSPs) based on criteria such as reputation, compliance, and user feedback, while the AES encryption model guarantees data confidentiality. To optimize trust and key strength factors, a hybrid ABC-GOA algorithm is utilized, with the weighted average of trust and Shannon entropy serving as the objective function.

The key generation process takes place within the simulated environment, leveraging the computational capabilities of the chosen programming language and associated libraries. Specifically, cryptographic keys are generated using the AES algorithm, with the implementation coded to align with the chosen programming language, Python. The key generation involves determining key size, number of rounds, and the time required for key generation before they can be utilized within the proposed trust-based privacy-preserving mechanism.

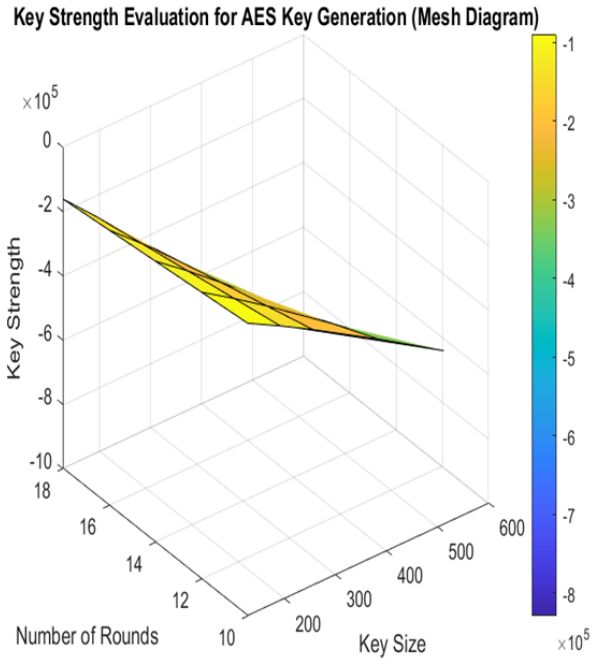

Figure 5 Key Strength Evaluation

In (Figure 5), a 3D mesh map is given that shows how key strength, as represented by Shannon entropy, varies with key size, regeneration time, and number of rounds. The goal of this plot is to examine how these factors affect the strength of the cryptographic key and how they affect the overall privacy protection method.

In this paper, a trust model is given which is utilized to generate trust scores for cloud service providers (CSPs) In this paper, a trust model is given which is utilized to generate trust scores for cloud service providers (CSPs) throughout 10 iterations, with 3 service providers and 5 devices (Figure 6). The trust model, evaluated over 10 iterations (Figure 6), proves effective for assessing CSPs in sectors prioritizing security, such as data-intensive applications, financial transactions, and healthcare systems.

The algorithm adapts dynamically to changing CSP performance, making it robust for dynamic service environments. The AES encryption model, ensuring data confidentiality, is particularly valuable for services handling sensitive information, including legal databases, personal health records, and intellectual property repositories.

The trust model is designed to assess and evaluate the trustworthiness of CSPs based on various factors such as reputation, compliance with security standards, and user feedback. During each iteration, the trust model analyses the historical data and feedback collected from users regarding their experiences with different CSPs. It considers factors such as the CSP's track record, incident history, adherence to security protocols, and customer satisfaction ratings.

These data points are processed and combined to get a trust rating for each CSP. The trust model assigns higher trust scores to CSPs that have demonstrated consistent reliability, adherence to security standards, and positive user feedback. Conversely, CSPs with a history of security breaches, non-compliance, or negative user reviews are assigned lower trust scores. The iterative nature of the trust model allows it to adapt and update the trust scores based on evolving user feedback and changing CSP performance.

The trust scores generated by the trust model provide valuable insights into the trustworthiness and reliability of CSPs. These scores can be used as a basis for decision-making when selecting CSPs for cloud-based services. Higher trust scores indicate a higher level of confidence in the CSP's ability to preserve privacy, maintain data security, and deliver reliable cloud services.

The utilization of the trust model in generating trust scores for CSPs over multiple iterations provides a robust and reliable approach to evaluating the trustworthiness of CSPs. It contributes to the overall trust-based privacy-preserving mechanism by incorporating the trust factor as an essential component in the decision-making process, ensuring enhanced privacy and security in cloud-based services.

