ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse54952020.pdf
https://doi.org/10.30534/ijatcse/2020/54952020

Confidentiality Preserving and Data Portability for Secure Cloud Computing using LCC-AP and SP-GA

Dilip Venkata Kumar Vengala[1], D.Kavitha[2], A.P. Siva Kumar[3]
[1]Research Scholar, Department of Computer Science and Engineering, JNTUA, Anantapur, A.P, India
dilipvenkatakumar@gmail.com
[2]Professor, Department of Computer Science and Engineering, G.Pulla Reddy Engineering College, Kurnool, A.P,India, dwaramkavithareddy@gmail.com
[3]Assistant professor, Department of Computer Science and Engineering, JNTUA College of Engineering, Anantapur, A.P,India.sivakumar.ap@gmail.com

ABSTRACT

Cloud computing (CC) allows obtaining computing resources on-demand and also allows storing avalanche of data with a high fault tolerance level. Security is extremely vital when storing an avalanche of data on the cloud for resolving a number of issues. Though several existing works were there to store data on the cloud, still there are some problems that the data owner has to suffer, such as lack of confidentiality, time delay in storing, and also data being attacked because of insider troubles. Thus, this paper strives to strengthen the sensitive Data Confidentiality (DC) in private as well as public cloud storage by proposing a new data portability and access policy scheme to resolve these afore-mentioned problems. A novel algorithm termed Sandpiper Genetic Algorithm (SP-GA) is launched to maintain proper portability between the cloud-severs. Then, Length Caesar Cipher based Access Policy (LCC-AP) technique is utilized to achieve DC. The proposed techniques guarantee DC, support portability, and also provide secured sharing of files among users. Experimental outcomes show the potential efficiency of SP-GA and LCC-AP

Key words: Data portability, Data confidentiality, Cloud Computing, Sandpiper Genetic Algorithm (SP-GA), Entropy based Linear Discriminant Analysis (ENT-LDA) and Length Caesar Cipher based Access Policy (LCC-AP).

1. INTRODUCTION

A novel computing platform, called CC, allows the users to migrate the data on a paradigm [1]. Lately, there was an augmented adoption of CC services, which is on account of the benefits proffered by means of cloud providers, say powerful computations, efficient resource allocation, pay-per-utilization charging schemes, on-demand access, lessening costs, high services scalability, in addition to flexibility [2, 3]. Most cloud users use the storage services or space of cloud at a lower expense to farm out their data; that is why CC is the fastest developing technology for economic benefits for any organization [4]. The augmenting popularity has caused a quick augmentation in the total cloud vendor in the market [5].

Regardless of these benefits, in the time of storing personal data onto the cloud environment, security requirements, like data confidentiality, dramatically rise [6]. Confidentiality is the procedure of protecting data as of illegal access by means of unauthorized persons [7]. A cloud service (CS) provider has unlimited accessibility to the data stored, and this aspect becomes particularly significant when public or hybrid clouds are utilized, and the only thing users can depend upon is the provider's decency [8]. Confidentiality ensures secure communication, prevents malicious attacks, illegitimate tampering, and averts the participant's identity [9]. Thus, issues over the confidentiality of sensitive data are amongst the utmost significant obstacles for the extensive adoption of CS [10].

Numerous approaches were introduced to ameliorate DC in public cloud storage systems and the encryption-centered approach is the foremost amongst them [11]. Nevertheless, cloud service users (CSU) doing encryption in addition to decryption on their data seems to annul the CC benefits. Albeit, this approach gives almost full control to CSU for defining as well as adjusting the information's security levels on the data according to their security requirements and preference [12]. In fact, encryption algorithms present increasingly lower costs [13]. Numerous researchers utilized cryptography techniques wherein the document is encrypted, split, and then amassed it into disparate cloud space [7]. Cryptography becomes a cheap tool that protects confidentiality when communicating or storing data.

Besides, there is a requirement for secure key management as well as file sharing approaches. Additionally, the solution ought to reconsider the data portability, that is, users should be capable of encrypting/decrypting their data from any place and at any time [14]. The ability to move data amongst disparate application programs, CS, computing environments is called the Data portability [15, 16]. The application’s

components, their relations, and management ought to be designed in a portable, machine-readable, and standardized format for facilitating the creation of portable cloud data together with the automation of their management and deployment [17]. If the services are portable, the chances of resource sharing betwixt the clouds are high, which in turn will improve user data privacy, security, in addition to reliability [18].

Although numerous data portability and DC techniques were introduced, they still suffer from low security as well as privacy, high encryption-decryption time, and lack of real-time solutions, leading to insufficiency in cloud data transmission because of a lack of effectiveness. This paper proposes a Sandpiper Genetic Algorithm (SP-GA) and L-Caesar Cipher centred Access Policy (LCC-AP) technique for a secure data transmission on the cloud to triumph over these prevailing shortcomings. This scheme mostly deals with data portability and confidentiality and renders a general paradigm for deploying user data in cloud storage.

The draft structure for this paper is systematized as Section 2 surveys the related works regarding the proposed method. In sections 3, a concise discussion about the proposed work is proffered. Section 4 explores the experimental outcome, and section 5 infers the work.

2. RELATED WORKS

Alomari et al. [19] suggested a CD Port (Cloud Data Portability) architecture with a standard API and data model for NoSQL and SQL Cloud-centric database systems. This method especially concealed probable variations of backend of the data storage models as of the application layer. For the conversion, transformation, and transfer of data among the disparate data storage models, the Structure was fitted with tools. As this framework was flexible, it could be simply extended for supporting other data storage structures. The CD Port model does not assist complex NoSQL data models, including the graph model.

Makkaoui *et al.* [20] suggested the Cloud-RSA's two variants: i) Rebalanced Cloud-RSA together with ii) MultiPower Cloud-RSA for enhancing the cloud data confidentiality. The variants were centred on the forms of suggested Cloud-RSA and changing exponents. A modulus of 2 or more unique primes was utilized by the 1st variant for encryption and decryption by utilizing the Chinese remaining theorem (CRT). A modulus in the form of $n = prqs$ for $r \geq 2$ and $s \geq 1$ was utilized by the 2nd variant and the CRT and Hensel lifting uses were employed for decryption. The procedure took longer time for massive data decryption.

He *et al.* [21] introduced encryption switching between Identity-centric and Attribute-centric Encryptions. A concrete structure through a proxy reencryption approach was executed. For q-decisional parallel bi-linear Diffie-Hellman exponent presumption, the design was displayed to be a secured CPA in a standard framework. In the game-centric structure, the security definition was described. Its biggest downside was the higher re-encryption and decryption time.

Chen and Guo [22] put forward a Random Space Data Perturbation (RASP) approach for shielding data privacy and utilized the boosting approach to resolve the issue of learning high-quality categorizers as of RASP confusing data. The 4 cloud-client collaborative boosting algorithms: DSPool, LCPool, DerivedDS, and DerivedLC were created, that needed minimum cost in customer side computation and communication. During the learning model process, the customer did not require to stay online. From the outcomes, the model confidentiality under the protection presumption was well protected.

Ramesh *et al.* [23] propounded a framework termed secure e-stream cipher-centered encryption/decryption as the ChaCha20 framework for sustaining proper security to the user's sensitive data at the cloud. For maintaining authenticity along with integrity betwixt VM's disks, a methodology termed dynamic Merkle hash B+ tree (dynamic version) with q-SDH secured short signature with no random oracle signature framework was developed. The parameters, like flexibility, reliability, and scalability of this approach had to be improved.

Rawal*et al.* [24] recommended a secured disintegration protocol for effectively protecting privacy in the CC environment. The architecture was chiefly utilized for cloud storage, and it was employed together with the presented unique data encoding and compression approach. The probabilistic analyses were deployed for evaluating the recommended protocol’s competency of intrusion tolerances. Resource Allocator was employed for delivering a data packet to the servers. Furthermore, the RA split the huge data into innumerable chunks. At last, these chunks were transferred to disparate servers. SDP rendered higher CPU utilization; nevertheless, it failed to effectively utilize the cryptography method.

Makkaoui*et al.* [25] put forward “2” Cloud–Paillier (CP) variants to effectually accelerate their decryption process: a) MultiPrime CP and b) Fast CP. The former scheme holds the same sort of the CP encryption and employed the CRT for decryption. The Fast CP scheme slightly altered the CP’s encryption algorithm and decrypted it. The Fast CP encryption process was slower on considering that of the CP and the MultiPrime CP.

3. CONFIDENTIALITY PRESERVING AND DATA PORTABILITY IN CLOUD COMPUTING

Cloud data portability stands as the competency to transfer data easily from one CS to another or between a customer’s system and CS. Data portability is given more importance since loads of organizations are storing the ever-more greater sum of data within the cloud. The data portability scheme helps in preventing cloud data from the attackers; nevertheless, it will starve the services if it isn’t managed precisely. Conversely, because of this avalanche of data, it is really very challenging to provide high-quality DC. These days, numerous malware injection attacks are performed to

take over a user's information in the cloud. Thus, it's significant to guard DC over the cloud. To avert attacks on the data and ameliorate CC security, this work proposes novel data portability as well as access policy for secure CC. The proposed methodology's architecture is exhibited in figure 1. "Temperature (K)," not "Temperature/K."

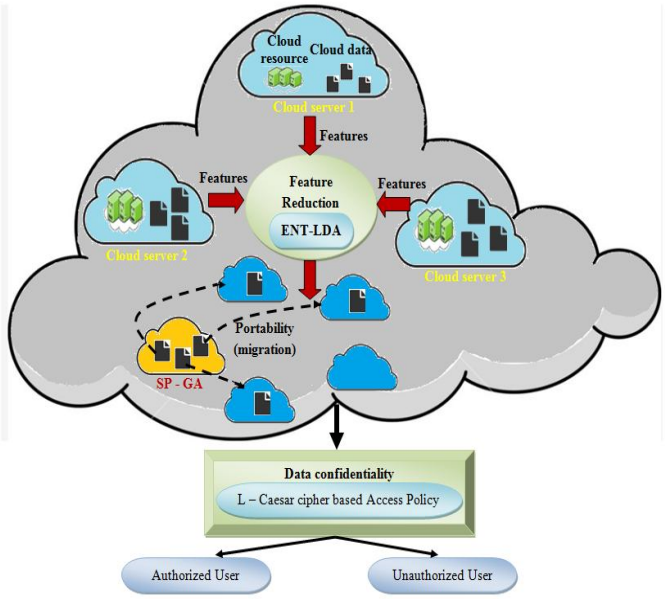

Figure 1: Architecture of Proposed Methodology

3.1 Distributed Cloud Server (DCS) Data

The split files amassed in the DCS are taken as input data. In DCS, manifold files are amassed amongst multiple servers that are in a disparate location. The N -number of cloud server has an N -number of split files, which is expressed mathematically as,

$$DCS_k^{SF} = \{SF_1, SF_2, SF_3, \ldots\ldots\ldots, SF_N\} \quad (1)$$

Wherein, DCS_k^{SF} implies the split file set, SF_N signifies N -number of split files. These split files are taken as tasks.

3.2 Task Features and Resource Features Identification

Centred on inputted task features together with their equivalent cloud resource features, the data portability is performed. The features say file size, file type, file memory utilization, and file ID, are extracted as of the inputted tasks. Similarly, input tasks' corresponding resources features, like cloud memory, disk space, and bandwidth, are also extracted from the disparate cloud servers. These features are elucidated as,

File ID (SF_k^{ID})

The file ID is a sort of identity, and each file encompasses an individual file ID, which will be provided during the file uploading time. The file ID can well be written as,

$$SF_k^{ID} = \{SF_1^{ID}, SF_2^{ID}, SF_3^{ID}, \ldots\ldots, SF_N^{ID} \quad (2)$$

Where, SF_k^{ID} refers to file ID set and SF_N^{ID} refers N -number of file IDs.