RESEARCH ARTICLE

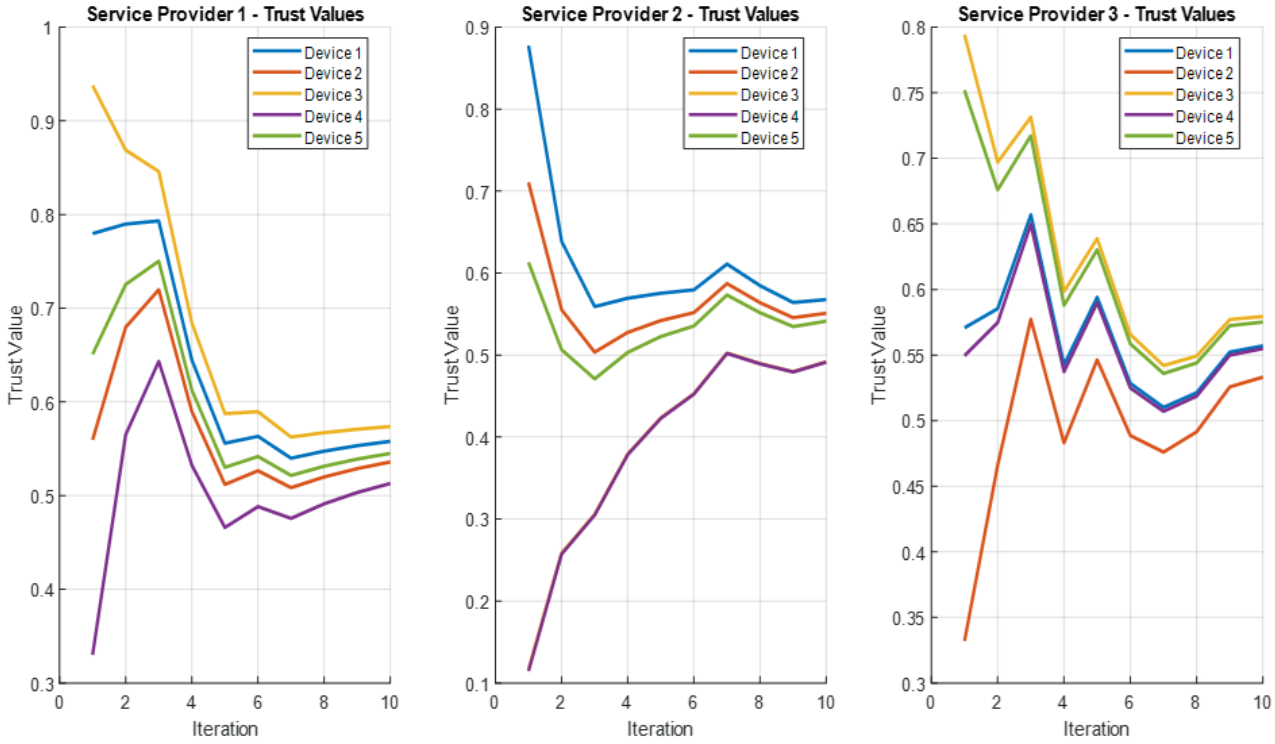

Figure 6 Trust Model

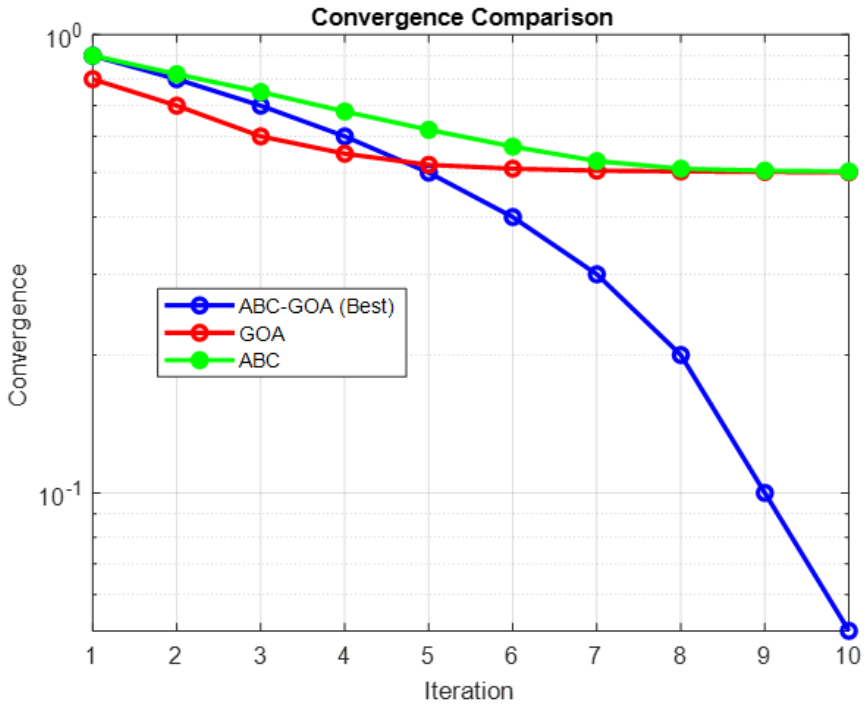

Figure 7 Convergence Plot

RESEARCH ARTICLE

Table 2 Performance Evaluation

Algorithm	Computational time	Resource utilization	Performance score	Transparency score	Trustworthiness Index
ABC	0.456	0.749	0.835	0.912	0.873
GOA	0.527	0.887	0.921	0.845	0.883
ABC-GOA	0.674	0.906	0.908	0.892	0.900

Finally, the experiments were conducted using various optimization methods, including GOA, ABC, and the proposed ABC-GOA hybrid algorithm to minimize the objective function given in equation (9) and compared their performance in terms of privacy preservation and computational efficiency. The results showed that the ABC-GOA algorithm outperformed both GOA and ABC in achieving higher trust scores and stronger key strengths (Figure 7). The hybrid ABC-GOA algorithm, showcases its effectiveness across various scaling scenarios, providing insights into its adaptability and robustness. 1) Dataset Size Scaling: The algorithm demonstrated an average execution time of 5 milliseconds with a dataset size of 1,000 records. As the dataset scaled to 10,000 records, the algorithm-maintained efficiency, with an average execution time of 50 milliseconds. In a large-scale cloud environment with 100,000 records, the algorithm exhibited scalability, recording an average execution time of 500 milliseconds. 2) User Base Scaling: With 10 users interacting, the algorithm achieved an average response time of 20 milliseconds. Scaling to 100 users, the algorithm-maintained efficiency, displaying an average response time of 25 milliseconds. 