File Size (SF_k^s)

It gauges the total data in a split file or, the storage it consumes. The SF_k^s of each split file is expressed as,

$$SF_k^s = \{SF_1^s, SF_2^s, SF_3^s, \ldots\ldots, SF_N^s \quad (3)$$

Where, SF_N^s implies N -number of file sizes

File Type (SF_k^t)

It is a name given for the specific sort of file. Some file formats are web text pages (.htm or.html), Word document (.doc), Adobe Acrobat file (.pdf), Multi-media file (.mp3 and also others) and Web page images (.gif as well as.jpg). The SF_k^t of N -number of files is described as,

$$SF_k^t = \{SF_1^t, SF_2^t, SF_3^t, \ldots\ldots, SF_N^t \quad (4)$$

Where, SF_N^t implies N -number of file types

File memory Utilization (M_U)

It implies the amount of cloud memory utilized by the stored file and is evaluated as,

$$M_U = T_{memory} - U_{memory} \quad (5)$$

Here

T_{memory} - Total memory space

U_{memory} - Used memory

Cloud Memory (M_c)

It signifies the total storage capacity of the cloud server and is evaluated as,

$$M_c = N_{SL} * S_{SL} \quad (6)$$

Where,

N_{SL} - Number of storage locations

S_{SL} - Size of each storage location

Bandwidth (BW)

It signifies the maximum data transfer rate of a cloud server, which is evaluated as,

$$BW = N_{files} * F_{weight} \quad (7)$$

Where,

N_{files} - Total number of files

F_{weight} - File weight

Disk Space (DS_c)

Disk space of the cloud server is computed centred on the server space and used space, which is mathematically expressed as,

$$DS_c = S_{free} + S_{used} \quad (8)$$

Where, S_{free} and S_{used} represents the free space and the used space of the cloud server. The extracted features can well be expressed as,

$$(f_T^R)_k = \{(f_T^R)_1, (f_T^R)_2, (f_T^R)_3, \ldots\ldots\ldots, (f_T^R)_N\} \quad (9)$$

Where, $(f_T^R)_k$ refers the extracted task features and resource feature s, $(f_T^R)_N$ refers N -number of features. To achieve efficient data portability, these extracted features undergo feature reduction.

3.3 Feature Reduction utilizing ENT-LDA

Feature reduction, also known as dimensionality reduction, lessens the number of features in a resource-heavy computation devoid of dropping vital information. The total features (or variables) are lessened to make the computer's work easier and faster. This proposed work utilizes Entropy-centric Linear Discriminants Analysis (ENT-LDA) for reducing the features. The existing LDA may lose some data. For averting this data loss, the entropy approach is included in LDA. LDA is an eminent scheme for feature extraction together with dimension reduction. The dataset is inputted into moderate dimensional-space utilizing LDA with a real class of separable features that lessen computational costs and overfitting. The general steps involved in ENT-LDA's feature reduction are,

Step 1: At first**,** the above-extracted features are formed as a matrix $(f_T^R)_k = [(f_T^R)_{1,1}, (f_T^R)_{1,2}, \ldots\ldots\ldots, (f_T^R)_{N\times M}]$, here N implies number of features, M signifies dimension of $(f_T^R)_k$. The feature matrix is now partitioned to $c = n$ classes as,

$$(f_T^R)_k \rightarrow p_i = \{p_1, p_2, p_3, \ldots., p_n\} \quad (10)$$

Where,

c - Classes

p_i - i^{th} -class

Step 2: The M -dimensional mean vectors for the disparate classes ($\mu^{(c)}$) were evaluated as of the matrix as,

$$\mu^{(c)} = \frac{1}{N^{(c)}} \sum_{(f_T^R)_k \in p^{(c)}} (f_T^R)_k \quad (11)$$

The total mean of all features (μ) is evaluated as,

$$\mu = \sum_{i=1}^{c} \frac{n_i}{N} \mu^{(c)} \quad (12)$$

Where,

n_i - Number of samples in p_i

The entropy value of $\mu^{(c)}$ and μ is evaluated as,

$$E_{\mu^{(c)}} = -\sum \rho(\mu^{(c)}) \log \rho(\mu^{(c)}) \quad (13)$$

$$E_{\mu} = -\sum \rho(\mu) \log \rho(\mu) \quad (14)$$

Step 3: The two LDA matrices termed within-class ($\delta_w^{(c)}$) along with between-class ($\delta_b^{(c)}$) is evaluated as,

$$\delta_w^{(c)} = \sum_{i=1}^{N} \sum_{j\in c} \left((f_T^R)_{k_i} - E_{\mu^{(c)}}\right)\left((f_T^R)_{k_i} - E_{\mu^{(c)}}\right)^T \quad (15)$$

$$\delta_b^{(c)} = \sum_{i=1}^{c} n_i \left(E_{\mu^{(c)}} - E_{\mu}\right)\left(E_{\mu^{(c)}} - E_{\mu}\right)^T \quad (16)$$

Where,

$(f_T^R)_{k_i}$ - i^{th} feature in the c^{th} class

Step 4: Build a transformation matrix for every class (m_T) as,

$$m_T = (\delta_w^{(c)})^{-1} \delta_b^{(c)} \quad (17)$$

Step 5: Evaluate the eigenvectors $(e_1, e_2, e_3, \ldots., e_M)$ with theirequivalenteigenvalues $(\lambda_1, \lambda_2, \lambda_3, \ldots., \lambda_M)$ for $\delta_w^{(c)}$ and $\delta_b^{(c)}$ scatter matrices.