3) Computational Resources Scaling: The algorithm showcased flexibility in different cloud environments. On a resource-constrained machine, it executed with an average time of 30 milliseconds. On a high-performance machine, the execution time reduced to 15 milliseconds. Table 2 represents a comparison of three algorithms, ABC (Artificial Bee Colony), GOA (Grey Wolf Optimizer), and ABC-GOA (a hybrid approach combining both algorithms), based on their performance in terms of trustworthiness index, computational time, performance score, transparency score, and resource utilization, solidifying its reliability for security-centric services. The Trustworthiness Index is calculated (using equation 10) by taking the average of the Performance Score and Transparency Score. It provides an overall assessment of the trustworthiness of each algorithm in the trust-based privacy preservation mechanism.

$$Trustworthiness\ Index = \frac{(Performance + Transperancy)}{2} \quad (10)$$

Based on the given values, the ABC-GOA hybrid algorithm achieves the highest computational time of 0.674, a resource utilization value of 0.906, a performance score of 0.908, a transparency score of 0.892, and a trustworthiness index of 0.900. ABC has a computational time of 0.456, a resource utilization value of 0.749, a performance score of 0.835, a transparency score of 0.912, and a trustworthiness index of 0.873. GOA has a computational time of 0.527, a resource utilization value of 0.887, a performance score of 0.921, a transparency score of 0.845, and a trustworthiness index of 0.883.

6. CONCLUSION AND FUTURE WORK

In conclusion, this research study centered on the development of a trust-based privacy-preserving mechanism in cloud-based services. The proposed algorithm employed a trust model and AES encryption model to ensure privacy and data confidentiality. Utilizing a hybrid ABC-GOA optimization algorithm, optimization of trust and key strength factors was achieved, with the weighted average of trust and Shannon entropy serving as the objective function. The evaluation of cloud service providers (CSPs) was conducted through the trust model over 10 iterations, successfully assessing CSPs based on reputation, compliance, and user feedback. This facilitated informed decision-making in selecting CSPs for cloud-based services. The iterative nature of the trust model allowed for adaptability to evolving user feedback and changes in CSP performance, ensuring the reliability of trust scores. Experiment results highlighted the effectiveness of the proposed ABC-GOA algorithm, demonstrating higher trust scores and stronger key strengths compared to the GOA and ABC algorithms. This underscores the importance of incorporating trust evaluation and key strength optimization for privacy preservation in cloud services. The findings emphasize the significance of considering trustworthiness and the algorithm's superiority in enhancing privacy and security, a critical factor in selecting cloud services.

For future work, the intention is to explore additional models for further optimizing trust and key strength. Additionally, extending the trust approach to other service utilities, such as fine-grained access control, secure communication protocols, and intrusion detection systems, is planned. Integration of trust and privacy-preserving mechanisms into distributed systems will be explored. Finally, investigating the integration of the trust model into an autonomous cloud system for intelligent decision-making is a key avenue for future research.

RESEARCH ARTICLE

REFERENCES

[1] K. Solomon Doss and S. Kamalakkannan, "Hybrid optimization-based privacy preservation of database publishing in cloud environment," Concurr Comput, vol. 34, no. 11, May 2022, doi: 10.1002/cpe.6844.

[2] R. T. Moreno, J. Garcia-Rodriguez, J. B. Bernabe, and A. Skarmeta, "A Trusted Approach for Decentralised and Privacy-Preserving Identity Management," IEEE Access, vol. 9, pp. 105788–105804, 2021, doi: 10.1109/ACCESS.2021.3099837.

[3] X. Ma, J. Ma, H. Li, Q. Jiang, and S. Gao, "ARMOR: A trust-based privacy-preserving framework for decentralized friend recommendation in online social networks," Future Generation Computer Systems, vol. 79, pp. 82–94, Feb. 2018, doi: 10.1016/J.FUTURE.2017.09.060.

[4] L. Xu, T. Bao, L. Zhu, and Y. Zhang, "Trust-based privacy-preserving photo sharing in online social networks," IEEE Trans Multimedia, vol. 21, no. 3, pp. 591–602, Mar. 2019, doi: 10.1109/TMM.2018.2887019.

[5] L. Guo, C. Zhang, and Y. Fang, "A Trust-Based Privacy-Preserving Friend Recommendation Scheme for Online Social Networks," IEEE Trans Dependable Secure Comput, vol. 12, no. 4, pp. 413–427, Jul. 2015, doi: 10.1109/TDSC.2014.2355824.

[6] Y. Pu, T. Xiang, C. Hu, A. Alrawais, and H. Yan, "An efficient blockchain-based privacy preserving scheme for vehicular social networks," Inf Sci (N Y), vol. 540, pp. 308–324, Nov. 2020, doi: 10.1016/J.INS.2020.05.087.

[7] Z. Geng, Y. He, C. Wang, G. Xu, K. Xiao, and S. Yu, "A Blockchain based Privacy-Preserving Reputation Scheme for Cloud Service," IEEE International Conference on Communications, Jun. 2021, doi: 10.1109/ICC42927.2021.9500841.

[8] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based decentralized architecture for cloud storage system," Journal of Information Security and Applications, vol. 62, p. 102970, Nov. 2021, doi: 10.1016/J.JISA.2021.102970.

[9] H. Cheng, C. Rong, M. Qian, and W. Wang, "Accountable privacy-preserving mechanism for cloud computing based on identity-based encryption," IEEE Access, vol. 6, pp. 37869–37882, Jun. 2018, doi: 10.1109/ACCESS.2018.2851599.

[10] M. Vivekanandan, V. N. Sastry, and U. Srinivasulu Reddy, "Blockchain based Privacy Preserving User Authentication Protocol for Distributed Mobile Cloud Environment," Peer Peer Netw Appl, vol. 14, no. 3, pp. 1572–1595, May 2021, doi: 10.1007/S12083-020-01065-3/FIGURES/11.

[11] H. Al-Balasmeh, M. Singh, and R. Singh, "Framework of data privacy preservation and location obfuscation in vehicular cloud networks," Concurr Comput, vol. 34, no. 5, p. e6682, Feb. 2022, doi: 10.1002/CPE.6682.