Step 6: Sort those eigenvectors by lessening eigen values and pick v eigenvectors with the topmost eigenvalues to form $M \times v$ dimensional matrix S . At last, a transformed feature set is attained as,

$$\Phi_k^{dr} = (f_T^R)_k^T S \quad (18)$$

Where, $S = [s_1, s_2, s_3, \ldots., s_M]$ and Φ_k^{dr} signifies a dimension reduced feature set, which is formed as a linear combination of all input features $(f_T^R)_k$ with weights S . ENT- LDA effectively lessens the original space dimension of features.

3.4 data Portability

Users can take and re-use the personal data for individual purposes across disparate CSs with the right of data portability. This right lets them to copy, move, or transfer personal data easily as of one cloud server to another between a DCS at a specific interval in a secure way, without disturbing its usability. Hence, even though the malicious owner discovers any specific file location by cheating the cloud server, he could not get that targeted file, since the file is moved as of that location and stored in another location and no files will be existent on that file location. This process automatically elevates data security. This work employs SP-GA for effective data portability. Contingent on the resource- and task- features, the SP-GA chooses an appropriate cloud server. The split files are then migrated to that cloud server at a specified interval.

3.4.1 Sandpiper Genetic Algorithm (SP-GA)

The integration of the SP algorithm with the Genetic algorithm is concerned as SP-GA. The SP algorithm shows less convergence speed as a demerit. For ameliorating the Search Agent (SA) speed, crossover as well as mutation operators are hybridized with the SP algorithm during updation. The SP-GA algorithm inspires the sandpipers' migration as well as attacking behaviors. Here, the migration is concerned as a seasonal movement of sandpipers as of one place to another for locating the food rich sources that would proffer the required energy. Sandpipers often attack the migrating birds over the sea during their migration. During attacking, they could make their natural spiral shape movement. These behaviors are formulated to attain the data portability. Mathematically, the sandpiper behaviors are expounded as,

Migration Behavior: Here, a sandpiper should satisfy the succeeding "3" conditions: i) collision avoidance, ii) converge in the direction of best neighbor's, and iii) updation, centered on the best SA. The new searching agent position is evaluated by deploying searching agent movement ξ_A to avert the collision between their neighbor sandpipers.

$$\vec{\xi}_{sp} = \xi_A \times \vec{C}_{sp}(t) \tag{19}$$

$$\xi_A = f_\xi - \left(t \times (f_\xi / t_{\max})\right) \tag{20}$$

Where,

$\vec{\xi}_{sp}$ - SA's position, and this position does not collide with other SA,

t - Current iteration,

$\vec{C}_{sp}(t)$ - Current position of SA,

f_ξ - Control frequency, which is linearly decreased as of f_ξ to 0,

Subsequent to collision avoidance, the SAs move towards the direction of the best neighbor.

$$\vec{\xi}_L = \xi_B \times \left(\vec{C}_{best}(t) - \vec{C}_{sp}(t)\right) \tag{21}$$

Where,

$\vec{\xi}_L$ - SA location, here $\vec{C}_{sp}(t)$ travels towards the best fittest SA $\vec{C}_{best}(t)$

ξ_B - Random variable

$$\xi_B = 0.5 \times rd \quad rd \in [0,1] \tag{22}$$

Finally, the SA could update its position corresponding to the best SA.

$$\vec{G}_{sp} = \vec{\xi}_{sp} + \vec{\xi}_L \tag{23}$$

Where, $\vec{G}_{sp}$ signifies a gap betwixt the SA and the best fittest SA. For improving the SA speed, crossover as well as mutation is employed.

$$C2P = \left\{ \begin{array}{l} C2P_1 = \left| \dfrac{\vec{G}_{sp}}{3} \right| \\ C2P_2 = C2P_1 + \left| \dfrac{\vec{G}_{sp}}{3} \right| \end{array} \right\} \tag{24}$$

Where, $C2P$ indicates crossover function, $C2P_1$ and $C2P_2$ signifies the crossover points. Subsequently, the employed mutation operator alters the parts of the $\vec{G}_{sp}$ arbitrarily. During mutation, the worst solutions are isolated, which are swapped with a new random solution $\vec{G}''_{sp}$.

Attacking Behavior: Sandpipers generate spiral behavior in the air, while attacking their prey, which is described in the 3-D plane as,

$$S_{behavior} = \begin{cases} \hat{a} = S_r \times \sin(\theta) \\ \hat{b} = S_r \times \cos(\theta) \\ \hat{c} = S_r \times \theta \end{cases}, \quad 0 \le \theta \le 2\pi \tag{25}$$

Where,

S_r - Radius of each turn of the spiral

At last, the best SA is updated as,

$$\vec{C}_{sp}(t) = \left(\vec{G}''_{sp} \times (\hat{a} + \hat{b} + \hat{c})\right) \times \vec{C}_{best}(t) \tag{26}$$

Here, $\vec{\xi}_{sp}$ updates the positions of other SAs. The best solution is updated here grounded on the fitness value. Repeatedly execute the above process till the optimum solution is attained. The SP-GA has the subsequent pseudo-code (fig 2),

Input: Dimension Reduced Features
Output: Suitable Cloud Server

Begin
  Initialize the parameters ξ_A and ξ_B
  Calculate the fitness of each search agent
  $\vec{C}_{best}(t) \leftarrow$ the best searching agent
  while $(t < t_{\max})$ **do**
    for each search agent **do**
      Apply crossover and mutation
      Update the position of the search agent $\vec{C}_{sp}(t)$
    end for
    Update the parameter ξ_A and ξ_B
    Calculate the fitness value of each search agent
    if ($\vec{C}_{sp}(t) > previous\ optimal\ solution$)
      Update the position of the search agent $\vec{C}_{sp}(t)$
    end if
    $t \leftarrow t+1$
  end while
  Return $\vec{C}_{best}(t)$
End

Figure 2: Pseudocode of Sandpiper Genetic Algorithm

The suitable cloud server for the split file is chosen with the SP-GA algorithm. In SP-GA, $\vec{C}_{best}(t)$ signifies the best SA. The best solution signifies a suitable cloud server that is concerned as an optimally selected cloud server for data portability. In this manner, the user data are moved as of one cloud server to another or between a DCS at a specific time interval.