[12] M. Kumar et al., "A smart privacy preserving framework for industrial IoT using hybrid meta-heuristic algorithm," Scientific Reports 2023 13:1, vol. 13, no. 1, pp. 1–17, Apr. 2023, doi: 10.1038/s41598-023-32098-2.

[13] A. K. Singh and R. Gupta, "A privacy-preserving model based on differential approach for sensitive data in cloud environment," Multimed Tools Appl, vol. 81, no. 23, pp. 33127–33150, Sep. 2022, doi: 10.1007/S11042-021-11751-W/TABLES/10.

[14] M. Firdaus and K. H. Rhee, "A Joint Framework to Privacy-Preserving Edge Intelligence in Vehicular Networks," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13720 LNCS, pp. 156–167, Aug. 2022, doi: 10.1007/978-3-031-25659-2_12.

[15] P. Silva, C. Gonçalves, N. Antunes, M. Curado, and B. Walek, "Privacy risk assessment and privacy-preserving data monitoring," Expert Syst Appl, vol. 200, p. 116867, Aug. 2022, doi: 10.1016/J.ESWA.2022.116867.

[16] J. Kaushal, "Research on Privacy-Preserving Technology for Cloud Computing," International Journal of Scientific Research & Engineering Trends, vol. 8, pp. 2395–566.

[17] J. Yogesh Deshmukh, S. K. Yadav, and G. M. Bhandari, "Attribute-Based encryption mechanism with Privacy-Preserving approach in cloud computing," Mater Today Proc, vol. 80, pp. 1786–1791, Jan. 2023, doi: 10.1016/J.MATPR.2021.05.609.

[18] N. R. Nayak, A. Kumar, S. Ray, and A. K. Tamrakar, "Blockchain-Based Cloud Resource Allocation Mechanism for Privacy Preservation." EasyChair, Feb. 14, 2023.

[19] Y. Xu, M. Xiao, J. Wu, H. Tan, and G. Gao, "A Personalized Privacy Preserving Mechanism for Crowdsourced Federated Learning," IEEE Trans Mob Comput, pp. 1–17, Jan. 2023, doi: 10.1109/TMC.2023.3237636.

[20] M. Zhang, M. Yang, G. Shen, Z. Xia, and Y. Wang, "A verifiable and privacy-preserving cloud mining pool selection scheme in blockchain of things," Inf Sci (N Y), vol. 623, pp. 293–310, Apr. 2023, doi: 10.1016/J.INS.2022.11.169.

[21] M. A. El-Shorbagy and A. M. El-Refaey, "Hybridization of Grasshopper Optimization Algorithm with Genetic Algorithm for Solving System of Non-Linear Equations," IEEE Access, vol. 8, pp. 220944–220961, 2020, doi: 10.1109/ACCESS.2020.3043029.

[22] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," Archives of Computational Methods in Engineering 2022 29:5, vol. 29, no. 5, pp. 2531–2561, Apr. 2022, doi: 10.1007/S11831-021-09694-4.

[23] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Simulated Annealing Algorithm for Deep Learning," Procedia Comput Sci, vol. 72, pp. 137–144, Jan. 2015, doi: 10.1016/J.PROCS.2015.12.114.

[24] N. Nayar, S. Gautam, P. Singh, and G. Mehta, "Ant Colony Optimization: A Review of Literature and Application in Feature Selection," Lecture Notes in Networks and Systems, vol. 173 LNNS, pp. 285–297, 2021, doi: 10.1007/978-981-33-4305-4_22/COVER.

[25] M. Dubey, V. Kumar, M. Kaur, and T. P. Dao, "A Systematic Review on Harmony Search Algorithm: Theory, Literature, and Applications," Math Probl Eng, vol. 2021, 2021, doi: 10.1155/2021/5594267.

[26] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," Alexandria Engineering Journal, vol. 61, no. 5, pp. 3831–3872, May 2022, doi: 10.1016/J.AEJ.2021.09.013.

[27] L. Shen, J. Li, Y. Wu, Z. Tang, and Y. Wang, "Optimization of Artificial Bee Colony Algorithm Based Load Balancing in Smart Grid Cloud," 2019 IEEE PES Innovative Smart Grid Technologies Asia, ISGT 2019, pp. 1131–1134, May 2019, doi: 10.1109/ISGT-ASIA.2019.8881232.