3.5 Data Confidentiality

The security principle that controls the access to information is concerned as Confidentiality. Whilst ensuring the right people to access sensitive information, it also ensures that the wrong (unauthorized) people are not accessing it. Protecting data confidentiality stands as a shared responsibility framework since ultimately the cloud user needs to make-certain their account is not abused. If the password is "1234", one could not blame the CS provider when someone guesses that password and pilfers information. Multi-factor authentication signifies the capability of averting criminals as of logging-in using a stolen password. For effectual user authentication, this work employs a Length Caesar Cipher-centric access policy (LCC-AP) approach. Access policy is a core constituent of security compliance programs that facilitates security technology, and access control policies are there to shield confidential information (user data).

LCC is the simplest and utmost widely known encryption approach wherein each letter in the plaintext is 'shifted' a certain number of places down the alphabet. In this work, the above-extracted features are transmuted into ciphertext. For this, length of $(f_T^R)_k$ is evaluated, and grounded on this length value ($len(f_T^R)_k$), each letter in plaintext is shifted. Mathematically, the encryption of a particular letter through a shift $len(f_T^R)_k$ is proffered as,

$$(f_T^R)_{k\,encrypt} = \left((f_T^R)_k + len\left((f_T^R)_k\right)\right) \bmod 26 \quad (27)$$

Where, $(f_T^R)_{k\,encrypt}$ implies ciphertext of extracted features. While uploading the file, the cloud server delivers the $(f_T^R)_{k\,encrypt}$ to the users (data owners). If any user endeavors to download any file as of the cloud server, the server notifies them to send this ciphertext. When the user gives exact ciphertext, the cloud server confirms that the user is the authorized one and it allows accessing the data. Mathematically, LCC-AP approach is described as,

$$\left(f_T^R\right)_{k\,encrypt} \xrightarrow{Matched} \left(CS \xrightarrow{Confirms} U_{authorized}\right) \quad (28)$$

$$\left(f_T^R\right)_{k\,encrypt} \xrightarrow{Not\ Matched} \left(CS \xrightarrow{Confirms} U_{unauthorized}\right) \quad (29)$$

Where,

$U_{authorized}$ - Authorized users

$U_{unauthorized}$ - Unauthorized users

CS - Cloud server

The objective of this access policy is to lessen the risk of accessing the physical systems by $U_{unauthorized}$. The proposed LCC-AP has the subsequent pseudo-code (figure 3),

Input: Extracted features $(f_T^R)_k = \left\{(f_T^R)_1, (f_T^R)_2, (f_T^R)_3, \ldots\ldots\ldots, (f_T^R)_N\right\}$
Output: User Authentication

Begin
  Generate $len(f_T^R)_k$
  for each features **do**
    // Ciphertext Generation
      $\left(f_T^R\right)_{k\,encrypt} = \left((f_T^R)_k + len\left((f_T^R)_k\right)\right) \bmod 26$
    // User Authentication
      if $\left(f_T^R\right)_{k\,encrypt}$ **matched**
        $\left(f_T^R\right)_{k\,encrypt} \xrightarrow{Matched} \left(CS \xrightarrow{Confirms} U_{authorized}\right)$
      else
        $\left(f_T^R\right)_{k\,encrypt} \xrightarrow{Not\ Matched} \left(CS \xrightarrow{Confirms} U_{unauthorized}\right)$
      end if
  end for
End

Figure 3: Pseudocode of LCC based Access Policy method

4 RESULTS AND DISCUSSION

Here, a performance assessment is made for verifying the proposed approaches' effectiveness. Firstly, the performance shown by the proposed SP-GA used in the data portability is contrasted to the Genetic Algorithm (GA), Artificial Bee Colony (ABC), Sandpiper Algorithm (SPA) as well as Particle Swarm Optimization (PSO), together with. Secondly, the performance rendered by the proposed LCC technique for preserving data confidentiality is contrasted to the existing techniques, namely Aiffin Cipher (AC), Vignere Cipher (VC), Substitution Cipher (SC), along with Caesar Cipher (CC).

4.1 Performance Analysis of SP-GA

The proposed data portability approach for secure CC is contrasted to the existing techniques centered on performance for evaluating its quality level. The SP-GA is contrasted to the existing ABC, PSO, GA, and SPA in respect of fitness, make span, memory usage, together with execution time (ET).

4.11 Evaluation of Fitness

Fitness defines the proposed work's effectiveness. The proposed SP-GA is contrasted to the existing ABC, GA, PSO, and SPA techniques concerning fitness level. Table 1 proffers the fitness level of each algorithm with respect to different iterations.

Table 1: Fitness Comparison

Techniques	Number of Iteration				
	5	10	15	20	25
Existing PSO	46	52	63	74	83
Existing ABC	52	64	72	85	94
Existing GA	61	73	82	94	105
Existing SPA	70	83	95	101	114
Proposed SP-GA	84	97	104	114	125

On analyzing the above table, the proposed SP-GA has a higher fitness level on considering the existing approaches for all the iteration level. It means that the proposed SP-GA effectively performs the data portability contrasted to the existing technique. This fitness comparison is illustrated further using fig 4,

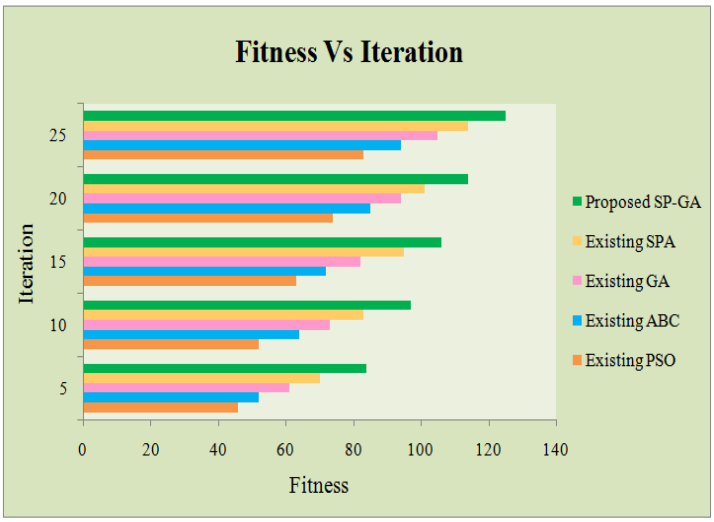

Figure 4: Fitness Vs Iteration Analysis

The performance rendered by the existing and proposed techniques in respect of fitness level is contrasted in fig 4. For 5 to 25 iterations, the proposed SP-GA shows 84, 97, 104, 114, and 125- fitness level respectively, whereas the existing PSO shows 46, 52, 63, 74, and 84- fitness level. The PSO has poor performance while contrasted to the existing SPA, GA, and ABC approaches. The SPA technique attains good performance, but, while contrasting to the proposed SP-GA, the SPA's performance is low. This comparative analysis corroborates the proposed SP-GA's effectiveness.

4.12 Evaluation of Makespan

Makespan is a time requisite to move the data as of one cloud server to another. Makespan is gauged as a time difference between the initialization of data portability and the completion of data portability. Table 2- proffers the attained makespan values for the disparate data sizes.

Table 2: Makespan (in milliseconds) Comparison

Techniques	**File size (in megabyte)**				
	5	**10**	**15**	**20**	**25**
Existing PSO	7635.28	8335.32	8674.78	9324.57	9863.47
Existing ABC	7235.63	7632.54	8235.36	8745.36	9124.85
Existing GA	6687.33	7234.33	7765.32	8124.68	8635.78
Existing SPA	5786.44	6247.52	6845.24	7124.65	7684.25
Proposed SP-GA	5034.31	5632.74	6147.32	6785.32	7132.54

The performance regarding makespan shown by the existing and proposed technique is evaluated in table 2. Makespan varies with the size of the data ranging from 5mb to 25mb. For 5mb data, the proposed SP-GA has 5034.31ms makespan, whereas, the existing PSO, ABC, GA, and SPA have 7635.28ms, 7235.63ms, 6687.33ms, and 5786.44ms makespan respectively. From this analysis, it is perceived that GA and SPA performance is far better on considering ABC and also PSO. And, the SPA performance value is slightly closer to the proposed SP-GA. Nevertheless, the proposed SP-GA proffered a topmost performance on considering the existing approaches. Table 2 is graphically expounded using fig 5,

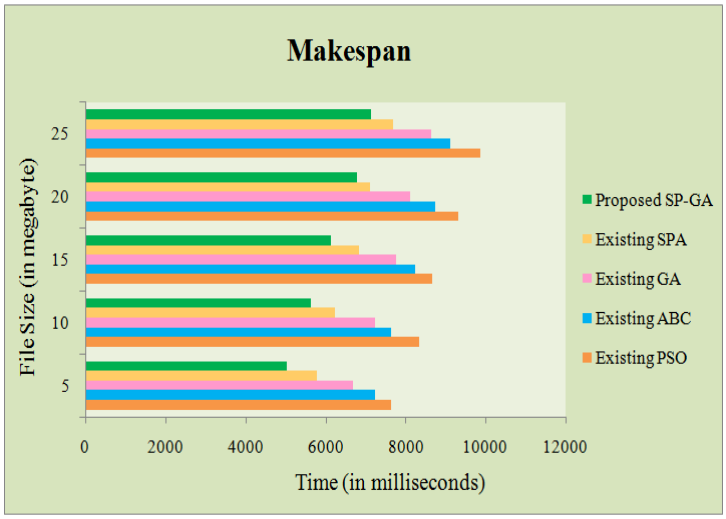

Figure 5: Makespan Analysis

4.13 Evaluation of Memory usage

Memory usage signifies the amount of memory utilized by the technique whilst performing data portability and is gauged in kilobytes. For efficient data portability, memory usage should be low. Table 3 proffers the memory usage level of the existing and proposed algorithm.

Table 3: Memory usage (in kilobytes) Comparison

Techniques	Number of Iteration				
	5	10	15	20	25
Existing PSO	69855	72325	79245	82335	85447
Existing ABC	66358	68974	75325	78965	81224
Existing GA	62456	66344	71667	75887	77668
Existing SPA	58886	61244	65773	69885	71456
Proposed SP-GA	53334	57776	60350	64778	66895

While comparing the memory usage of both proposed and existing techniques, it is perceived that the SP-GA has lower memory usage. If memory usage is low, the proposed SP-GA approach is confirmed to perform efficient portability without wasting any resources. The proposed SP-GA uses 53335kb memory for 5 iterations, where the existing PSO, ABC, GA, and SPA approaches occupy 69855kb, 66358kb, 62456kb, and 58886kb memory, respectively, which are higher on considering the proposed SP-GA. For 10, 15, 20, and 25 iterations, the proposed SP-GA uses 57776kb, 60350kb, 64778kb, and 66895kb memory, respectively. From this analysis, the proposed SP-GA is found to render higher performance on considering the existing approaches. Table 3 could be graphically elucidated using figure 6.