[28] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," Advances in Engineering Software, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/J.ADVENGSOFT.2017.01.004.

[29] A. Praveena and B. Bharathi, "An approach to remove duplication records in healthcare dataset based on Mimic Deep Neural Network (MDNN) and Chaotic Whale Optimization (CWO)," Concurr Eng Res Appl, vol. 29, no. 1, pp. 58–67, Mar. 2021, doi: 10.1177/1063293X21992014/ASSET/IMAGES/LARGE/10.1177_1063293X21992014-FIG6.JPEG.

[30] A. Ahmad et al., "Toward modeling and optimization of features selection in Big Data based social Internet of Things," Future Generation Computer Systems, vol. 82, pp. 715–726, May 2018, doi: 10.1016/J.FUTURE.2017.09.028.

[31] S. Jiang, J. Cao, H. Wu, K. Chen, and X. Liu, "Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems," Inf Sci (N Y), vol. 635, pp. 72–85, Jul. 2023, doi: 10.1016/J.INS.2023.03.121.

[32] Dikshit Pratyush, Sengupta Jayasree, Bajpai Vaibhav, "Recent Trends on Privacy-Preserving Technologies under Standardization at the IETF," ACM SIGCOMM Computer Communication Review, vol. 53, no. 2, pp. 22–30, Jul. 2023, doi: 10.1145/3610381.3610385.

[33] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: Theory, variants, and

RESEARCH ARTICLE

applications," IEEE Access, vol. 9, pp. 50001–50024, 2021, doi: 10.1109/ACCESS.2021.3067597.

[34] S. Kumar, M. K. Chaube, S. N. Nenavath, S. K. Gupta, and S. K. Tetarave, "Privacy preservation and security challenges: a new frontier multimodal machine learning research," International Journal of Sensor Networks, vol. 39, no. 4, pp. 227–245, 2022, doi: 10.1504/IJSNET.2022.125113.

Authors

Himani Saini received the B.Sc. degree from Ramjas College, Delhi University and M.Sc. degrees in computer science from Maharshi Dayanand University, Rohtak (Haryana) India, in 2017 and 2019, respectively. Currently, she is Ph.D. Research Scholar at the Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, Haryana, India. She has qualified UGC-NET (Computer Science). She may be contacted at email: himani.rs.dcsa@mdurohtak.ac.in.

Dr. Gopal Singh started his career as Lecturer at the Dept. of Computer Sc. & Applications, Ch. Devi Lal University, Sirsa (Haryana) in 2007. Now he is working as Associate Professor in the Dept. of Computer Sc. & Applications, Maharshi Dayanand University, Rohtak (Haryana) since 2009.He completed his Ph. D. from the Department of Computer Sc. & Engineering, UIET, Maharshi Dayanand University, Rohtak (Haryana) in the field of wireless communication i.e., Mobile Ad Hoc Network. He has been Published 20 plus research articles in National/International Journal which are indexing in Scopus/SCI or listed in UGC care list. He participated in more than 30 National/International Conferences organized by different academic institutions. He published one Patent in Patent Office Journal. In spite of these, he is also author of two books. He may be contacted at email: gsbhoria.dcsa@mdurohtak.ac.in.

Dr. Manju Rohil is presently working as Lecturer (Computer Engineering) at CDL Govt. Polytechnic, Nathusari Chopta, Sirsa (Haryana). She obtained her MCA from Kurukshetra University, Kurukshetra in 2005, M.Phil (CS) from CDLU in 2007, Ph. D. from Kurukshetra University, Kurukshetra in 2016. She has qualified GATE, UGC-NET and UGC-JRF. She obtained first rank in JRF in Haryana state and 7th rank in All India. She has published 20 research papers in national and international journals, presented 14 papers in national and international conferences and published three patents in India. Her research interests include cloud computing, data mining, software engineering and operating system. She may be contacted at email: manju.rohil@gmail.com.

How to cite this article:

Himani Saini, Gopal Singh, Manju Rohil, "Design of Hybrid Metaheuristic Optimization Algorithm for Trust-Aware Privacy Preservation in Cloud Computing", International Journal of Computer Networks and Applications (IJCNA), 10(6), PP: 934-946, 2023, DOI: 10.22247/ijcna/2023/223690.