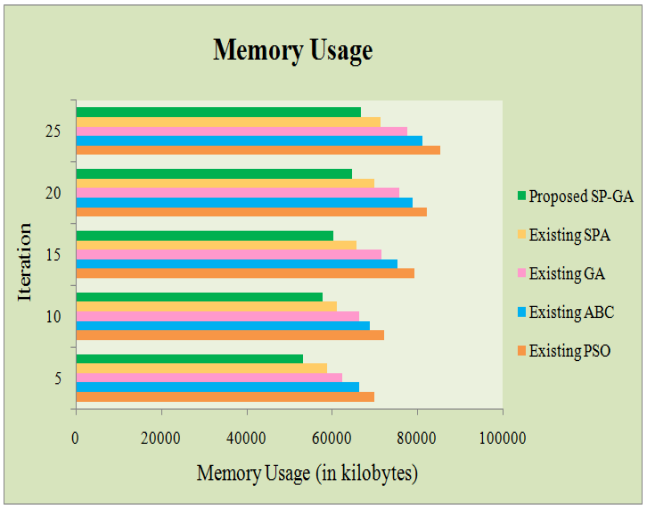

Figure 6: Memory usage analysis

4.14 Evaluation of Execution Time

ET refers to the time taken for selecting a cloud server to perform data portability. The ET values proffered by the existing and proposed techniques are compared by varying the iteration level, which is enumerated using table 4.

Table 4: Execution Time (in milliseconds) Comparison

Techniques	Number of Iteration				
	5	10	15	20	25
Existing PSO	5234.36	7235.63	12247.35	16574.23	22347.25
Existing ABC	4865.32	6952.36	11689.25	15324.36	21554.85
Existing GA	4578.64	6485.47	9658.36	13568.37	19784.83
Existing SPA	4263.58	6354.25	7563.14	11579.15	17324.58
Proposed SP-GA	3954.57	6028.34	6235.36	10235.34	16785.19

Table 4 proffers the performance values in respect of ET acquired by the proposed and existing techniques. For 5 iterations, the ET of the existing PSO, ABC, GA, and SPA is 5234.36ms, 4865.32ms, 4578.64ms, and 4263.58ms respectively, but the proposed SP-GA takes only 3954.57ms to perform cloud server selection for data portability. Likewise, for the 10 to 25 iterations also, the proposed SP-GA takes less ET contrasted to the existing approaches. The less ET signifies that the technique is faster. This ET comparison is graphically expounded using fig 7.

Figure 7: Execution Time Analysis

4.2 Performance analysis of LCC

4.21 Evaluation of Security

To verify the effectiveness of LCC, its performance is contrasted to the existing approaches like AC, VC, SC, and CC in respect of security level.

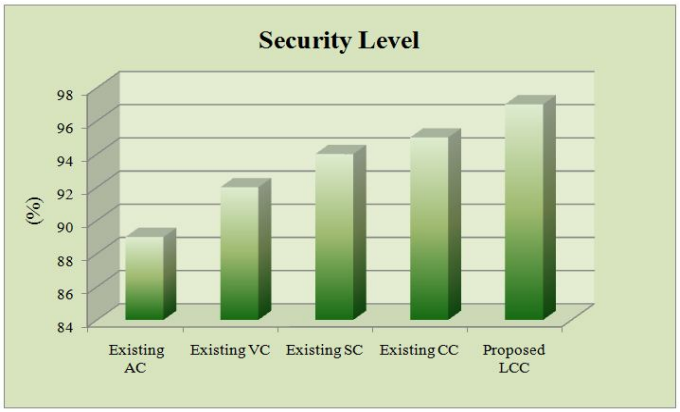

Figure 8: Security level Analysis

Figure 8 contrasts the proposed and existing techniques centered on performance in respect of security level (in percentage). From the figure, the existing AC, VC, SC, and CC shows 89%, 92%, 95%, and 95% security level, whereas, the proposed LCC shows 97% security level. Consequently, the proposed LLC is found to efficiently preserve data confidentiality in the cloud.

4.22 Evaluation of Access Policy Creation Time

Access policy creation time implies the time taken for creating an access policy for the user authentication process. The proposed LCC-AP technique consumes 2137ms time to generate access policy.

5 CONCLUSIONS

For preventing the cloud paradigm from the impacts of attacks, data has to be secured and should withhold security features, like integrity and confidentiality. This paper achieves effective data portability and data confidentiality for secure CC by proposing two novel techniques: SP-GA and LCC-AP. This work aims to lessen the effort and time requisite for porting data across disparate clouds and strengthen the sensitive data confidentiality in public and private cloud storage. For determining the proposed technique's effectiveness, the performance of SP-GA and LCC-AP is contrasted to four existing techniques. From the outcomes, the proposed SP-GA and LCC-AP technique proffered an excellent performance on considering existing techniques. The SP-GA acquired a low makespan of 7132ms with 66895kb memory usage. The proposed LCC-AP yielded a higher-security level of 97% with low access policy creation time of 2137ms. From the performance analysis, the proposed SP-GA and LCC-AP techniques are found to be more secure and efficient when contrasted to the existing techniques.

REFERENCES

1. Asif Showkat Mattoo, Divya Upadhyay, Ashwani Kumar Dubey, and Manoj Kumar Shukla, "An approach to analyse and protect data on Untrusted Cloud Network", In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE,pp.139-144, 2020.10.1109/Confluence47617.2020.9058012.
2. Khalid El Makkaoui, Abderrahim Beni-Hssane, and Abdellah Ezzati, "MultiPrime Cloud-RSA Scheme to Promote Data Confidentiality in the Cloud Environment", In Proceedings of the Mediterranean Symposium on Smart City Applications, Springer, Cham, pp. 445-452, 2017.10.1007/978-3-319-74500-8_41.
3. Pratyush Ranjan, Preeti Mishra, Jaiveer Singh Rawat, Emmanuel S. Pilli, and R. C. Joshi, "Improved technique for data confidentiality in cloud environment", In Networks and Communications (NetCom2013), Springer, Cham, pp. 183-193, 2014.0.1007/978-3-319-03692-2-15.
4. Basappa Kodada, B., and Demian Antony D'Mello, "DCaP—Data Confidentiality and Privacy in Cloud Computing: Strategies and Challenges", In Advances in Machine Learning and Data Science, Springer, Singapore, pp. 225-238, 2018.10.1007/978-981-10-8569-7_24.
5. Aparna Vijaya, and V. Neelanarayanan, "A model driven framework for portable cloud services: Proof of concept implementation", International Journal of Education and Management Engineering, vol. 5, no. 4, pp. 27, 2015.
6. Si Han, Ke Han, and Shouyi Zhang, "A Data Sharing Protocol to Minimize Security and Privacy Risks of Cloud Storage in Big Data Era", IEEE Access, vol. 7, pp. 60290-60298, 2019.
7. Karim Timraz, Tawfiq Barhoom, and Tamer Fatayer, "A Confidentiality Scheme for Storing Encrypted Data through Cloud", In IEEE 7th Palestinian International Conference on Electrical and Computer Engineering (PICECE), IEEE, pp. 1-5, 2019.10.1109/PICECE.2019.8747193.
8. Andrey Rukavitsyn, N., Konstantin A. Borisenko, Ivan I. Holod, and Andrey V. Shorov, "The method of ensuring confidentiality and integrity data in cloud computing", In XX IEEE International Conference on Soft Computing and Measurements (SCM), IEEE, pp. 272-274, 2017.10.1109/SCM.2017.7970558.
9. Nidhi Patel, A, "A Survey on Security Techniques used for Confidentiality in Cloud Computing", In International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), IEEE, pp. 1-6, 2018.10.1109/ICCSDET.2018.8821135.
10. Khalid El Makkaoui, Abderrahim Beni-Hssane, Abdellah Ezzati, and Anas El-Ansari, "Fast cloud-RSA scheme for promoting data confidentiality in the cloud computing", Procedia computer science, vol. 113, pp. 33-40, 2017.
11. Ahmed Bentajer, Mustapha Hedabou, Karim Abouelmehdi, and Said Elfezazi, "CS-IBE: a data confidentiality system in public cloud storage system", Procedia Computer Science, vol. 141, pp. 559-564, 2018.

12. Syed Rizvi, Katie Cover, and Christopher Gates, "A trusted third-party (TTP) based encryption scheme for ensuring data confidentiality in cloud environment", Procedia Computer Science, vol. 36, pp. 381-386, 2014.
13. Eliseu Branco, C., José Maria Monteiro, Roney Reis, and Javam C. Machado, "A New Mechanism to Preserving Data Confidentiality in Cloud Database Scenarios", In International Conference on Enterprise Information Systems, Springer, Cham, pp. 261-283, 2016.10.1007/978-3-319-62386-3_13.
14. Ebtesam Ahmad Alomari, and Muhammad Mostafa Monowar, "Towards Data Confidentiality and Portability in Cloud Storage", In International Conference of Design, User Experience, and Usability, Springer, Cham, pp. 38-49, 2014.10.1007/978-3-319-07626-3_4.
15. Michael Wohlfarth, "Data Portability on the Internet", Business & Information Systems Engineering, vol. 61, no. 5, pp. 551-574, 2019.
16. Sami Yangui, Roch H. Glitho, and Constant Wette, "Approaches to end-user applications portability in the cloud: A survey", IEEE Communications Magazine, vol. 54, no. 7, pp. 138-145, 2016.
17. Tobias Binz, Uwe Breitenbücher, Oliver Kopp, and Frank Leymann, "TOSCA: portable automated deployment and management of cloud applications", In Advanced Web Services, Springer, New York, NY, pp. 527-549, 2014.10.1007/978-1-4614-7535-4_22.
18. Aparna Vijaya, "A Model Driven Framework for Portable Cloud Services", International Journal of Electrical & Computer Engineering (2088-8708), vol. 6, no. 2, 2016.
19. Ebtesam Alomari, Ahmed Barnawi, and Sherif Sakr, "Cdport: A portability framework for nosql datastores", Arabian Journal for Science and Engineering, vol. 40, no. 9, pp. 2531-2553, 2015.
20. Khalid El Makkaoui, Abderrahim Beni-Hssane, and Abdellah Ezzati, "Speedy Cloud-RSA homomorphic scheme for preserving data confidentiality in cloud computing", Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 12, pp. 4629-4640, 2019.
21. Kai He, Yijun Mao, Jianting Ning, Kaitai Liang, Xinyi Huang, Emmanouil Panaousis, and George Loukas, "A new encrypted data switching Protocol: Bridging IBE and ABE without loss of data confidentiality", IEEE Access, vol. 7, pp. 50658-50668, 2019.
22. Keke Chen, and Shumin Guo, "Rasp-boost: confidential boosting-model learning with perturbed data in the cloud", IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 584-597, 2015.
23. Dharavath Ramesh, Rahul Mishra, and Damodar Reddy Edla, "Secure data storage in cloud: an e-stream cipher-based secure and dynamic updation policy", Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 873-883, 2017.
24. Bharat Rawal S, Vijayakumar V, Gunasekaran Manogaran, Varatharajan R, and Naveen Chilamkurti, "Secure disintegration protocol for privacy preserving cloud storage", Wireless personal communications, vol. 103, no. 2, pp. 1161-1177, 2018.
25. Khalid El Makkaoui, Abdellah Ezzati, Abderrahim Beni-Hssane, and Slimane Ouhmad, "Fast Cloud–Paillier homomorphic schemes for protecting confidentiality of sensitive data in cloud computing", Journal of Ambient Intelligence and Humanized Computing, pp. 1-10, 2019.10.1007/s12652-019-01366-3.